Proceedings of AUTOMATA 2011 : 17th International Workshop on Cellular Automata and Discrete Complex Systems

Nazim Fatès, Eric Goles, Alejandro Maass, Ivan Rapaport

▶ To cite this version:

HAL Id: hal-00654706
https://hal.inria.fr/hal-00654706

Submitted on 22 Dec 2011

AUTOMATA 2011

17th International Workshop on Cellular Automata and Discrete Complex Systems

Proceedings

editors:

Nazim Fatès

Eric Goles

Alejandro Maass

Ivan Rapaport

Preface

This volume contains all the contributed papers presented at AUTOMATA 2011, the 17th international workshop on cellular automata and discrete complex systems. The workshop was held on November 21-23, 2011, at the Center for Mathematical Modeling, University of Chile, Santiago, Chile.

AUTOMATA is an annual workshop on the fundamental aspects of cellular automata and related discrete dynamical systems. The spirit of the workshop is to foster collaborations and exchanges between researchers on these areas. The workshop series was started in 1995 by members of the Working Group 1.5 of IFIP, the International Federation for Information Processing.

The volume contains the « full » papers and « short » papers selected by the program committee. The « full papers » will also appear as proceedings in a volume of *Discrete Mathematics and Theoretical Computer Science* (DMTCS). The program committee consisted of 27 international experts on cellular automata and related models, and the selection was based on 3 peer reviews on each paper.

Papers in this volume represent a rich sample of current research topics on cellular automata and related models. The papers include theoretical studies of the classical cellular automata model, but also many investigations into various variants and generalizations of the basic concept. The versatile nature and the flexibility of the model is evident from the presented papers, making it a rich source of new research problems for scientists representing a variety of disciplines.

In addition to the papers of this volume, the program of AUTOMATA 2011 contained four one-hour plenary lectures given by distinguished invited speakers :

- Peter Gacs (Boston University, USA)
- Tom Meyerovitch (University of British Columbia, Canada)
- Nicolas Schabanel (CNRS, Universié Paris VII & ENS Lyon, France)
- Damien Woods (Caltech, USA)

The organizers gratefully acknowledge the support by the following institutions:

- Centro de Modelamiento Matemático
- Departamento de Ingeniería Matemática
- Universidad de Chile
- Conicyt
- CNRS
- Universidad Adolfo Ibáñez

As the editors of these proceedings, we thank all contributors to the scientific program of the workshop. We are especially indebted to the invited speakers and the authors of the contributed papers. We would also like to thank the members of the Program Committee and the external reviewers of the papers. Last but not least, the editors thank Nikolaos Vlassopolous for his valuable help in the compilation of these proceedings.

Nazim Fatès, Eric Goles, Alejandro Maass, Iván Rapaport

Program Committee

Andrew Adamatzky	University of West England, UK
Stefania Bandini	Università degli Studi di Milano-Bicocca, Italy
Marie-Pierre Béal	Université Paris-Est, France
Bruno Durand	Université de Provence, France
Nazim Fatès	Inria Nancy Grand-Est, France, **co-chair**
Paola Flocchini	University of Ottawa, Canada
Enrico Formenti	Université de Nice-Sophia Antipolis, France
Henryk Fuks	Brock University. Canada
Anahí Gajardo	Universidad de Concepción, Chile
Eric Goles	Universidad Adolfo Ibáñez, Chile, **co-chair**
Martin Kutrib	University of Giessen, Germany
Alejandro Maass	Universidad de Chile, **co-chair**
Andrés Moreira	Universidad Técnica Federico Santa María, Chile
Kenichi Morita	Hiroshima University, Japan
Pedro de Oliveira	Universidade Presbiteriana Mackenzie, Brazil
Nicolas Ollinger	Université de Provence, France
Ronnie Pavlov	Denver University, USA
Marcus Pivato	Trent University, Canada
Ivan Rapaport	Universidad de Chile, **co-chair**
Dipanwita Roychowdhury	Indian Institute of Technology, India
Mathieu Sablik	Université de Provence
Michael Schraudner	Universidad de Chile
Klaus Sutner	Carnegie Mellon, USA
Guillaume Theyssier	CNRS, Université de Savoie, France
Edgardo Ugalde	Universidad Autónoma de San Luis Potosí, Mexico
Hiroshi Umeo	Osaka Electro-Communication University, Japan
Thomas Worsch	Karlsruhe University, Germany

Table of Contents

A fixed point theorem for Boolean networks expressed in terms of forbidden subnetworks

Adrien Richard[†]

Laboratoire I3S, CNRS & Université de Nice-Sophia Antipolis, France

We are interested in fixed points in Boolean networks, *i.e.* functions f from $\{0,1\}^n$ to itself. We define the sub-networks of f as the restrictions of f to the hypercubes contained in $\{0,1\}^n$, and we exhibit a class $\mathcal{F}$ of Boolean networks, called even or odd self-dual networks, satisfying the following property: if a network f has no subnetwork in $\mathcal{F}$, then it has a unique fixed point. We then discuss this "forbidden subnetworks theorem". We show that it generalizes the following fixed point theorem of Shih and Dong: if, for every x in $\{0,1\}^n$, there is no directed cycle in the directed graph whose the adjacency matrix is the discrete Jacobian matrix of f evaluated at point x, then f has a unique fixed point. We also show that $\mathcal{F}$ contains the class $\mathcal{F}'$ of networks whose the interaction graph is a directed cycle, but that the absence of subnetwork in $\mathcal{F}'$ does not imply the existence and the uniqueness of a fixed point.

Keywords: Boolean network, fixed point, self-dual Boolean function, discrete Jacobian matrix, feedback circuit.

1   Introduction

A function f from $\{0,1\}^n$ to itself is often seen as a Boolean network with n components. On on hand, the dynamics of the network is described by the iterations of f; for instance, with the synchronous iteration scheme, the dynamics is described by the recurrence $x^{t+1} = f(x^t)$. On the other hand, the "structure" of the network is described by a directed graph $G(f)$: the vertices are the n components, and there exists an arc from j to i when the evolution of the ith component depends on the evolution of the jth one.

Boolean networks have many applications. In particular, from the seminal works of Kauffman (1969) and Thomas (1973), they are extensively used to model gene networks. In most cases, fixed points are of special interest. For instance, in the context of gene networks, they are often seen as stable patterns of gene expression at the basis of particular biological processes.

In this paper, we are interested in sufficient conditions for the existence and the uniqueness of a fixed point for f. Such a condition was first obtained by Robert (1980), who proved that *if $G(f)$ has no directed cycle, then f has a unique fixed point*. This result was then generalized by Shih and Dong (2005). They associated to each point x in $\{0,1\}^n$ a local interaction graph $Gf(x)$, which is a subgraph of $G(f)$ defined as the directed graph whose the adjacency matrix is the discrete Jacobian matrix of f evaluated at point x, and they proved that *if $Gf(x)$ has no directed cycle for all x in $\{0,1\}^n$, then f has a unique fixed point*. Up

to our knowledge, this is the weakest condition known to be sufficient for the presence and the uniqueness of a fixed point.

In this paper, we establish a sufficient condition for the existence and the uniqueness of a fixed point that is *not* expressed in terms of directed cycles. In Section 2, we defined, in a natural way, the subnetworks of f as the restrictions of f to the hypercubes contained in $\{0,1\}^n$, and we introduce the class $\mathcal{F}$ of even and odd self-dual networks. In Section 3, we prove the main result: *if f has no subnetworks in $\mathcal{F}$, then it has a unique fixed point*. The rest of the paper discusses this "forbidden subnetworks theorem". In section 4, we show that it generalizes the fixed point theorem of Shih and Dong mentioned above. In section 5, we study the effect of the absence of subnetwork in $\mathcal{F}$ on the asynchronous state graph of f, which is a directed graph on $\{0,1\}^n$ constructed from the asynchronous iterations of f and proposed by Thomas (1973) as a model for the dynamics of gene networks. Finally, in Section 6, we compare $\mathcal{F}$ with the well-known class $\mathcal{F}'$ of networks f whose the interaction graph $G(f)$ is a directed cycle. Mainly, we show that $\mathcal{F}' \subseteq \mathcal{F}$ and that the absence of subnetwork in $\mathcal{F}'$ is not sufficient for the existence and the uniqueness of a fixed point.

2   Definitions and notations

In this section, we introduce the definitions needed to state and prove the main result. Let $\mathbb{B} = \{0,1\}$, let n be a positive integer, let $[n] = \{1,\dots,n\}$, and let $i \in [n]$. The ith unit vector of $\mathbb{B}^n$ is denoted e_i (all the components are 0, excepted the ith one which is 1). The sum modulo two is denoted $\oplus$. It is applied componentwise on elements of $\mathbb{B}^n$: for all $x, y \in \mathbb{B}^n$,

$$x \oplus y = (x_1 \oplus y_1, \dots, x_n \oplus y_n) \qquad \text{and} \qquad x \oplus 1 = (x_1 \oplus 1, \dots, x_n \oplus 1).$$

Hence, $x \oplus 1$ may be seen as the negation of x. The number of ones that x contains is denoted $\|x\|$, *i.e.* $\|x\| = \sum_{i=1}^n x_i$. Thus $\|x \oplus y\|$ gives the *Hamming distance* between two points x and y of $\mathbb{B}^n$. We say that x is *even* (*odd*) if $\|x\|$ is even (odd) (there exists 2^{n-1} even (odd) points in $\mathbb{B}^n$). The point of $\mathbb{B}^n$ obtained from x by assigning the ith component to $\alpha \in \mathbb{B}$ is denoted $x^{i\alpha}$, *i.e.*

$$x^{i\alpha} = (x_1, \dots, x_{i-1}, \alpha, x_{i+1}, \dots, x_n).$$

If $n > 1$, the point of $\mathbb{B}^{n-1}$ obtained from x be removing the ith component is denoted x_{-i}, *i.e.*

$$x_{-i} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n).$$

We call *(n-dimensional Boolean) networks* any function f from $\mathbb{B}^n$ to itself.

Definition 1 (Conjugate) *The* conjugate *of $f : \mathbb{B}^n \to \mathbb{B}^n$ is the following n-dimensional network:*

$$\tilde{f} : \mathbb{B}^n \to \mathbb{B}^n, \qquad \tilde{f}(x) = x \oplus f(x) \qquad \forall x \in \mathbb{B}^n.$$

Remark that $\tilde{f}(x) = 0$ if and only if x is a fixed point of f, *i.e.* $f(x) = x$.

Definition 2 (Self-dual networks and even/odd networks) *f is* self-dual *if*

$$f(x) = f(x \oplus 1) \oplus 1 \qquad \forall x \in \mathbb{B}^n.$$

f is even *(odd) if the image of $\tilde{f}$ is the set of even points of $\mathbb{B}^n$, i.e.*

$$\{\tilde{f}(x) \mid x \in \mathbb{B}^n\} = \{x \mid x \in \mathbb{B}^n \text{ and } \|x\| \text{ is even (odd)}\}.$$

We say that f is *even (odd) self-dual* if it is both even (odd) and self-dual. Note that $f(x) = f(x \oplus 1) \oplus 1$ if and only if $\tilde{f}(x \oplus 1) = \tilde{f}(x)$. Note also that if f is even (odd) self-dual, then for each even (odd) point $x \in \mathbb{B}^n$, the preimage of x by $\tilde{f}$ is of cardinality two, *i.e.* there exists exactly two distinct points $y, z \in \mathbb{B}^n$ such that $\tilde{f}(y) = \tilde{f}(z) = x$. Since $\tilde{f}(x) = 0$ if and only if $f(x) = x$, we deduce that if f is even self-dual, then it has exactly two fixed points (obviously, if f is odd self-dual, then it has no fixed point).

Definition 3 (Immediate subnetworks) *If $n > 1$, $\alpha \in \mathbb{B}$ and $i \in [n]$, we call* immediate subnetwork *of f (obtained by fixing the ith component to α) the following $(n-1)$-dimensional network:*

$$f^{i\alpha} : \mathbb{B}^{n-1} \to \mathbb{B}^{n-1}, \qquad f^{i\alpha}(x_{-i}) = f(x^{i\alpha})_{-i} \qquad \forall x \in \mathbb{B}^n.$$

Remark that conjugate of $f^{i\alpha}$ is equal to the immediate subnetwork $\tilde{f}^{i\alpha}$ of the conjugate $\tilde{f}$ of f:

$$\widetilde{f^{i\alpha}}(x_{-i}) = x_{-i} \oplus f^{i\alpha}(x_{-i}) = x_{-i} \oplus f(x^{i\alpha})_{-i} = (x \oplus f(x^{i\alpha}))_{-i} = \tilde{f}(x^{i\alpha})_{-i} = \tilde{f}^{i\alpha}(x_{-i}).$$

Definition 4 (Subnetworks) *The* subnetworks *of f are inductively defined by: (1) if $n = 1$, then f has a unique subnetwork, which is f itself; and (2) if $n > 1$, the subnetworks of f are f and the subnetworks of the immediate subnetworks of f. A* strict subnetwork *of f is a subnetwork of f different than f.*

3   Main result

Theorem 1 (Forbidden subnetworks theorem) *If a network $f : \mathbb{B}^n \to \mathbb{B}^n$ has no even or odd self-dual subnetwork, then the conjugate of f is a bijection, and in particular, f has a unique fixed point.*

The proof of Theorem 1 needs the following two lemmas.

Lemma 1 *Let X be a non-empty subset of $\mathbb{B}^n$ and $V(X) = \{x \oplus e_i \mid x \in X, i \in [n]\}$. If X and $V(X)$ are disjoint and $|X| \geq |V(X)|$, then X is either the set of even points of $\mathbb{B}^n$ or the set of odd points of $\mathbb{B}^n$.*

Proof: by induction on n. The case $n = 1$ is obvious. So suppose that $n > 1$ and that the lemma holds for the dimensions less than n. Let X be a non-empty subset of $\mathbb{B}^n$ satisfying the conditions of the statement. Let $\alpha \in \mathbb{B}$, and consider the following subsets of $\mathbb{B}^{n-1}$:

$$X^\alpha = \{x_{-n} \mid x \in X, x_n = \alpha\}, \qquad V(X)^\alpha = \{x_{-n} \mid x \in V(X), x_n = \alpha\}.$$

We first prove that $V(X^\alpha) \subseteq V(X)^\alpha$ and $X^\alpha \cap V(X^\alpha) = \emptyset$. Let $x \in \mathbb{B}^n$ with $x_n = \alpha$ be such that $x_{-n} \in V(X^\alpha)$. To prove that $V(X^\alpha) \subseteq V(X)^\alpha$, it is sufficient to prove that $x_{-n} \in V(X)^\alpha$. Since $x_{-n} \in V(X^\alpha)$, there exists $y \in \mathbb{B}^n$ with $y_n = \alpha$ and $i \in [n-1]$ such that $y_{-n} \in X^\alpha$ and $x_{-n} = y_{-n} \oplus e_i$. So $x = y \oplus e_i$, and since $y_n = \alpha$, we have $y \in X$. Hence $x \in V(X)$ and since $x_n = \alpha$, we have $x_{-n} \in V(X)^\alpha$. We now prove that $X^\alpha \cap V(X^\alpha) = \emptyset$. Indeed, otherwise, there exists $x \in \mathbb{B}^n$ with $x_n = \alpha$ such that $x_{-n} \in X^\alpha \cap V(X^\alpha)$. Since $V(X^\alpha) \subseteq V(X)^\alpha$, we have $x_{-n} \in X^\alpha \cap V(X)^\alpha$, and since $x_n = \alpha$, we deduce that $x \in X \cap V(X)$, a contradiction.

Now, since $V(X^\alpha) \subseteq V(X)^\alpha$, we have

$$|X| = |X^0| + |X^1| \geq |V(X)| = |V(X)^0| + |V(X)^1| \geq |V(X^0)| + |V(X^1)|.$$

So $|X^0| \geq |V(X^0)|$ or $|X^1| \geq |V(X^1)|$. Suppose that $|X^0| \geq |V(X^0)|$, the other case being similar. Since $X^0 \cap V(X^0) = \emptyset$, by induction hypothesis X^0 is either the set of even points of $\mathbb{B}^{n-1}$ or the

set of odd points of $\mathbb{B}^{n-1}$. So in both cases, we have $|X^0| = |V(X^0)| = 2^{n-1}$. We deduce that $|X^1| \geq |V(X^1)|$, and so, by induction hypothesis, X^1 is either the set of even points of $\mathbb{B}^{n-1}$ or the set of odd points of $\mathbb{B}^{n-1}$. But X^0 and X^1 are disjointed: for all $x \in \mathbb{B}^n$, if $x_{-n} \in X^0 \cap X^1$, then x^{n0} and x^{n1} are two points of X, and $x^{n1} = x^{n0} \oplus e_n \in V(X)$, a contradiction. So if X^0 is the set of even (odd) points of $\mathbb{B}^{n-1}$, then X^1 is the set of odd (even) points of $\mathbb{B}^{n-1}$, and we deduce that X is the set of even (odd) points of $\mathbb{B}^n$. □

Lemma 2 *Let $f : \mathbb{B}^n \to \mathbb{B}^n$. Suppose that the conjugate of every immediate subnetwork of f is a bijection. If the conjugate of f is not a bijection, then f is even or odd self-dual.*

Proof: Suppose that f satisfies the conditions of the statement, and that the conjugate $\tilde{f}$ of f is not a bijection. Let $\tilde{X} \subseteq \mathbb{B}^n$ be the image of $\tilde{f}$, and let $X = \mathbb{B}^n \setminus \tilde{X}$. Since $\tilde{f}$ is not a bijection, X is not empty. We first prove the following property:

(∗) For every $x \in X$ and $i \in [n]$, the preimage of $x \oplus e_i$ by $\tilde{f}$ is of cardinality two.

Let $x \in X$ and $i \in [n]$. By hypothesis, the conjugate of f^{i0} is a bijection, so there exists a unique point in $\mathbb{B}^{n-1}$ whose the image by $\tilde{f}^{i0}$ is x_{-i}. We deduce that there exists a unique point $y \in \mathbb{B}^n$ such that $y_i = 0$ and $\tilde{f}^{i0}(y_{-i}) = x_{-i}$. Then, $\tilde{f}(y)_{-i} = \tilde{f}(y^{i0})_{-i} = \tilde{f}^{i0}(y_{-i}) = x_{-i}$. We deduce that either $\tilde{f}(y) = x$ or $\tilde{f}(y) = x \oplus e_i$. Since $x \in X$ we have $\tilde{f}(y) \neq x$ so $\tilde{f}(y) = x \oplus e_i$. Hence, we have proved that there exists a unique point $y \in \mathbb{B}^n$ such that $y_i = 0$ and $\tilde{f}(y) = x \oplus e_i$, and we prove with similar arguments that there exists a unique point $z \in \mathbb{B}^n$ such that $z_i = 1$ and $\tilde{f}(z) = x \oplus e_i$. This proves (∗).

We are now in position to prove that f is even or odd. Let $V(X) = \{x \oplus e_i \mid x \in X, i \in [n]\}$. We have

$$|X| + |\tilde{X}| = 2^n = |\tilde{f}^{-1}(\tilde{X})| = |\tilde{f}^{-1}(V(X))| + |\tilde{f}^{-1}(\tilde{X} \setminus V(X))| \geq |\tilde{f}^{-1}(V(X))| + |\tilde{X} \setminus V(X)|.$$

Following (∗), we have $|\tilde{f}^{-1}(V(X))| = 2|V(X)|$ and $V(X) \subseteq \tilde{X}$, so

$$|X| + |\tilde{X}| \geq 2|V(X)| + |\tilde{X} \setminus V(X)| = 2|V(X)| + |\tilde{X}| - |V(X)| = |V(X)| + |\tilde{X}|.$$

Therefore, $|X| \geq |V(X)|$, and since $V(X) \subseteq \tilde{X} = \mathbb{B}^n \setminus X$, we have $X \cap V(X) = \emptyset$. So according to Lemma 1, X is either the set of even points of $\mathbb{B}^n$ or the set of odd points of $\mathbb{B}^n$. We deduce that in the first (second) case, $\tilde{X}$ is the set of odd (even) points of $\mathbb{B}^n$. Thus, f is even or odd.

It remains to prove that f is self-dual. Let $x \in \mathbb{B}^n$. For all $i \in [n]$, since $\|\tilde{f}(x)\|$ and $\|\tilde{f}(x) \oplus e_i\|$ have not the same parity, and since f is even or odd, we have $\tilde{f}(x) \oplus e_i \in X$. Thus, according to (∗), the preimage of $(\tilde{f}(x) \oplus e_i) \oplus e_i = \tilde{f}(x)$ by $\tilde{f}$ is of cardinality two. Consequently, there exists a point $y \in \mathbb{B}^n$, distinct from x, such that $\tilde{f}(y) = \tilde{f}(x)$. Let us proved that $x = y \oplus 1$. Indeed, if $x_i = y_i = 0$ for some $i \in [n]$, then $\tilde{f}^{i0}(x_{-i}) = \tilde{f}(x)_{-i} = \tilde{f}(y)_{-i} = \tilde{f}^{i0}(y_{-i})$. Since $x \neq y$, we deduce that $\tilde{f}^{i0}$ is not a bijection, a contradiction. We show similarly that if $x_i = y_i = 1$, then $\tilde{f}^{i1}$ is not a bijection. So $x = y \oplus 1$. Consequently, $\tilde{f}(x \oplus 1) = \tilde{f}(x)$, and we deduce that f is self-dual. □

Proof of Theorem 1: by induction on n. The case $n = 1$ is obvious. So suppose that $n > 1$ and that the theorem holds for the dimensions less than n. Suppose that f has no even or odd self-dual subnetwork. Under this condition, f is neither even self-dual nor odd self-dual (since f is a subnetwork of f), and every immediate subnetwork of f has no even or odd self-dual subnetwork. So, by induction hypothesis, the dual of every strict subnetwork of f is a bijection, and we deduce from Lemma 2 that the dual of f is a

bijection. Thus, in particular, there exists a unique point $x \in \mathbb{B}^n$ such that $\tilde{f}(x) = 0$, and since $\tilde{f}(x) = 0$ if and only if $f(x) = x$, this point x is the unique fixed point of f. $\qquad\square$

Clearly, if f has no even or odd self-dual subnetwork, then every subnetwork of f has no even or odd self-dual subnetwork, and according to Theorem 1, the conjugate of every subnetwork of f is a bijection. Conversely, if the conjugate of every subnetwork of f is a bijection, then f has no even or odd self-dual subnetwork, since the conjugate of an even or odd self-dual network is not a bijection. Consequently, we have the following characterization:

Corollary 1 *The conjugate of each subnetwork of f is a bijection if and only if f has no even or odd self-dual network.*

Example 1 $f : \mathbb{B}^3 \to \mathbb{B}^3$ *is defined by:*

$$f(x_1, x_2, x_3) = (\overline{x_2} \wedge x_3, \overline{x_3} \wedge x_1, \overline{x_1} \wedge x_2).$$

Remark that f is not self-dual, since $f(000) = f(111) = 000$. The immediate subnetworks of f are:

$$f^{10}(x_2, x_3) = (0, x_2)$$
$$f^{11}(x_2, x_3) = (\overline{x_3}, 0)$$
$$f^{20}(x_1, x_3) = (x_3, 0)$$
$$f^{21}(x_1, x_3) = (0, \overline{x_1})$$
$$f^{30}(x_1, x_2) = (0, x_1)$$
$$f^{31}(x_1, x_2) = (\overline{x_2}, 0)$$

So each immediate subnetwork $f^{i\alpha}$ of f has one component fixed to zero, and so is not self-dual. Futhermore, each immediate subnetwork of $f^{i\alpha}$ is the one dimensional network h defined by $h(0) = h(1) = 0$, which is not self-dual. So f has no self-dual subnetwork, and we deduce from Theorem 1 that the conjugate of $\tilde{f}$ of f is a bijection, and that f has a unique fixed point. Indeed:

x	$f(x)$	$\tilde{f}(x)$
000	000	000
001	100	101
010	001	011
011	001	010
100	010	110
101	100	001
110	010	100
111	000	111

4 Remarks on the theorem of Shih and Dong

In this section, we show that Theorem 1 implies a fixed point theorem due to Shih and Dong (2005). In order to state this theorem, we need additional definitions. Let

$$f : \mathbb{B}^n \to \mathbb{B}^n, \qquad f(x) = (f_1(x), \ldots, f_n(x)).$$

Definition 5 (Discrete Jacobian matrix) *The* discrete Jacobian matrix *of f evaluated at point $x \in \mathbb{B}^n$ is the following $n \times n$ Boolean matrix*

$$f'(x) = (f_{ij}(x)), \qquad f_{ij}(x) = f_i(x^{j1}) \oplus f_i(x^{j0}) \qquad (i, j \in [n]).$$

In the next definition, we represent $f'(x)$ under the form of a directed graph, in order to use graph theoretic notions instead of matrix theoretical notions. In fact, we mainly focus on *elementary directed cycles*, that we simply call *cycles* in the following.

Definition 6 (Local interaction graph) *The* local interaction graph of f *evaluated at point $x \in \mathbb{B}^n$ is the directed graph $Gf(x)$ defined by: the vertex set is $[n]$, and for all $i, j \in [n]$, there exists an arc $j \to i$ if and only if $f_{ij}(x) = 1$.*

The discrete Jacobian matrix of f was first defined by Robert (1983), who also introduced the notion of Boolean eigenvalue. This material allowed Shih and Ho (1999) to state a combinatorial analog of the Jacobian conjecture: *if f has the property that, for each $x \in \mathbb{B}^n$, all the boolean eigenvalues of $f'(x)$ are zero, then f has a unique fixed point.* This conjecture was proved by Shih and Dong (2005). Since Robert proved that all the boolean eigenvalues of $f'(x)$ are zero if and only if $Gf(x)$ has no cycle, the theorem of Shih and Dong can be stated as follows.

Theorem 2 (Shih and Dong (2005)) *If $Gf(x)$ has no cycle $\forall x \in \mathbb{B}^n$, then f has a unique fixed point.*

A short prove of this theorem, independent of Theorem 1, is given in appendix. In the following of this section, we show, using Theorem 1, that the condition *"if $Gf(x)$ has no cycle for all x"* can be weakened into a condition of the form *"if there exists "few" point x such that $Gf(x)$ has a "short" cycle".* The exact statement is given after the following proposition.

Proposition 1 *If f is even or odd, then for every $x \in \mathbb{B}^n$ the out-degree of each vertex of $Gf(x)$ is odd. In particular, $Gf(x)$ has a cycle.*

Proof: The out-degree d_j^+ of any vertex j of $Gf(x)$, which equals the number of ones in the jth column of $f'(x)$, is $d_j^+ = \|f(x^{j1}) \oplus f(x^{j0})\| = \|f(x) \oplus f(x \oplus e_j)\|$. Since

$$\|f(x) \oplus f(x \oplus e_j)\| = \|(x \oplus \tilde{f}(x)) \oplus ((x \oplus e_j) \oplus \tilde{f}(x \oplus e_j))\| = \|\tilde{f}(x) \oplus \tilde{f}(x \oplus e_j) \oplus e_j\|,$$

the parity of d_j^+ is the parity of $\|\tilde{f}(x)\| + \|\tilde{f}(x \oplus e_j)\| + 1$. Hence, if f is even or odd, then $\|\tilde{f}(x)\| + \|\tilde{f}(x \oplus e_j)\|$ is even, and d_j^+ is odd. $\qquad\qquad\square$

Corollary 2 (Extension of Shih-Dong's fixed point theorem) *If for $k = 1, \ldots, n$, there exists at most $2^k - 1$ points $x \in \mathbb{B}^n$ such that $Gf(x)$ has a cycle of length at most k, then the conjugate of f is a bijection. In particular, f has a unique fixed point.*

Proof: According to Theorem 1, it is sufficient to prove, by induction on n, that if f satisfies the conditions of the statement, then f has no even or odd self-dual subnetwork. The case $n = 1$ is obvious. Suppose that $n > 1$ and that f satisfies the conditions of the statement. Let $i, j \in [n-1]$. For each $x \in \mathbb{B}^n$ such that $x_n = 0$, we have

$$f_{ij}^{n0}(x_{-n}) = f_i^{n0}(x_{-n}^{j1}) \oplus f_i^{n0}(x_{-n}^{j0}) = f_i(x^{j1}) \oplus f_i(x^{j0}) = f_{ij}(x).$$

So $Gf^{n0}(x_{-n})$ is the subgraph of $Gf(x)$ induced by $[n-1]$, and we deduce that $f^{n\alpha}$ satisfies the condition of the theorem (for every $k \in [n-1]$, there exists at most $2^k - 1$ points $x \in \mathbb{B}^{n-1}$ such that $Gf^{n0}(x)$ has a cycle of length at most k). Thus, by induction hypothesis, f^{n0} has no even or odd self-dual subnetwork. More generally, we prove with similar arguments, that for all $i \in [n]$, f^{i0} and f^{i1} have no even or odd self-dual subnetwork. So f has no odd or even self-dual strict subnetwork. If f is itself even or odd self-dual, then by Proposition 1, $Gf(x)$ has a cycle for every $x \in \mathbb{B}^n$, so f does not satisfy that conditions of the statement (for $k = n$). Therefore, f has no even or odd self-dual subnetwork. □

Example 2 (Continuation of Example 1) *Take again*

$$f(x_1, x_2, x_3) = (\overline{x_2} \wedge x_3, \overline{x_3} \wedge x_1, \overline{x_1} \wedge x_2).$$

We have seen that f has no self-dual subnetwork. So it satisfies the conditions of Theorem 1, but not the conditions of Shih-Dong's theorem, since $Gf(000)$ and $Gf(111)$ have a cycle:

x	000	001	010	011	100	101	110	111

$Gf(x)$
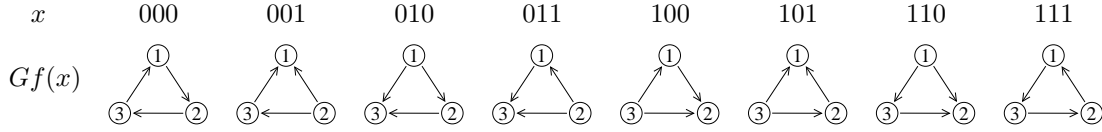
However, f satisfies the condition of Corollary 2 (there is $0 < 2^1$ point x with a cycle of length at most 1; $0 < 2^2$ point x such that $Gf(x)$ has a cycle of length at most 2, and $2 < 2^3$ points x such that $Gf(x)$ has a cycle of length at most 3).

Now, consider the following "extension" $h : \mathbb{B}^5 \to \mathbb{B}^5$ of f:

$$h(x_1, x_2, x_3, x_4, x_5) = (\overline{x_2} \wedge x_3, \overline{x_3} \wedge x_1, \overline{x_1} \wedge x_2, 0, 0) = (f(x_1, x_2, x_3), 0, 0)$$

Using the fact that f has no self-dual subnetwork, it's easy to see that h has no self-dual subnetwork. So h satisfies the conditions of Theorem 1. But it does not satisfy the conditions of Corollary 2. Indeed, there exists 2^3 points x such that $Gh(x)$ has a cycle of length at most 3:

x	00000	00001	00010	00011	11100	11101	11110	11111

$Gh(x)$
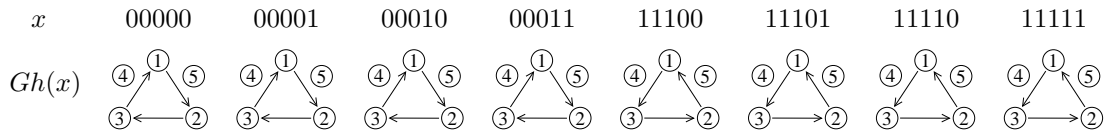
5   Remarks on asynchronous state graphs

In the following definition, we associate with $f : \mathbb{B}^n \to \mathbb{B}^n$ a directed graph on $\mathbb{B}^n$, called the *asynchronous state graph* of f, which has been proposed by Thomas (1973) as a model for the dynamics of gene networks; see also Thomas and d'Ari (1990).

Definition 7 (Asynchronous state graphs) *The* asynchronous state graph of f *is the directed graph $\Gamma(f)$ defined by: the vertex set is $\mathbb{B}^n$, and for every $x, y \in \mathbb{B}^n$, there exists an arc $x \to y$ if and only if there exists $i \in [n]$ such that $y = x \oplus e_i$ and $f_i(x) \neq x_i$.*

Remark that $\Gamma(f)$ and f share the same information. Remark also that for every $i \in [n]$ and $\alpha \in \mathbb{B}$, $\Gamma(f^{i\alpha})$ is isomorphic to the subgraph of $\Gamma(f)$ induced by the set of points $x \in \mathbb{B}^n$ such that $x_i = \alpha$. Indeed: for every $x, y \in \mathbb{B}^n$,

$$
\begin{aligned}
x_{-i} \to y_{-i} \text{ is an arc of } \Gamma(f^{i\alpha}) \quad &\Longleftrightarrow \quad \exists j \neq i \text{ such that } y_{-i} = x_{-i} \oplus e_j \text{ and } f_j^{i\alpha}(x_{-i}) \neq x_j \\
&\Longleftrightarrow \quad \exists j \neq i \text{ such that } y^{i\alpha} = x^{i\alpha} \oplus e_j \text{ and } f_j(x^{i\alpha}) \neq x_j \quad (\star) \\
&\Longleftrightarrow \quad x^{i\alpha} \to y^{i\alpha} \text{ is an arc of } \Gamma(f).
\end{aligned}
$$

Corollary 3 *If f has no even or odd self-dual subnetwork, then f has a unique fixed point x, and for all $y \in \mathbb{B}^n$, $\Gamma(f)$ contains a directed path from y to x of length $\|x \oplus y\|$.*

By the definition of $\Gamma(f)$, a path from x to y cannot be of length strictly less than $\|x \oplus y\|$; a path from x to y of length $\|x \oplus y\|$ can thus be seen has a *shortest path*.

Proof of Corollary 3: by induction on n. The case $n = 1$ is obvious, so suppose that $n > 1$ and that the corollary holds for the dimensions less than n. Let $f : \mathbb{B}^n \to \mathbb{B}^n$, and suppose that f has no even or odd self-dual subnetwork. By Theorem 1, f has a unique fixed point x. Let $y \in \mathbb{B}^n$. Suppose first that there exists $i \in [n]$ such that $x_i = y_i = 0$. Then x_{-i} is the unique fixed point of f^{i0}. So, by induction hypothesis, $\Gamma(f^{i0})$ has a path from y_{-i} to x_{-i} of length $\|x_{-i} \oplus y_{-i}\|$. Since $x_i = y_i = 0$, we deduce from $(\star)$ that $\Gamma(f)$ has a path from y to x of length $\|x_{-i} \oplus y_{-i}\| = \|x \oplus y\|$. The case $x_i = y_i = 1$ is similar. So, finally, suppose that $y = x \oplus 1$. Since y is not a fixed point, there exists $i \in [n]$ such that $f_i(y) \neq y_i$. Then, $\Gamma(f)$ has an arc from y to $z = y \oplus e_i$. So $z_i = x_i$, and as previously, we deduce that $\Gamma(f)$ has a path from z to x of length $\|x \oplus z\|$. This path together with the arc $y \to z$ forms a path from y to x of length $\|x \oplus z\| + 1 = \|x \oplus y\|$. □

According to $(\star)$, the asynchronous state graph of each subnetwork of f is a subgraph of asynchronous state graph of f induced by an hypercube contained in $\mathbb{B}^n$. Hence, one can see the asynchronous state graphs of the subnetworks of f as "*dynamical modules*" of asynchronous state graph of f. The previous corollary shows that if f has no even or odd self-dual subnetwork, then the asynchronous state graph of f is "simple": it describes a "weak convergence" toward a unique fixed point. An interpretation is then that **the asynchronous state graphs of even and odd self-dual networks are "dynamical modules" that are necessary for the "emergence" of "complex" asynchronous behaviors**.

Example 3 (Continuation of Example 1) *Take again the 3-dimensional network f defined in Example 1, which has no self-dual subnetwork. The asynchronous state graph $\Gamma(f)$ of f is the following:*

x	$f(x)$	$\tilde{f}(x)$
000	000	000
001	100	101
010	001	011
011	001	010
100	010	110
101	100	001
110	010	100
111	000	111
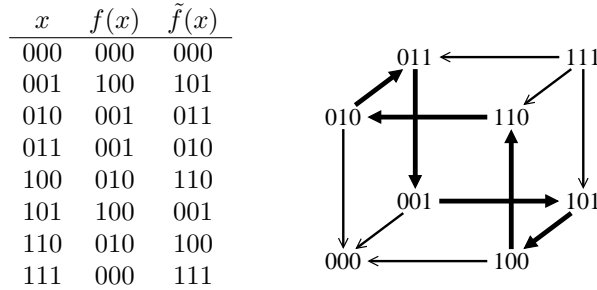
In agreement with Corollary 3, there exists, from any initial point, a shortest path leading to the unique fixed point of f (the point 000*): the asynchronous state graph describes a "weak asynchronous convergence" (by shortest paths) toward a unique fixed point. However, $\Gamma(f)$ has a cycle (of length 6), so every path does not lead to the unique fixed point: the condition "has no even or odd self-dual subnetworks" does no ensure a "strong asynchronous convergence" toward a unique fixed point.*

6   Remarks on positive and negative cycles

In this section, we show that positive (negative) circular networks, *i.e.* Boolean networks whose the global interaction graph reduces to a positive (negative) cycle, are simple instances of even (odd) circular networks. From this fact and existing results about positive and negative cycles, we will see that natural ideas of generalizations of Theorem 1 arise, but that none of these generalizations is true.

Let us begin with additional definitions. A *signed directed graph* is a directed graph in which each arc is either *positive*, *negative* or *unsigned*. In such a graph, a cycle is *positive* (*negative*) if it contains an unsigned arc or an even (odd) number of negative arcs (a directed cycle may be both positive and negative).

Definition 8 (Global interaction graph) *The* global interaction graph *of $f : \mathbb{B}^n \to \mathbb{B}^n$ is the signed directed graph $G(f)$ defined by: the vertex set is $[n]$, and for all $i, j \in [n]$, there exists an arc $i \to j$ if and only if $f_i(x^{j1}) \neq f_i(x^{j0})$ for at least one $x \in \mathbb{B}^n$; and an arc $j \to i$ of $G(f)$ is: positive if $f_i(x^{j1}) \geq f_i(x^{j0})$ for all $x \in \mathbb{B}^n$; negative if $f_i(x^{j1}) \leq f_i(x^{j0})$ for all $x \in \mathbb{B}^n$; and unsigned in the other cases.*

Remark that $G(f)$ has an arc $j \to i$ if and only if f_i depends on the jth variable x_j (and that $f_i(x^{j1}) \neq f_i(x^{j0})$ if and only if $f_{ij}(x) = 1$).

Definition 9 (Positive and negative circular networks) *f is a* positive *(*negative*) circular network if $G(f)$ is a positive (negative) cycle.*

The dynamics of positive and negative circular networks has been widely studied; see Remy et al. (2003) and Demongeot et al. (2010). Here, we prove that they are simple instances of even and odd self-dual networks.

Proposition 2 *Every positive (negative) circular network is even (odd) and self-dual.*

Proof: Let f be a circular network. Without loss of generality, suppose that the n arcs of $G(f)$ are $i + 1 \to i$ for all $i \in [n]$; $n + 1$ being identified to 1 (here and in the rest of the proof). Then f_i depends only on x_{i+1}, so either $f_i(x) = x_{i+1}$ (and $i + 1 \to i$ is positive), or $f_i(x) = x_{i+1} \oplus 1$ (and $i + 1 \to i$ is negative); in the first case, we set $s_i = 0$, and in the second case, we set $s_i = 1$ (so that $f_i(x) = x_{i+1} \oplus s_i$ in both cases). Let $s = (s_1, \ldots, s_n) \in \mathbb{B}^n$. By construction, f is positive if $\|s\|$ is even, and negative if $\|s\|$ is odd. Furthermore,

$$f(x) = (x_2, x_3, \ldots, x_n, x_1) \oplus s \qquad \forall x \in \mathbb{B}^n.$$

Hence

$$f(x \oplus 1) = (x_2 \oplus 1, \ldots, x_n \oplus 1, x_1 \oplus 1) \oplus s = (x_2, \ldots, x_n, x_1) \oplus 1 \oplus s = f(x) \oplus 1.$$

So f is self-dual. Also, we have $\tilde{f}(x) = x \oplus (x_2, \ldots, x_n, x_1) \oplus s$ so the parity of $\tilde{f}(x)$ is the parity of $\|x\| + \|(x_2, \ldots, x_n, x_1)\| + \|s\|$. Since $\|x\| = \|(x_2, \ldots, x_n, x_1)\|$, we deduce that the parity of $\tilde{f}(x)$ is the parity of $\|s\|$. So if f is positive (negative) then the image of $\tilde{f}$ only contains even (odd) points.

It remains to prove that if f is positive (negative) then each even (odd) point is in the image of $\tilde{f}$. Suppose that f is positive (negative), and let z be an even (odd) point of $\mathbb{B}^n$. Let $x \in \mathbb{B}^n$ be recursively defined by

$$x_1 = z_n, \qquad x_{i+1} = z_i \oplus s_i \oplus x_i \quad \text{for all } i \in [n-1].$$

Then, for every $i \in [n-1]$, we have

$$\tilde{f}_i(x) = x_i \oplus f_i(x) = x_i \oplus x_{i+1} \oplus s_i = x_i \oplus (z_i \oplus s_i \oplus x_i) \oplus s_i = z_i.$$

If remains to prove that $\tilde{f}_n(x) = z_n$. By the definition of x, we have

$$
\begin{aligned}
x_n &= (z_{n-1} \oplus s_{n-1}) \oplus x_{n-1} \\
&= (z_{n-1} \oplus s_{n-1}) \oplus (z_{n-2} \oplus s_{n-2}) \oplus x_{n-2} \\
&\vdots \\
&= (z_{n-1} \oplus s_{n-1}) \oplus (z_{n-2} \oplus s_{n-2}) \oplus \cdots \oplus (z_1 \oplus s_1) \oplus z_n \\
&= (z_1 \oplus z_2 \oplus \cdots \oplus z_n) \oplus (s_1 \oplus s_2 \oplus \cdots \oplus s_{n-1}).
\end{aligned}
$$

So z and $(s_1, s_2, \ldots, s_{n-1}, x_n)$ have the same parity, and since z and s have the same parity, we deduce that $x_n = s_n$. Thus $\tilde{f}_n(x) = x_n \oplus f_n(x) = s_n \oplus x_1 \oplus s_n = x_1 = z_n$, and we deduce that $\tilde{f}(x) = z$. So f is even (odd) self-dual.   □

Remark 1 *There are $2^{n-1}!$ n-dimensional even (odd) self-dual networks, but "only" $(n-1)!2^{n-1}$ n-dimensional positive (negative) circular networks. Since $2^{n-1}! = (n-1)!2^{n-1}$ for $n = 1, 2$, we deduce that every one or two-dimensional even (odd) self-dual network is a positive (negative) circular network.*

Since the class of positive and negative circular networks is contained in the class of even and odd self-dual networks, it is natural to think about the following generalization of Theorem 1: *if f has no positive or negative circular networks, then f has a unique fixed point.* However, this is false, as showed by the following example. Hence, **Theorem 1 becomes false if "has no even or odd self-dual subnetwork" is replaced by "has no positive or negative circular subnetwork".**

Example 4 $f : \mathbb{B}^4 \to \mathbb{B}^4$ *is defined by*

$$
\begin{aligned}
f_1(x) &= (\overline{x_2} \wedge x_3 \wedge \overline{x_4}) \vee ((\overline{x_2} \vee x_3) \wedge x_4) \\
f_2(x) &= (\overline{x_3} \wedge x_1 \wedge \overline{x_4}) \vee ((\overline{x_3} \vee x_1) \wedge x_4) \\
f_3(x) &= (\overline{x_1} \wedge x_2 \wedge \overline{x_4}) \vee ((\overline{x_1} \vee x_2) \wedge x_4) \\
f_4(x) &= (x_2 \wedge x_3 \wedge \overline{x_1}) \vee ((x_2 \vee x_3) \wedge x_1)
\end{aligned}
$$

The table of f and $\tilde{f}$, and the asynchronous state graph of f are as follow:

x	$f(x)$	$\tilde{f}(x)$
0000	0000	0000
0010	1000	1010
0100	0010	0110
0110	0011	0101
1000	0100	1100
1010	1001	0011
1100	0101	1001
1110	0001	1111
0001	1110	1111
0011	1010	1001
0101	0110	0011
0111	1011	1100
1001	1100	0101
1011	1101	0110
1101	0111	1010
1111	1111	0000

One can see that f is even self-dual. The immediate subnetworks of f are the following:
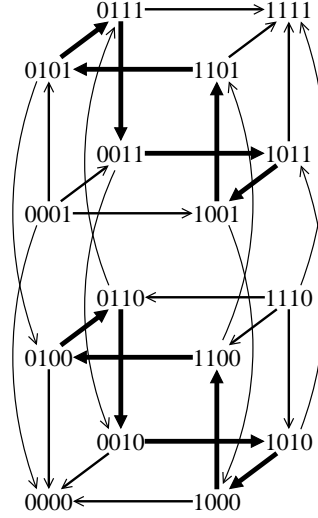
$$f^{10}(x_2, x_3, x_4) = (\overline{x_3} \wedge x_4, x_2 \vee x_4, x_2 \wedge x_3)$$
$$f^{11}(x_2, x_3, x_4) = (\overline{x_3} \vee x_4, x_2 \wedge x_4, x_2 \vee x_3)$$

$$f^{20}(x_1, x_3, x_4) = (x_3 \vee x_4, \overline{x_1} \wedge x_4, x_3 \wedge x_1)$$
$$f^{21}(x_1, x_3, x_4) = (x_3 \wedge x_4, \overline{x_1} \vee x_4, x_3 \vee x_1)$$

$$f^{30}(x_1, x_2, x_4) = (\overline{x_2} \wedge x_4, x_1 \vee x_4, x_2 \wedge x_1)$$
$$f^{31}(x_1, x_2, x_4) = (\overline{x_2} \vee x_4, x_1 \wedge x_4, x_2 \vee x_1)$$

$$f^{40}(x_1, x_2, x_3) = (\overline{x_2} \wedge x_3, \overline{x_3} \wedge x_1, \overline{x_1} \wedge x_2) \qquad \textit{(as in Examples 1-3)}$$
$$f^{41}(x_1, x_2, x_3) = (\overline{x_2} \vee x_3, \overline{x_3} \vee x_1, \overline{x_1} \vee x_2)$$

Proceeding as in Example 1, one can check that none immediate subnetwork of f has a self-dual subnetwork (actually, it is sufficient to check this for each f^{i0} since $f^{i1}(x) = f^{i0}(x \oplus 1) \oplus 1$, $1 \leq i \leq 4$). So f has no circular strict subnetwork, and since f is not circular, f has no circular subnetwork, but it has not a unique fixed points. Note that for $1 \leq i \leq 4$, the 4-dimensional network h defined by $h(x) = f(x) \oplus e_i$ is odd self-dual, has no circular subnetwork, and no fixed point.

Now, consider the following three fundamental theorems about cycles and fixed points (the last two theorems result from two conjectures of Thomas; see Remy et al. (2008); Richard (2010) and the references therein).

Theorem 3 (Robert (1980)) *If $G(f)$ has no cycle, then f has a unique fixed point.*

Remark 2 *Clearly, each local interaction graph $Gf(x)$ is a subgraph of the (unsigned version of the) global interaction graph $G(f)$. Hence, the condition "$G(f)$ has no cycle" of Robert's theorem is (much*

more) stronger than the condition "$Gf(x)$ has no cycle for every x" of Shih-Dong's Theorem. Consequently, Shih-Dong's theorem is a generalization of Robert's theorem. Thus, Theorem 1 is also a generalization of Robert's theorem.

Remark 3 *Actually, Robert proved, in Robert (1980) and Robert (1995), that if $G(f)$ has no cycle, then f has a unique fixed point x and: (1) the synchronous iteration $x^{t+1} = f(x^t)$ converges toward x in at most n steps for every initial point $x^0 \in \mathbb{B}^n$; (2) every path of $\Gamma(f)$ leads to x in at most n steps ("strong asynchronous convergence by shortest paths toward a unique fixed points"). These results shows the necessity of cycles for obtaining "complex" synchronous or asynchronous behaviors (e.g. multiple fixed points, cyclic attractors, long transient phases...).*

Theorem 4 (Remy et al. (2008)) *If $G(f)$ has no positive cycle, then f has at most one fixed point.*

Remark 4 *Actually, by saying that an arc $j \to i$ of $Gf(x)$ is positive if $f_i(x^{j1}) > f_i(x^{j0})$ and negative if $f_i(x^{j1}) < f_i(x^{j0})$, Remy et al. (2008) proved the following more general statement: if $Gf(x)$ has no positive cycle for all $x \in \mathbb{B}^n$, then f has at most one fixed point.*

Theorem 5 (Richard (2010)) *If $G(f)$ has no negative cycle, then f has at least one fixed point.*

Hence, Theorems 4 and 5 give a nice proof "by dichotomy" of Robert's theorem: the absence of positive cycle gives the uniqueness, and absence of negative cycle gives the existence. Seeing the relationship between positive (negative) circular networks and even (odd) self-dual networks, one may ask if a "proof by dichotomy" occurs for Theorem 1, *i.e.*, if the absence of even self-dual subnetwork gives the uniqueness, and if the absence of odd self-dual network gives the existence. The following example shows that both cases are false. Hence: **if f has no even (odd) self-dual subnetworks, than it has not necessarily at most (at least) one fixed point.**

Example 5 $f : \mathbb{B}^3 \to \mathbb{B}^3$ *is defined by*

$$f_1(x) = (\overline{x_1} \wedge (x_2 \vee x_3)) \vee (x_2 \wedge x_3)$$
$$f_2(x) = (\overline{x_2} \wedge (x_3 \vee x_1)) \vee (x_3 \wedge x_1)$$
$$f_3(x) = (\overline{x_3} \wedge (x_1 \vee x_2)) \vee (x_1 \wedge x_2)$$

The table of f and $\tilde{f}$, and the asynchronous state graph of f are as follow:

x	$f(x)$	$\tilde{f}(x)$
000	000	000
001	110	111
010	101	111
011	100	111
100	011	111
101	010	111
110	001	111
111	111	000

f is self-dual, but not even since $\|f(001)\|$ *is odd. The immediate subnetworks of f are:*

$$f^{10}(x_2, x_3) = (\overline{x_2} \wedge x_3, \overline{x_3} \wedge x_2)$$
$$f^{11}(x_2, x_3) = (\overline{x_2} \vee x_3, \overline{x_3} \vee x_2)$$

$$f^{20}(x_1, x_3) = (\overline{x_1} \wedge x_3, \overline{x_3} \wedge x_1)$$
$$f^{21}(x_1, x_3) = (\overline{x_1} \vee x_3, \overline{x_3} \vee x_1)$$

$$f^{30}(x_1, x_2) = (\overline{x_1} \wedge x_2, \overline{x_2} \wedge x_1)$$
$$f^{31}(x_1, x_2) = (\overline{x_1} \vee x_2, \overline{x_2} \vee x_1)$$

So each $f^{i\alpha}$ *is not circular, and according to Remark 1, it is not even and self-dual. Furthermore, each strict subnetwork h of* $f^{i\alpha}$ *is either constant or defined by* $h(0) = 1$ *and* $h(1) = 0$ *(in the second case, h is odd and self-dual). So* $f^{i\alpha}$ *has no strict even self-dual subnetwork. We deduce that f has no even self-dual subnetwork. But it has two fixed points.*

Now consider the network $f : \mathbb{B}^3 \to \mathbb{B}^3$ *is defined by*

$$f_1(x) = \overline{x_2}$$
$$f_2(x) = \overline{x_3}$$
$$f_3(x) = (x_3 \wedge (\overline{x_1} \vee x_2)) \vee (\overline{x_1} \wedge x_2)$$

The table of f and $\tilde{f}$, *and the asynchronous state graph of f are as follow:*

x	$f(x)$	$\tilde{f}(x)$
000	110	110
001	101	100
010	011	001
011	001	010
100	110	010
101	100	001
110	010	100
111	001	110

f is self-dual, but not odd since $\|f(000)\|$ *is even. The immediate subnetworks of f are:*

$$f^{10}(x_2, x_3) = (\overline{x_3}, x_3 \vee x_2)$$
$$f^{11}(x_2, x_3) = (\overline{x_3}, x_3 \wedge x_2)$$

$$f^{20}(x_1, x_3) = (1, x_3 \wedge \overline{x_1})$$
$$f^{21}(x_1, x_3) = (0, x_3 \vee \overline{x_1})$$

$$f^{30}(x_1, x_2) = (\overline{x_2}, 1)$$
$$f^{31}(x_1, x_2) = (\overline{x_2}, 0)$$

So each $f^{i\alpha}$ *is not circular, and according to Remark 1, it is not odd and self-dual. Furthermore, each strict subnetwork h of* $f^{i\alpha}$ *is either constant or defined by* $h(0) = 0$ *and* $h(1) = 1$ *(in the second case, h is even and self-dual). So* $f^{i\alpha}$ *has no strict odd self-dual subnetwork. We deduce that f has no odd self-dual subnetwork. But it has no fixed point.*

Acknowledgements

A A short proof of the theorem of Shih and Dong

The "trick" consists in proving, by induction on n, the following more general statement:

> ($*$) If $Gf(x)$ has no cycle for all $x \in \mathbb{B}^n$, then the conjugate of f is a bijection (so that f has a unique fixed point).

The case $n = 1$ is obvious, so suppose that $n > 1$ and that ($*$) holds for the dimensions less than n. Suppose that $Gf(x)$ has no cycle for all $x \in \mathbb{B}^n$. Let $i, j \in [n-1]$, and $x \in \mathbb{B}^n$ such that $x_n = 0$. We have

$$f_{ij}^{n0}(x_{-n}) = f_i^{n0}(x_{-n}^{j1}) \oplus f_i^{n0}(x_{-n}^{j0}) = f_i(x^{j1}) \oplus f_i(x^{j0}) = f_{ij}(x).$$

So $Gf^{n0}(x_{-n})$ is the subgraph of $Gf(x)$ induced by $[n-1]$, and thus, it has no cycle. We deduce that f^{n0} satisfies the conditions of ($*$). Thus, by induction hypothesis, the conjugate of f^{n0} is a bijection. We prove with similar arguments that $\tilde{f}^{i0}$ and $\tilde{f}^{i1}$ are bijections for all $i \in [n]$.

Now, suppose, by contradiction, that $\tilde{f}$ is not a bijection. Then, there exists two distinct points $x, y \in \mathbb{B}^n$ such that $\tilde{f}(x) = \tilde{f}(y)$. Let us proved that $x = y \oplus 1$. Indeed, if $x_i = y_i = \alpha$ for some $i \in [n]$, then $\tilde{f}^{i\alpha}(x_{-i}) = \tilde{f}(x)_{-i} = \tilde{f}(y)_{-i} = \tilde{f}^{i\alpha}(y_{-i})$. Thus $\tilde{f}^{i\alpha}$ is not a bijection, a contradiction. So $x = y \oplus 1$. Since $Gf(x)$ has no cycle, it contains at least one vertex of out-degree 0. In other words, there exists $i \in [n]$ such that $f(x^{i1}) = f(x^{i0})$. Thus $\tilde{f}(x^{i1})_{-i} = \tilde{f}(x^{i0})_{-i} = \tilde{f}(x)_{-i}$. Hence, setting $\alpha = y_i$, we obtain

$$\tilde{f}^{i\alpha}(x_{-i}) = \tilde{f}(x^{i\alpha})_{-i} = \tilde{f}(x)_{-i} = \tilde{f}(y)_{-i} = \tilde{f}(y^{i1})_{-i} = \tilde{f}^{i1}(y_{-i}).$$

So $\tilde{f}^{i\alpha}$ is not a bijection, a contradiction. Thus $\tilde{f}$ is a bijection and ($*$) is proved.

References

J. Demongeot, M. Noual, and S. Sené. On the number of attractors of positive and negative Boolean automata circuits. In *Proceedings of WAINA'10*, pages 782–789. IEEE press, 2010.

S. A. Kauffman. Metabolic stability and epigenesis in randomly connected nets. *Journal of Theoretical Biology*, 22:437–467, 1969.

E. Remy, B. Mossé, C. Chaouiya, and D. Thieffry. A description of dynamical graphs associated to elementary regulatory circuits. *Bioinformatics*, 19:172–178, 2003.

E. Remy, P. Ruet, and D. Thieffry. Graphic requirements for multistability and attractive cycles in a boolean dynamical framework. *Advances in Applied Mathematics*, 41(3):335 – 350, 2008. ISSN 0196-8858.

A. Richard. Negative circuits and sustained oscillations in asynchronous automata networks. *Advances in Applied Mathematics*, 44(4):378 – 392, 2010. ISSN 0196-8858.

F. Robert. Iterations sur des ensembles finis et automates cellulaires contractants. *Linear Algebra and its Applications*, 29:393–412, 1980.

F. Robert. Dérivée discrete et convergence locale d'une itération Booléenne. *Linear Algebra Appl.*, 52: 547–589, 1983.

F. Robert. *Les systèmes dynamiques discrets*, volume 19 of *Mathématiques et Applications*. Springer, 1995.

M.-H. Shih and J.-L. Dong. A combinatorial analogue of the Jacobian problem in automata networks. *Advances in Applied Mathematics*, 34:30–46, 2005.

M.-H. Shih and J.-L. Ho. Solution of the Boolean Markus-Yamabe problem. *Advances in Applied Mathematics*, 22:60–102, 1999.

R. Thomas. Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, 42(3):563 – 585, 1973. ISSN 0022-5193.

R. Thomas and R. d'Ari. *Biological Feedback*. CRC Press, 1990.

Characterization of non-uniform number conserving cellular automata

Sukanta Das[†]

Department of Information Technology, Bengal Engineering and Science University, Shibpur, India

This paper characterizes the one dimensional two-state 3-neighborhood non-uniform (hybrid) number conserving cellular automata (NCCA). The reachability tree is utilized to do such characterization. The paper has developed a set of theorems targeting the characterization of the NCCA. An algorithm of $O(n)$ time is developed to verify whether CA with n cells are NCCA. Finally, another algorithm is designed that synthesizes NCCA with given number of cells.

Keywords: Number conserving cellular automata (NCCA), hybrid CA, rule min term (RMT), reachability tree

I   Introduction

The number conserving cellular automata (NCCA) in one dimensional two-state 3-neighborhood dependency are the cellular automata (CA) where the number of 1s (0s) of initial configuration is preserved during the evolution of the CA. Due to their similarity with the physical law of conservation, the NCCA have received a wide attention of the researchers in last two decades [HT91, BF98, BF02, DFR03]. The major application area of NCCA is the development of highway traffic models [NS92, FI96, DSS09, Das11].

A few of the pioneering works are due to Boccara and Fukś who gave necessary and sufficient conditions for one-dimensional CA to be NCCA [BF98, BF02]. The computational universality, decidability, reversibility and other properties of NCCA also have been studied [DFR03, MI98]. However, all the works focus on uniform cellular automata, where all the cells are assumed to obey same transition rule. The characterization of non-uniform or hybrid NCCA is not addressed till date. In non-uniform or hybrid NCCA, different cells may follow different rules. This work targets such characterization.

To identify the characteristics of non-uniform NCCA, we utilize the *reachability tree* which was proposed as a tool for characterizing CA [DS09]. We present a set of theorems and corollaries to characterize reachability tree for NCCA. Based on such characterization, we develop a linear time algorithm to verify whether given CA are NCCA. Finally, an algorithm to synthesize NCCA is reported.

The paper is organized as follows. The preliminaries of CA and reachability tree are noted in the next section. Section III characterizes the reachability tree for NCCA. Finally, the algorithms are presented in Section IV. Section V concludes the paper.

Tab. 1: Look-up table for rule 184 and 226

Present state :	111	110	101	100	011	010	001	000	*Rule*
(RMT)	(7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	
(i) Next State :	1	0	1	1	1	0	0	0	184
(ii) Next State :	1	1	1	0	0	0	1	0	226

II Cellular automata and reachability tree

The cellular automata (CA) are the discrete spatially-extended dynamical systems that have been studied extensively as models of physical systems. They evolve in discrete space and time. In their simplest form, CA consist of a lattice of cells, each of which stores a discrete variable at time t that refers to the present state of the CA cell [vN66]. The next state of a cell is affected by its present state and the present states of its neighbors at time t. In two-state 3-neighborhood (self, left and right neighbors) 1-dimensional CA, next state of a cell is determined as:

$$S_i^{t+1} = f_i(S_{i-1}^t, S_i^t, S_{i+1}^t) \tag{1}$$

where f_i is the next state function of ith cell; S_{i-1}^t, S_i^t and S_{i+1}^t are the present states of the left neighbor, self and right neighbor of the ith CA cell at time t. Therefore, the $f_i : \{0,1\}^3 \mapsto \{0,1\}$ can be expressed as a look-up table. The decimal equivalent of the 8 outputs is called 'rule' [Wol86]. Two such rules are 184 and 226 (Tab. 1). The CA are *uniform* if all the CA cells follow same rule; otherwise they are *non-uniform/ hybrid*. In case of hybrid CA, we need a *rule vector* $\mathcal{R} = \langle \mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_i, \cdots, \mathcal{R}_n \rangle$, where $\mathcal{R}_i$ configures CA cell i ($1 \leq i \leq n$). If the left most and right most cells are the neighbors of each other, the CA are *periodic boundary* CA.

The collection of states of the cells $\mathcal{S}^t = (S_1^t, S_2^t, \cdots, S_n^t)$ at time t is the present configuration or state of CA. Therefore, the next state of CA with n cells is determined as:

$$\mathcal{S}^{t+1} = (f_1(S_n^t, S_1^t, S_2^t), f_2(S_1^t, S_2^t, S_3^t), \cdots, f_n(S_{n-1}^t, S_n^t, S_1^t)) \tag{2}$$

In case of number conserving cellular automata (NCCA), for each pair of $\mathcal{S}^t$ and $\mathcal{S}^{t+1}$, the number of 0s and 1s in $\mathcal{S}^t$ remain unchanged in $\mathcal{S}^{t+1}$. The present work concentrates on the characterization of hybrid NCCA with periodic boundary condition.

Rule Min Term (RMT): From the view point of *Switching Theory*, a combination of the present states (as noted in the 1st row of Tab. 1) can be viewed as the *Min Term* of a 3-variable $(S_{i-1}^t, S_i^t, S_{i+1}^t)$ switching function. Therefore, each column of the first row of Tab. 1 is referred to as *Rule Min Term* (RMT). The RMTs have binary values (0/1) which correspond to the next states for these RMTs. For example, the RMT 011 (RMT 3) in Tab. 1 has the value 1 for rule 184 and 0 for rule 226. The characterization reported in the following section is based on the analysis of RMTs of the CA rules.

Reachability tree

The reachability tree, we proposed in [DSC04, DS06, DS09, DS10], is a binary tree that represents the reachable states of CA. A state is reachable if it has at least one predecessor. That is, the reachable state

Tab. 2: Relationship among RMTs of rules for cell i and $(i+1)$ for next state computation

RMT at i^{th} rule	RMTs at $(i+1)^{th}$ rule
0	0, 1
1	2, 3
2	4, 5
3	6, 7
4	0, 1
5	2, 3
6	4, 5
7	6, 7

is derived from some other state of the CA. Each node of the tree is constructed with RMT(s) of a rule. The left edge of a node is referred to as the 0-edge and the right edge is as 1-edge (Fig. 1). The number of levels in a reachability tree, for n-cell CA, is $(n+1)$. The root node is at level 0 and the leaves are at level n. The nodes at level i are constructed from the RMTs of $(i+1)^{th}$ CA cell rule $\mathcal{R}_{i+1}$. The number of leaves in the reachability tree denotes the number of reachable states of the CA. A sequence of edges from the root to a leaf node, representing an n-bit binary string, is the reachable state, where the 0-edge represents 0 and 1-edge represents 1.

Since the CA are in 3-neighborhood dependency, an RMT can be considered as a 3-bit window. To get the next state of a given CA state, we consider that the 3-bit window slides 1-bit right in each step over the given state. Here, the window for i^{th} cell contains $b_{i-1}b_ib_{i+1}$ ($b_i = 0/1$), where b_i is the ith bit of present state. Now to get the next state for i^{th} cell, RMT $b_{i-1}b_ib_{i+1}$ of $\mathcal{R}_i$ is to be considered. If, for example, a window contains 101 at i^{th} cell, the next state is determined by RMT 5 of $\mathcal{R}_i$. Now, while the ith cell is being processed, then the content of window for $(i+1)^{th}$ cell can be predicted. The content is either $(b_ib_{i+1}0)$ or $(b_ib_{i+1}1)$. In other words, if the ith CA cell changes its state following the RMT k (decimal equivalent of $b_{i-1}b_ib_{i+1}$) of rule $\mathcal{R}_i$, then the $(i+1)^{th}$ cell can generate the next state based on the RMT $2k \bmod 8$ ($b_ib_{i+1}0$) or $(2k+1) \bmod 8$ ($b_ib_{i+1}1$) of rule $\mathcal{R}_{i+1}$. This actually shows that the RMTs of two consecutive cells are related. All such relationships between the RMTs of $\mathcal{R}_i$ and $\mathcal{R}_{i+1}$, while computing next state of CA, is shown in Tab. 2. The reachability tree for some CA is generated based on such relationship. Before proceeding further, we define the following.

Definition 1 *Two RMTs of a rule $\mathcal{R}_i$ are **sibling** of each other, if these are resulted from the same RMT of $\mathcal{R}_{i-1}$. Two sibling RMTs differ only in the right most bit.*

The RMTs 0 and 1 of $\mathcal{R}_i$ are the sibling RMTs as these two are resulted in either from RMT 0 or from RMT 4 of $\mathcal{R}_{i-1}$ (Tab. 2). These sibling RMTs are associated with a single node of the reachability tree. Therefore, if a node of reachability tree associates an RMT k, it also associates the sibling of k.

Definition 2 *Two RMTs of a rule $\mathcal{R}_i$ are **equivalent** if they produce two same RMTs for $\mathcal{R}_{i+1}$. The equivalent RMTs differ only in the left most bit.*

The RMTs 0 and 4 of $\mathcal{R}_i$ are equivalent as they both produce RMTs 0 and 1 for the next rule (Tab. 2). Similarly, RMTs 1 & 5, 2 & 6, and 3 & 7 are equivalent.

Tab. 3: Binary values of the CA ⟨136, 252, 238, 192⟩ cell rules

RMT	111	110	101	100	011	010	001	000	Rule
	(7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	
First cell	1	0	0	0	1	0	0	0	136
Second cell	1	1	1	1	1	1	0	0	252
Third cell	1	1	1	0	1	1	1	0	238
Fourth cell	1	1	0	0	0	0	0	0	192
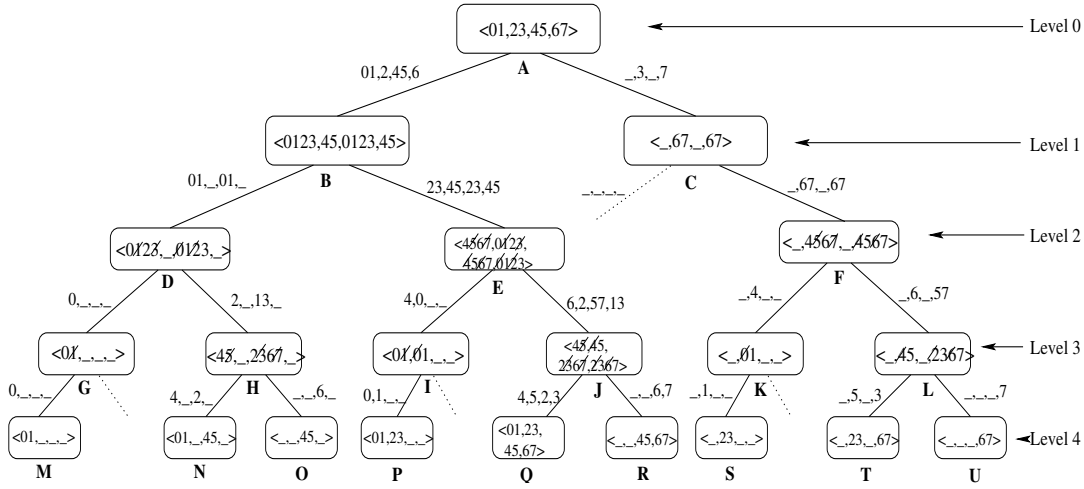
Fig. 1: Reachability Tree for the CA ⟨136, 252, 238, 192⟩

Consider the CA with rule vector ⟨136, 252, 238, 192⟩. The RMTs of CA rules are noted in Tab. 3. The reachability tree for the CA is shown in Fig. 1. The decimal digits within a node of the tree, at level i, represent the RMTs of the CA cell rule $\mathcal{R}_{i+1}$. The cell $(i + 1)$ changes its state depending upon those RMT values for $\mathcal{R}_{i+1}$. For example, the root node (level 0) is constructed with all the 8 RMTs – 0, 1, 2, 3, 4, 5, 6 and 7 of $\mathcal{R}_1$. The first cell changes its state according to the values of these RMTs. In the tree, the sibling RMTs (Definition 1) of root are grouped into four sets – {0, 1}, {2, 3}, {4, 5} and {6, 7}. To simplify the presentation, the sets are noted in the root as ⟨01, 23, 45, 67⟩ (Fig. 1). The RMTs (of a rule) for which we follow an edge (0-edge or 1-edge) are noted above the edge.

An RMT of a rule is a member of ith set, implies that the RMT is derived (following Tab. 2) from set i of the root ($0 \le i \le 3$ and set 0 is {0, 1}, set 1 is {2, 3}, set 2 is {4, 5} and {6, 7} is the set 3). If no RMT is the member of a set (that is, the set is empty), the set is noted as '₋'. This grouping of RMTs is required for the characterization of periodic boundary CA [DS09].

The RMTs 3 (set 1) and 7 (set 3) of 136 are 1 and rest are 0 (Tab. 3). So, the edges from the root are labeled with 01,2,45,6 (for 0-edge) and ₋,3,₋,7 (for 1-edge) accordingly. Here, '₋' indicates the empty set. Therefore, the nodes B and C (Fig. 1) are constructed, following the respective edges and Tab. 2,

with $\langle 0123, 45, 0123, 45 \rangle$ and $\langle \_, 67, \_, 67 \rangle$ respectively. However, there is only a single edge from node C which derives the child node F. The dotted edge indicates that no RMT of node C (for rule 252) can derive its 0-edge. Hence, the CA states begin with 10 are non-reachable. There are 9 leaf nodes of Fig. 1, so the CA have 9 reachable states.

A number of RMTs are dropped from the nodes at level $(n-2)$ (level 2 of Fig. 1) and level $(n-1)$ – that is, level 3 of Fig. 1. The RMTs of the nodes at level $(n-2)$ correspond to the CA cell rule $\mathcal{R}_{n-1}$. The RMTs of set 0 and set 1 assume that the cell n is always 0 while we compute the next state, whereas the RMTs of set 2 and set 3 assume that the cell n is always 1. Therefore, odd RMTs of set 0 and set 1, and even RMTs of set 2 and set 3 are invalid, and so striked out. For example, RMTs 1 and 3 from set 0, and RMTs 0 and 2 from set 2 in node D are striked out. Similarly, the RMTs of the nodes at level $(n-1)$ correspond to the CA cell rule $\mathcal{R}_n$. Therefore, the RMTs of set 0 for $\mathcal{R}_n$ (at level $(n-1)$) have to generate the set 0 for $\mathcal{R}_1$, since next to the last cell is the first cell. The set 0 for the first cell contains always RMTs 0 and 1. However, few RMTs of set 0 at level $(n-1)$ may not generate RMT 0 and 1 for $\mathcal{R}_1$, these are marked as invalid, and striked out. Similar actions are taken for other sets. In node G (Fig. 1), RMT 1 of set 0 is striked out as it can not generate set 0 for $\mathcal{R}_1$ ($\{0, 1\}$).

We next characterize such reachability tree to get characterization of NCCA.

III Characterization of reachability tree for NCCA

This section characterizes the reachability tree that represents number conserving cellular automata (NCCA). We identify here the required properties of the tree so that the corresponding CA can be NCCA. To facilitate our further discussion, we define the following.

Definition 3 *A sequence of n RMTs those derive a reachable state of CA with n cells is called the RMT sequence (RS).*

For example, $\langle 4012 \rangle$ is an RMT sequence (RS). In Fig. 1, this RS derives the state 0010. RMT 4 of rule 136 is associated with 0-edge from the root. Similarly, RMTs 0, 1 and 4 of the rules 252, 238 and 192 are associated with 0-, 1- and 0-edges from nodes B, D and H respectively. 0001 is the previous state of 0010. It can be noted that the middle bits of the RMTs 4, 0, 1 and 2 are 0, 0, 0 and 1. Hence, the RS $\langle 4012 \rangle$ corresponds to the state 0001, and derives the state 0010. In the reachability tree, the reachable states are associated with corresponding RSs. However, one reachable state may be derived from two or more RSs. For example, state 0010 of Fig. 1 is derived from $\langle 4012 \rangle$ as well as from $\langle 0124 \rangle$.

Theorem 1 : *The CA are NCCA if and only if the number of 1s (0s) of each RS remains unchanged in the reachable state derived from the RS.*

Proof: The pair of an RS and its derived state forms actually a pair of present state and next state (or, previous state and present state) of CA. To be NCCA, the number of 1s (0s) in a present state of each such pair has to be equal with that of the next state. Hence the proof. □

Following result can be derived from the Theorem 1.

Corollary 1 : *All-0 and all-1 states in NCCA are reachable and derived only from RS $\langle 00 \cdots 0 \rangle$ and $\langle 77 \cdots 7 \rangle$ respectively.*

Proof: Since the number of 0s and 1s are preserved in NCCA, the states $00 \cdots 0$ and $11 \cdots 1$, two special states, can not have any predecessor other than itself. So, these all-0 and all-1 states are reachable and the RSs $\langle 00 \cdots 0 \rangle$ and $\langle 77 \cdots 7 \rangle$ can only derive them. $\qquad \square$

Since the RSs $\langle 00 \cdots 0 \rangle$ and $\langle 77 \cdots 7 \rangle$ derive the states $00 \cdots 0$ and $11 \cdots 1$, RMT 0 and RMT 7 of each of the rules of NCCA are to be 0 and 1 respectively. Now, to verify whether the CA are NCCA, one has to concentrate only on the reachable states and their RSs. For such verification, we form the reachability tree for the given CA and assign a weight to each of the RMTs of a node. The assignment of weight to the RMTs are based on the following rule:

1. The weights of RMTs at root are 0.

2. Suppose, r_i is an RMT of a node at level i with weight w_i, and it derives RMT r_{i+1} (following Tab. 2) of another node at level $i + 1$. If the middle bit of 3-bit RMT is 1 and r_i to r_{i+1} follows 0-edge, then $w_{i+1} = w_i + 1$; if middle bit is 0 but r_i to r_{i+1} follows 1-edge, then $w_{i+1} = w_i - 1$; otherwise $w_{i+1} = w_i$.

It is obvious that the weights in the above rule indicate the surplus or deficiency of 1s in the RMTs of an RS compared to the corresponding reachable state. If an RMT of a node at level i has weight w_i, this means, the ith RMT of the corresponding RS carries w_i number of 1s as the surplus compared to the 1s generated by the previous RMTs of the RS. For example, the RMTs of RS $\langle 4012 \rangle$ that derives the state 0010 have the weights 0, 0, 0 and -1 respectively. The weight -1 is nullified at leaf node (node N in Fig. 2) as RMT 2 is 0 here. Therefore, the RMTs of the leaves in NCCA have zero weight. If it is found in a reachability tree that any RMT at some leaf node has non-zero weight, the CA are not NCCA; otherwise they are NCCA.

Example 1 *Consider the CA with rule vector $\langle 136, 252, 238, 192 \rangle$ (Tab. 3). We assign the weights to the RMTs according to the above rule. The reachability tree with such weight is noted in Fig. 2. The weights are shown in the bottom of the RMTs (within first brackets). Many of the RMTs of intermediate nodes has non-zero weights. However, the leaves of the tree have RMTs with zero weight. Hence, the CA are NCCA.*

Theorem 2 : *Equivalent RMTs with same next state value of a rule of NCCA carry same weight if they belong to a single set in a node.*

Proof: Equivalent RMTs, such as 0 and 4, derive same set of RMTs for the next level in reachability tree (Tab. 2). If they have same next state value (either 0 or 1), they follow the same edge, either 0- or 1-edge. Now, if a set of equivalent RMTs have different weights and they are together in a single set of a node, an RMT derived from the equivalent RMTs for the next level have different weights. This difference in weights of a single RMT is carried up to the leaves. Finally, one or more leaves can be found with different weights. To be NCCA, all the RMTs at leaves have to have weight 0. So, the CA are not NCCA. Hence, the weights of equivalent RMTs are to be same. $\qquad \square$

Following corollary can be derived from Theorem 2.

Corollary 2 : *The weight of an RMT at a node of reachability tree of NCCA is unique.*
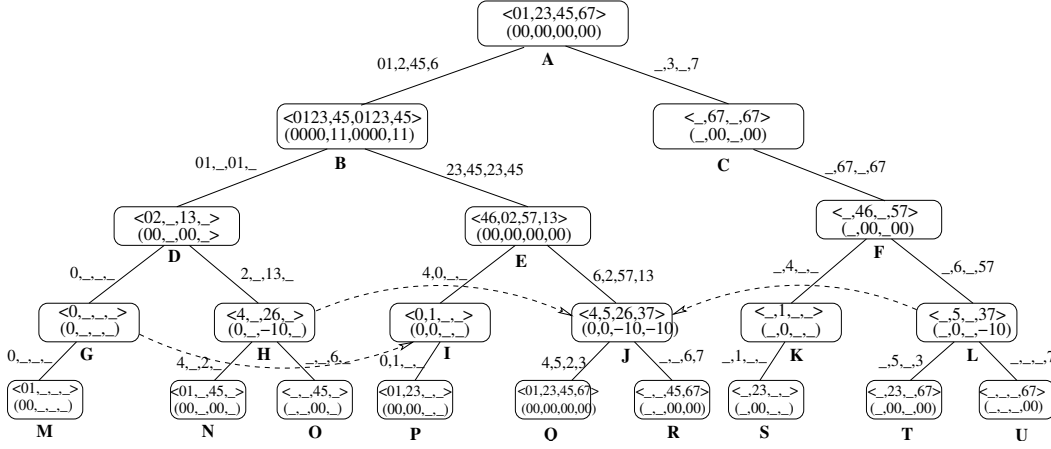
Fig. 2: Reachability tree of NCCA with weights of RMTs at different nodes

Proof: Since the RMTs at root have unique weights, there is only a way to get different weights for an RMT at a node – the RMT is derived from two RMTs (equivalent RMTs) with different weights at the node's parent node. However, those equivalent RMTs can not have different weights while the CA are NCCA (Theorem 2). Hence, the weights of RMTs are to be unique for NCCA. □

There are 2^i number of nodes at i^{th} level of reachability tree. However, all the nodes may not be unique. A number of nodes may be represented by some other nodes of that level. To characterize such nodes, we define the following.

Definition 4 *A node $\mathcal{N}_i$ is sub-node of another node $\mathcal{N}_j$ if $\mathcal{N}_i$ and $\mathcal{N}_j$ are at same level and each of the 4 sets of $\mathcal{N}_i$ is the subset of that of $\mathcal{N}_j$. We write $\mathcal{N}_i \subseteq \mathcal{N}_j$.*

In Fig. 2, node G is the sub-node of I ($G \subseteq I$). Similarly, $H \subseteq J$ and $L \subseteq J$. To find the relationship between a node and its sub-node(s) in NCCA, we next state the following theorem.

Theorem 3 : *An RMT that belongs to the same set of two nodes at some level of the reachability tree for NCCA has the same weight.*

Proof: Consider an RMT r of some rule belongs to any one of the four sets of two nodes, N_i and N_j at some level of the reachability tree. For a proof by contradiction, assume that r has two different weights in two nodes. Hence, the RMTs derived from r (following Tab. 2) for the next levels have different weights. As a result, at most one set of leaf nodes derived from N_i or N_j can have RMTs with weight 0. The RMTs with non-zero weight in any leaf node indicates the CA are not NCCA. Hence to be NCCA, the RMT r has to have same weight in two nodes. □

Following corollary directly follows from Theorem 3. This corollary establishes the relation between a node and its sub-node in terms of carry of 1s.

Corollary 3 : *The RMTs of a sub-node of a node in NCCA have the same weight with the respective RMTs of the node.*

Therefore, during the verification of CA for NCCA, we can remove the sub-nodes from a level of reachability tree. If the remaining nodes can derive the leaves where the RMTs have zero weight, then obviously the CA are NCCA. However, the CA are declared as non-NCCA if Theorem 3 is not followed.

Following corollary can also be derived from Theorem 3.

Corollary 4 : *The weight of an RMT in the reachability tree for NCCA can vary from -2 to 2.*

Proof: An RMT r of a node at level i produces two sibling RMTs (Tab. 2) for the next level in reachability tree. The sibling RMTs are at same node with same weight. These two RMTs also produce other RMTs for the lower layers. If the siblings of r have different next state values, then only the weights of produced RMTs can vary. Obviously, these newly produced RMTs are in different nodes. However, the RMT r is returned back in two nodes at level $i + 4$. For example, the productions of RMT 0 at level i are the following: $0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0$ and $0 \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 0$ (Tab. 2). Here, RMT 0 is returned back after 4th level. Now, to be NCCA, the weight of r at two nodes are to be same (Theorem 3). Suppose, the weight of r at level i was w. So, minimum and maximum weights of intermediate RMTs that are produced from r can be $w - 2$ and $w + 2$. At root ($i = 0$), the weights of RMTs are 0. So, minimum and maximum weights of RMTs in the nodes between level 0 and level 4 are -2 and 2 respectively. The same is true for the RMTs in the nodes of other levels. Hence the proof.   □

Observation 4 *The maximum difference in weight between two RMTs at some level of reachability tree for NCCA that are produced from same RMT at some upper level is 1.*

Based on the above characterization of reachability tree for NCCA, we next present two algorithms – the first one is to verify whether given CA are NCCA and the second one synthesizes NCCA for a given size.

IV Algorithms for NCCA

The reported algorithms are designed considering the reachability tree for NCCA. While we design the verification algorithm, reachability tree for NCCA are virtually generated to apply the above characterization on the tree. On the other hand, the synthesis algorithm generates reachability tree (hence the CA) in such a way that the mentioned properties of the tree for NCCA are maintained.

IV.1 Verification

The verification algorithm takes the CA rule vector as input. Reachability tree for the given CA is virtually constructed. At each level, the nodes of the tree are generated and it is checked whether the nodes obey Theorem 2 and Theorem 3. If any one of the nodes disobeys, the CA are reported as not NCCA. We follow Corollary 3 to reduce the number of nodes at each level. As a result, the number of nodes never becomes exponential. Finally, the algorithm reports that the CA are NCCA if each of the leaves has RMTs with zero weight. Following is the algorithm.

Algorithm 1 *VerifyNCCA*
Input: *CA with rule vector* $\mathcal{R} = \langle \mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_n \rangle$
Output: *'No', if CA are not NCCA; 'Yes' otherwise.*
Step 1: *Form root of reachability tree. Assign weights for RMTs at root as 0.*
Step 2: *For $i = 1$ to n repeat* Step 3 *to* Step 6.

Step 3: *Get the nodes of ith level depending on the nodes of $(i-1)^{th}$ level and rule $\mathcal{R}_i$.*
Step 4: *Remove invalid RMTs from the nodes at level $i = n-2$ and $i = n-1$.*
Step 5: *If any node disobeys Theorem 2 or Theorem 3, output 'No' and return.*
Step 6: *Identify and remove sub-nodes while Corollary 3 is obeyed.*
Step 7: *If there is any (leaf) node with non-zero weight, output 'No'; otherwise, output 'Yes'.*

Complexity: The time requirement of Algorithm 1 depends on n, the size of CA (Step 2) and the maximum number of nodes in the reachability tree (Steps 5 & 6). Since the nodes are formed with only 8 RMTs and the sub-nodes are removed in each level of the tree, the maximum number of nodes for NCCA with an arbitrary number of cells remains finite. Hence, the time complexity of Algorithm 1 is $O(n)$.

Observation 5 *Maximum number of nodes that represent all the sub-nodes in a level of reachability tree for NCCA is 7.*

Example 2 *Let consider the input to Algorithm 1 is the rule vector $\langle 136, 252, 238, 192 \rangle$ (Tab. 3). The root of the reachability tree is formed with 8 RMTs, and the weights for those RMTs are also set as 0 (Step 1). Since $\mathcal{R}_1 = 136$, RMTs 3 and 7 derive 1-edge (as they are 1) and the rest derive 0-edge. Two nodes are formed at level 1. The weights for RMTs of the nodes are also assigned. Here, no node disobeys Theorem 2 and Theorem 3. Since there is no sub-node (Step 6), the next level of nodes are formed based on $\mathcal{R}_2 = 252$. In level 2 also, no sub-node is found. However, in level 3, two sub-nodes are found and Theorem 3 is satisfied. Therefore, the sub-nodes can be removed (Step 6). Finally, we get the leaves having RMTs with zero weight. Hence, the output is 'Yes'. The reachability tree for the CA is noted in Fig. 2.*

IV.2 Synthesis

Synthesis is the reverse process of verification. In the synthesis algorithm, input is the number of cells (n) and output is a rule vector for NCCA. The algorithm is designed in such a way that Theorem 2 and Theorem 3 are followed in each step. Moreover, the following properties, obtained from Theorem 1, are to be satisfied in the reachability tree of NCCA.

1. RMT 0 at any set of a node can have weight either 0 or 1.

2. RMT 7 at any set of a node can have weight either 0 or -1.

Reason for the first property: RMT 0 for all the rules of NCCA are 0. RMT 0 produces again RMT 0 for lower layer ($0 \rightarrow 0 \rightarrow \cdots$). So, if RMT 0 at any node of the reachability tree is received any weight, the weight is carried by the successive RMT 0. This weight can only be nullified by the nodes at level $n-1$ and level n. Now, RMT 0 can appear in any of the 4 sets of a node. Consider for example, RMT 0 is generated at set 1 of a node with weight w. It can be noted that RMT 0 can receive non-zero weight in the reachability tree if it is generated from RMT 4. Now, according to the formation of reachability tree, the productions of RMTs at levels $n-2$ and $n-1$ are: $\cdots 0 \rightarrow 0(level\ n-2) \rightarrow 1(level\ n-1) \rightarrow 2, 3(level\ n)$. To be NCCA, weights of RMTs 2 and 3 at level n are to be 0. Hence, if RMT 1 at level $n-1$ is 0 due to the rule $\mathcal{R}_{n-1}$, w is to be 0; otherwise it is to be 1. So, RMT 0 of any node can have weight either 0 or 1. Similarly consider, RMT 0 is generated at set 2 of a node with weight w. The productions in lower layers are: $\cdots 0 \rightarrow 1(level\ n-2) \rightarrow 2(level\ n-1) \rightarrow 4, 5(level\ n)$. The only way to nullify w is, RMT 1 at level $n-2$ and RMT 2 at level $n-1$ are 0, or they both are 1. Here also, w can either be 0 or 1. The same thing is also true for other two sets – set 0 and set 3.

Reason for the second property: RMT 7 for each rule of NCCA is 1. With similar logic, it can be shown that RMT 7 of any node can have weight either 0 or -1.

Since the RMT 0 (7) can also be generated from RMT 4 (3) which can again be generated from RMT 2 (1) (Tab. 2), the above properties also specify the limit of weight that an RMT can take at any level of reachability tree. This property takes a role in the synthesis of NCCA. Next we present the algorithm.

Algorithm 2 *SynthesizeNCCA*
Input: n *(size if CA)*
Output: *NCCA with rule vector* $\mathcal{R} = \langle \mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_n \rangle$
Step 1: *Form root of reachability tree. Assign weights for RMTs at root as 0.*
Step 2: *For* $i = 1$ *to* n *repeat* Step 3 *to* Step 6.
Step 3: *Randomly synthesize the rule* $\mathcal{R}_i$ *so that*

  1. *RMT 0 and RMT 7 of* $\mathcal{R}_i$ *are 0 and 1 respectively.*

  2. *The above mentioned properties for RMT 0 and RMT 7 are followed in* i^{th} *level.*

  3. *Theorem 2 and Theorem 3 are obeyed by the nodes of* i^{th} *level.*

Step 4: *If no such rule exists, go to* Step 1.
Step 5: *Get the nodes of* i^{th} *level depending on the nodes of* $(i-1)^{th}$ *level and rule* $\mathcal{R}_i$.
Step 6: *Identify and remove sub-nodes while Corollary 3 is obeyed.*
Step 7: *Get the nodes at level* $n-2$ *and* $n-1$ *to synthesize* $\mathcal{R}_{n-1}$ *and* $\mathcal{R}_n$ *respectively, so that conditions at* Step 3 *are satisfied and the leaves can have RMTs with zero weight.*
Step 8: *If no such* $\mathcal{R}_{n-1}$ *or* $\mathcal{R}_n$ *is found, go to* Step 1.
Step 9: *Output the NCCA rule vector* $\mathcal{R} = \langle \mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_n \rangle$.

V   Conclusion

The paper has presented the characterization of non-uniform or hybrid number conserving cellular automata (NCCA). To characterize such NCCA, we utilize the reachability tree of CA, which was proposed as a tool for characterization of CA. This paper has characterized the reachability tree for NCCA. A set of theorems and corollaries are developed to complete such characterizations. Finally, depending on such characterization two algorithms are developed – one for verification of NCCA and another for the synthesis of the same. The major application area of such hybrid NCCA may be the modeling the highway traffic.

References

[BF98]   Nino Boccara and Henryk Fukś. Cellular automaton rules conserving the number of active sites. *Journal of Physics A: math. gen*, 31:6007, 1998.

[BF02]   Nino Boccara and Henryk Fukś. Number-conserving cellular automaton rules. *Fundam. Inf.*, 52:1–13, April 2002.

[Das11]   Sukanta Das. Cellular automata based traffic model that allows the cars to move with a small velocity during congestion. *Chaos, Solitons & Fractals*, 44(4-5):185–190, May 2011.

[DFR03] Bruno Durand, Enrico Formenti, and Zsuzsanna Róka. Number-conserving cellular automata i: decidability. *Theoretical Computer Science*, 299:523–535, April 2003.

[DS06] Sukanta Das and Biplab K Sikdar. Classification of CA Rules Targeting Synthesis of Reversible Cellular Automata. In *Proceedings of International Conference on Cellular Automata for Research and Industry, ACRI,* France, pages 68–77, September 2006.

[DS09] Sukanta Das and Biplab K. Sikdar. Characterization of 1-d periodic boundary reversible ca. *Electr. Notes Theor. Comput. Sci.*, 252:205–227, 2009.

[DS10] Sukanta Das and Biplab K. Sikdar. A scalable test structure for multicore chip. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 29(1):127–137, 2010.

[DSC04] Sukanta Das, Biplab K Sikdar, and P Pal Chaudhuri. Characterization of Reachable/Nonreachable Cellular Automata States. In *Proceedings of Sixth International Conference on Cellular Automata for Research and Industry, ACRI, The Netherlands*, pages 813–822, October 2004.

[DSS09] Sukanta Das, Meghnath Saha, and Biplab K Sikdar. A cellular automata based model for traffic in congested city. In *Proc. of IEEE SMC conference*, pages 2397–2402, 2009.

[FI96] M. Fukui and Y. Ishibashi. Traffic flow in 1d cellular automaton model including cars moving with high speed. *Journal of the Physical Society of Japan*, 65(6):1868–1870, 1996.

[HT91] Tetsuya Hattori and Shinji Takesue. Additive conserved quantities in discrete-time lattice dynamical systems. *Phys. D*, 49:295–322, April 1991.

[MI98] Kenichi Morita and Katsunobu Imai. Number-conserving reversible cellular automata and their computation-universality. In *Proc. of Satellite Workshop on Cellular Automata MFCS'98*, pages 51–68, 1998.

[NS92] K. Nagel and M. Schreckenberg. A cellular automata model for freeway traffic. *Journal de Physique I*, 2:2221 – 2229, December 1992.

[vN66] John von Neumann. *The theory of self-reproducing Automata, A. W. Burks ed.* Univ. of Illinois Press, Urbana and London, 1966.

[Wol86] S. Wolfram. *Theory and applications of cellular automata*. World Scientific, Singapore, 1986. ISBN 9971-50-124-4 pbk.

On the Reversibility of 1-dimensional Asynchronous Cellular Automata [†]

Anindita Sarkar[‡] and Sukanta Das[§]

Department of Information Technology
Bengal Engineering and Science University, Shibpur
Howrah, West Bengal, India 711103

The cells of asynchronous cellular automata (ACA) are independent, and they are updated independently. However, two adjacent cells can not act simultaneously since the actions taken by the cells are atomic. This paper addresses the question of reversibility of such ACA with dimension one. To our knowledge, the reversibility of 1-dimensional ACA is an untouched issue. We classify the CA rules for ACA reversible and irreversible to address this issue. The irreversible rules can not configure reversible ACA. The reversible rules, on the other hand may configure reversible ACA if update of ACA cells follow some pattern. Finally, we sketch an algorithm to get reversible ACA while some criteria of the update of ACA cells are fulfilled.

Keywords: Asynchronous cellular automata (ACA), reversibility, CA rules, update pattern

I   Introduction

The concept of asynchronous cellular automata (ACA) was first developed on 1-dimensional lattice [IB84]. Zielonka provided a formal definition of asynchronous automata, as well as asynchronous cellular automata (ACA) for 2-dimensional CA structure [CMZ93, Zie87]. The property making them different from the usual cellular automata is that they have decentralized control structure and they perform actions asynchronously. While a cell makes an action, it examines the states of all of its neighbors, and then changes its own state in accordance with its transition function [CMZ93, Zie87]. This single action is considered as atomic which implies that two neighboring cells cannot act simultaneously.

The reversibility of synchronous cellular automata has been studied extensively for years [AP72, DS06, Tof77]. However, reversibility of ACA is almost an untouched issue. A very few works on the issue for 2-dimensional ACA are found in literature [LPA⁺02]. However, the reversibility of 1-dimensional ACA is an unexplored field. In this scenario, we target to explore the issue for 1-dimensional two-state 3-neighborhood ACA. We use the term *reversibility* in classical sense – that is, starting from a CA state, reversible ACA can reach to that particular CA state after a number of steps. During their evolution,

more than one ACA cell may be updated simultaneously. But no two neighboring cells can be updated in a single step. This is required to achieve the atomicity of an action which makes the action indivisible to the outside world. However, compromising the atomicity property, another work on reversibility of 1-dimensional ACA is also developed [SMD11].

Based on the update of ACA cells in subsequent steps, we, like [Neh04], define *update pattern* to know which cell is updated when. While an update pattern along with an initial state is given, the transition of CA states for some ACA can be observed. The update patterns play a major role in the reversibility of ACA.

We have also identified a number of CA 'rules' [Wol86] for ACA *irreversible rules*, which can not configure reversible ACA with any set of update patterns. Only *reversible rules* can configure reversible ACA with a particular set of update patterns. An algorithm is also developed to find an update pattern of a cycle for some reversible ACA.

The paper is organized as follows. The preliminaries of cellular automata are provided next. Section III identifies the irreversible rules for ACA in two different boundary conditions. An algorithm to design reversible ACA is reported in Section IV. Section V concludes the paper.

II Cellular Automata

The cellular automata (CA) are the discrete spatially-extended dynamical systems that have been studied extensively as models of physical systems. They evolve in discrete space and time. In their simplest form, as it is proposed by Wolfram [Wol86], CA consist of a lattice of cells, each of which stores a discrete variable at time t that refers to the present state of the CA cell. The next state of a cell is affected by its present state and the present states of its $neighbors$ at time t. In 1-dimensional two-state 3-neighborhood (self, left and right neighbors) CA, next state of each cell is determined as:

$$S_i^{t+1} = f(S_{i-1}^t, S_i^t, S_{i+1}^t) \tag{1}$$

where f is the next state function; S_{i-1}^t, S_i^t and S_{i+1}^t are the present states of the left neighbor, self and right neighbor of the ith CA cell at time t. The $f : \{0,1\}^3 \mapsto \{0,1\}$ can be expressed as a look-up table as shown in Table 1. The decimal equivalent of the 8 outputs is called 'rule' [Wol86]. There are 2^8 (256) CA rules in two-state 3-neighborhood dependency. Two such rules are 123 and 51 (Table 1). From the view point of *Switching Theory*, a combination of the present states (first row of Table 1) can be viewed as the *Min Term* of a 3-variable $(S_{i-1}^t, S_i^t, S_{i+1}^t)$ switching function. So, each column of the first row of Table 1 is referred to as Rule Min Term (RMT).

The collection of states of all cells $(S_1^t, S_2^t, \cdots, S_n^t)$ at time t is called a CA state on that time. If the left most and right most cells are the neighbors of each other (that is, $S_0^t = S_n^t$ and $S_{n+1}^t = S_1^t$ for CA with n cells), the CA are *periodic boundary* CA. On the other hand, in *null boundary* CA, $S_0^t = S_{n+1}^t = 0$ (null). The present work concentrates on both the boundary conditions – periodic and null.

II.1 Asynchronous CA

Traditional CA are *synchronous* where all the cells of CA update their states simultaneously in each discrete time step. In *asynchronous CA*, the cells are updated independently. The asynchronous cellular automata (ACA) have decentralized control structure, and as a result, any number of ACA cells may be updated in a single time step. When a cell changes its state, it reads first the states of all of its neighbors,

Tab. 1: Look-up table for rule 123 and 51

Present state :	111	110	101	100	011	010	001	000	*Rule*
(RMT)	(7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	
(i) Next State :	0	1	1	1	1	0	1	1	123
(ii) Next State :	0	0	1	1	0	0	1	1	51
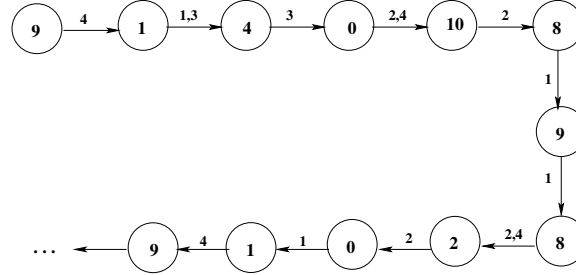
Fig. 1: Partial state transition diagram of rule 123 ACA. The cells updated during evolution are noted over the arrows

and then follows the state transition function (Table 1) to get the next state. The entire operation (reading of neighbors' states and change of cell's own state) is considered as atomic. It implies that no two neighboring cells can act simultaneously to change its state [CMZ93, Zie87]. However, more than one cell following the atomicity property may act simultaneously. Since this paper deals with only 1-dimensional two-state 3-neighborhood ACA, at most half of the cells may act together. As a special case, a single cell may be updated in each discrete time step (like [IB84]).

II.2 Update pattern

During their evolution with time, CA (synchronous and asynchronous) generate a sequence of states. The next state of a CA state can be determined in synchronous CA configured with a particular rule. However, the next state of ACA depends not only on the rule, but also on the cells which are updated at that time. We denote the set of cells, updated at time t, as u_t. Therefore, one can get an *update pattern* $U = < u_1, u_2, \cdots, u_t, \cdots >$ to observe which cells are updated when. If the CA rule and an update pattern with an initial state is given, the state transitions for the ACA can be identified. A partial state transition diagram of 4-cell rule 123 ACA with null boundary condition is shown in Figure 1. The states are noted in circles (decimal numbers in states are the decimal equivalent of binary states), whereas the cells updated during state transitions are noted over arrows. The update pattern for this transition $U = < \{4\}, \{1,3\}, \{3\}, \{2,4\}, \{2\}, \{1\}, \{1\}, \{2,4\}, \{2\}, \{1\}, \{4\}, \cdots >$, is associated with CA state 9. The output of first cell is considered as the LSB (least significant bit) of CA state. It is, therefore, obvious that the state transition of ACA depends on both, the CA rule and the update pattern. However, a single state transition diagram may not cover all the CA states. To observe the transitions of other CA states, another one or more update patterns are to be there. A set of update patterns can actually illustrate the transition of all states.
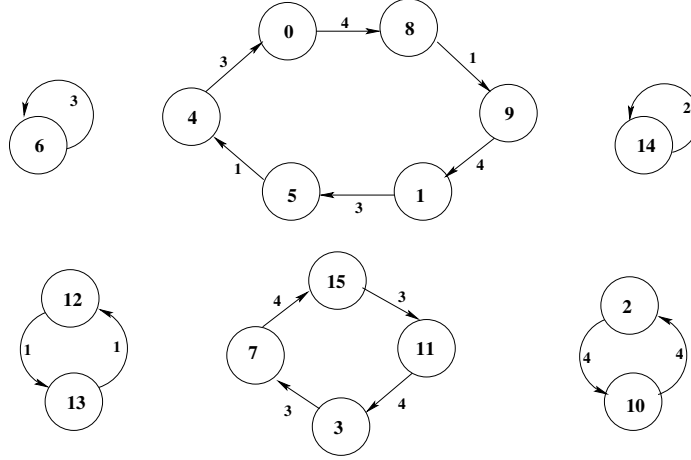
Fig. 2: 4-cell rule 123 reversible ACA in null boundary condition. The cells updated are noted on edges

III   The irreversible ACA rules

State transition diagram classifies the CA states as *cyclic* and *acyclic*. If a CA state lies on some cycle in state transition diagram of CA, the state is cyclic, otherwise it is acyclic. The CA are *reversible* if all the CA states are cyclic, otherwise they are *irreversible*. The reversibility, explored in synchronous domain, guarantees that each CA state has unique predecessor and successor.

Definition 1 *The ACA are* **reversible** *if each CA state can be reached starting from the state with an update pattern and without generating any intermediate CA state twice or more during the evolution. Otherwise, they are* **irreversible***.*

The ACA of Fig. 1 are irreversible. On the other hand, Fig. 2 depicts the state transition diagram of 4-cell rule 123 reversible ACA with null boundary condition. There are 6 update patterns, one for each cycle, in the ACA. The update patterns (with corresponding initial states) are $<\{1\}, \{1\}>$ (12), $<\{4\}, \{1\}, \{4\}, \{3\}, \{1\}, \{3\} >$ (0), $<\{4\}, \{4\}>$ (2), $<\{3\}>$ (6), $<\{3\} ,\{4\}, \{3\}, \{4\} >$ (15) and $<\{2\}>$ (14). The CA rules, building block of reversible and irreversible ACA, are classified next as the reversible and irreversible rules.

Definition 2 *A CA rule R is an* **irreversible rule** *if there is a CA state which can never be cyclic for any update pattern, while the ACA are configured with R. Otherwise, R is a* **reversible rule***.*

For example, rule 77 (01001101_2) in null-boundary condition is an irreversible rule. Starting from the all-0 CA state, one can not return back to $00\cdots0$ state in rule 77 ACA with any update pattern. On the other hand, rule 123 is a reversible rule in null and periodic boundary conditions. Each state can be reached for some update pattern (Figure 2).

Now we characterize the irreversible rules that can never configure reversible ACA. Following theorem characterizes the irreversible rules in both the boundary conditions – periodic and null.

Theorem 1 *ACA rule R is irreversible if and only if any one of the following states of ACA, configured with R, cannot be returned back after arbitrary no of steps with any update pattern -*

(i) all-0 state
(ii) all-1 state
(iii) 1010... state
(iv) 100100...state
(v) 110110...state

Proof: If all-0, all-1, $1010\cdots$, $100100\cdots$ or $110110\cdots$ state of ACA, configured with R, can not be returned back with any update pattern, then obviously the ACA, and hence the R are irreversible. Now, we shall show that R is irreversible if only any of these states is acyclic.

A CA state can be viewed as a sequence of RMTs. For example, the state 1100 in periodic boundary condition can be viewed as 3641, where 3, 6, 4 and 1 are corresponding RMTs. We club the 8 RMTs into 4 sets – $\{0, 2\}$, $\{1, 3\}$, $\{4, 6\}$ and $\{5, 7\}$. The 3-bit binary representation of the RMTs show that the middle bit of RMTs of each set is the complement of each other. We next show that if a sequence of RMTs of an arbitrary rule, corresponding to some CA state, contains both the elements of any one of the above sets i.e. $\{0, 2\}$ or $\{1, 3\}$ or $\{4, 6\}$ or $\{5, 7\}$ set, the state is cyclic.

Rest of the states whose corresponding RMTs are from different sets (for example, one RMT from $\{0, 2\}$, one from $\{4, 6\}$ and other from $\{1, 3\}$ to get RMT sequence $< 2, 4, 1, 2 >$ in null boundary), may form single length cycles depending upon the RMT values and by updating a single cell. The states which are not in some cycle, can form two length cycles by updating two or more cells at a time. Hence, these states are also cyclic.

Therefore, all the states other than all-0, all-1, $1010\cdots$, $100100\cdots$ or $110110\cdots$ of any ACA can be cyclic for some update patterns. Hence, if all-0, all-1, $1010\cdots$, $100100\cdots$ or $110110\cdots$ states are cyclic, the rule R that configures ACA is reversible, otherwise R is irreversible. $\square$

To identify the irreversible rules in null and periodic boundary conditions, we next report two corollaries following Theorem 1.

Corollary 1 : *A rule R is irreversible while it configures ACA in periodic-boundary condition if*
(i) the RMTs 0 and 2 of R are 1, or
(ii) the RMTs 5 and 7 of R are 0, or
(iii) the RMTs 5 and 7 are 1 and RMTs 0 and 2 of R are 0, or
(iv)the RMTs 1, 3, 4, 6 of R are 1 and RMTs 0 and 2 are 0, or
(v) the RMTs 1, 3, 4, 6 of R are 0 and RMTs 5 and 7 are 1.

Proof: We shall prove the corollary by identifying the RMTs of R for which all-0, all-1, 1010...10, 100100...100, 11011...110 state can not be returned back (Theorem 1).

If RMT 0 is 1, the ACA, configured with R in periodic boundary condition, can not form a single length cycle with all-0 state. Because, the next state contains at least one 1 while the ACA are updated. To form a cycle, these 1s are to be 0 in subsequent steps. However, these 1s can not be 0 if RMTs 2 is 1. Therefore, all-0 state can not be returned back if the RMTs 0, 2 of R are 1.

Similarly, the ACA can not form a single length cycle with all-1 state if RMT 7 is 0. Moreover, the ACA with the state can never form a cycle of any length in periodic boundary condition if RMTs 5 of R is 0. Hence, all-1 state can not be returned back if the RMTs 5 and 7 are 0.

Similarly, the ACA can not form a single length cycle with $1010\cdots10$ state if RMT 5 is 1 and RMT 2 is 0. The ACA with the state can never form cycle of any other length in periodic boundary condition if

Tab. 2: Irreversible rules of periodic−boundary ACA

0	1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	20	21	24	25
28	29	31	37	39	45	47	53	55	61	63
64	65	66	67	68	69	70	71	72	73	74
75	76	77	78	79	80	81	84	85	87	88
89	90	92	93	94	95	101	103	109	111	117
119	122	125	127	133	135	141	143	149	151	157
159	160	161	162	164	165	167	168	170	173	175
176	187	181	183	184	186	189	191	197	199	205
207	213	215	218	221	223	224	226	229	231	232
234	237	239	240	242	245	247	248	250	253	255

RMT 7 of R is 1 and RMT 0 of R is 0. So, this state can not be returned back if the RMTs 5 and 7 are 1 and RMTs 0 and 2 are 0.

The state $100100\cdots100$ contains RMTs 1, 2 and 4 in periodic boundary condition. It can not form single length cycle if the RMTs 1 and 4 are 1 and RMT 2 is 0. Again, if the RMTs 3 and 6 are 1 and RMT 0 is 0 the state $100100\cdots10$ can not returned back from the previous state. Thus the state $100100\cdots10$ cannot form cycle if the RMTs 1, 3, 4, 6 are 1 and RMTs 0 and 2 are 0.

Similarly, in periodic boundary condition the state $110110\cdots110$ contains RMTs 3, 5 and 6. To avoid single length cycle the RMTs 3 and 6 are needed to be 0 and RMT 5 is to be 1. Further this state will be acyclic if the RMTs 1 and 4 are 0 and RMT 7 is 1. Hence the state $110110\cdots110$ cannot form cycle of any length if the RMTs 1, 3, 4 and 6 are 0 and RMTs 5 and 7 are 1. $\qquad\square$

For periodic-boundary condition, there are (i) 64 rules where the RMTs 0, 2 of R are 1, (ii) 64 rules where the RMTs 5, 7 are 0, (iii) 16 rules where the RMTs 5 and 7 are 1 and RMTs 0 and 2 are 0, (iv) 4 rules where the RMTs 1, 3, 4, 6 are 1 and RMTs 0 and 2 are 0, (v) 4 rules where the RMTs 1, 3, 4, 6 are 0 and RMTs 5 and 7 are 1. There are 121 irreversible rules in total. All irreversible rules for periodic-boundary ACA are listed in Table 2.

To derive the irreversible rules in null-boundary condition, another corollary of Theorem 1 is followed.

Corollary 2 : *A rule R is irreversible in $null\text{-}boundary$ condition if*
(i) the RMTs 0, 2 of R are 1, or
(ii) the RMTs 1, 3, 4, 5, 6 and 7 are 0, or
(iii) the RMTs 5 and 7 are 1 and the RMTs 0 and 2 are 0, or
(iv) the RMTs 1, 3, 4, 6 are 1 and 0 and 2 are 0, or
(v) the RMTs 1, 3, 4, 6 are 0 and 5 and 7 are 1.

Proof: We shall prove the corollary by identifying the RMTs of R for which all-0, all-1, $10101\cdots01$, $100100\cdots001$ or $110110\cdots11$ state can not be returned back (Theorem 1).

If RMT 0 is 1, the ACA, configured with R in null boundary condition, can not form a single length cycle with all-0 state, as the next state always contains at least one 1. To form a cycle, these 1s are to be 0 in subsequent steps. However, these 1s can not be 0 if RMTs 2 is 1. Therefore, all-0 state can not be returned back if the RMTs 0, 2 of R are 1.

Tab. 3: Irreversible rules of null−boundary ACA

0	1	4	5	7	13	15	21	23	29	31
37	39	45	47	53	55	61	63	69	71	77
79	85	87	90	93	95	101	103	109	111	117
119	122	125	127	133	135	141	143	149	151	157
159	160	161	162	164	165	167	168	170	173	175
176	178	181	183	184	186	189	191	197	199	205
207	213	215	218	221	223	224	226	229	231	232
234	237	239	240	242	245	247	248	250	253	255

In null boundary condition, the state of left (right) neighbor of left most (right most) cell is always 0. So, RMT 3 and RMT 7 (RMT 6 and RMT 7) of R are equivalent for the left most (right most) cell. Therefore, the ACA with all-1 state can form single length cycle for some update pattern if RMT 3, RMT 6 or RMT 7 be 1. To restrict such cycle, the RMTs 3, 6 and 7 of R are to be 0. While these RMTs are 0, the ACA with all-1 state, due to the update of cells, reaches to another state that contains at least one 0. However, the all-1 state can not be returned back if RMTs 0, 1, 4 and 5 are 0. So, the all-1 state is acyclic if the RMTs 0, 1, 3, 4, 5, 6 and 7 are 0.

To form a single length cycle with $10101\cdots01$ state, RMT 2 is to be 1 or RMT 5 is to be 0. If RMT 2 is 0 and RMT 5 is 1, single length cycle in null boundary condition can not be formed with $10101\cdots01$ state. The $10101\cdots01$ state can not be returned back in null boundary condition if RMT 7 is 1 and RMT 0 is 0.

Thus, $100100\cdots001$ can not be returned back if the RMTs 1, 3, 4, 6 are 1 and 0 and 2 are 0 and $110110\cdots11$ state will be acyclic if the RMTs 1, 3, 4, 6 are 0 and 5 and 7 are 1 as discussed in periodic boundary condition. □

For *null-boundary* condition, there are (*i*) 64 rules where RMTs 0 and 2 are 1 and either, (*ii*) 4 rules where RMTs 1, 3, 4, 5, 6 and 7 are 0, (*iii*) 16 rules where the RMTs 5 and 7 are 1 and the RMTs 0, 1, 2 and 4 are 0, (*iv*) 4 rules where the RMTs 1,3,4,6 are 1 and 0,2 are 0, or (*v*) 4 rules where the RMTs 1, 3, 4, 6 are 1 and RMTs 5 and 7 are 1. There are 88 irreversible rules in total. All irreversible rules for *null-boundary* ACA are listed in Table 3.

The rules other than irreversible rules may configure reversible ACA. We next report the design of reversible ACA.

IV   Design of reversible ACA

The reversibility of ACA depends not only on the rule, but also on update patterns. For example, rule 60 can configure irreversible ACA (Figure 1), as well as reversible ACA (Figure 2) depending upon the update patterns. Since the ACA cells are independent, and so updated arbitrarily, it can not be predicted in advance that ACA configured with a reversible rule are reversible. If a set of update patterns, received from ACA configured with a reversible rule during generation of all states, are given, then only one can analyze whether the ACA were reversible. This discussion leads to the following theorem.

Theorem 2 *It is impossible to synthesize reversible 1-d ACA.*

However, the update patterns can be designed for the cycles of some reversible ACA. The reversible rules require different sets of update patterns to get reversible ACA. Even, for a particular reversible rule, various sets of update patterns may be identified that result in different reversible ACA. An update pattern can produce a cycle if the initial state and the ACA are given. In this section, we identify such an update pattern that forms a cycle for some reversible ACA.

To get a cycle for some reversible ACA, an update pattern along with some initial state is required which generates l distinct CA states for a cycle of length l. Since the states are to be distinct, the update pattern should be designed in such a way that at least one bit of a state flips to get the next state. But no two neighboring bits can be selected at a time. Moreover, in any sub-sequence of states, the bits of states are not to be flipped in even number of times to get unique states in a cycle of reversible ACA. If they flip, the l states can not be distinct. Therefore, the following points are to be taken care of to design the update patterns for some reversible ACA.

1. No two neighbors can get updated simultaneously.

2. At least one bit of a state must be flipped to get the next state.

3. To get cycle of length l, all the bits of an initial state (S) are flipped even number of times in total to regenerate S. Before regenerating S, the bits are not to be flipped even number of times simultaneously.

The generation of distinct states depends not only on the update pattern, but also on the initial state. Because, the initial state may not allow an arbitrary bit to flip for some arbitrary reversible rule that configure the ACA. However, rule 51 (Table 1) is the only rule that always allows a cell to flip its previous state when updated. So, rule 51 ACA do not depend on the initial state to form a cycle. We have designed the following rule to generate an update pattern for getting a cycle of length 2^i ($1 \leq i \leq n$) by updating a single cell at a time, where n is the number of ACA cells.

- To get a cycle of length 2^i ($1 \leq i \leq n$) of rule 51 ACA with n cells, form a sequence of i cells, to be updated, arbitrarily. Start with an arbitrary state. Update $(2^{j-1})^{th}$ state by updating j^{th} cell ($1 \leq j \leq i$) of the sequence to generate the next state. Repeat the update of the jth cell after each 2^j state, where $j < i$. However, update the ith cell again after 2^{i-1} state to get a cycle of length 2^i.

Example 1 To design a full length cycle for 4-cell rule 51 ACA ($length = 2^4$), all the cells are to be updated in some sequence. Consider, the sequence of updating is $SEQ =< 1, 2, 3, 4 >$ and the initial state is 0100. Each jth cell of SEQ is selected for the first time to update $(2^{j-1})^{th}$ state. Hence, to get the second state, the first bit of the initial state ($(2^{j-1})^{th}$ state, where $j = 1$) is updated. Similarly, the second, third and fourth cells are selected for the first time to update the second, fourth and eighth states respectively. The first cell is again selected to update third, fifth, and all odd states (that is, after each 2^j states where $j = 1$). After the first time update, the second and third cells are selected repeatedly to update after every 2^2 and 2^3 states respectively. The last cell is updated for the second time after 2^3 states (2^{i-1} states where $i = 4$) to complete the cycle. Therefore, the sequence of states in the cycle is <0100, 1100, 1000, 0000, 0010, 1010, 1110, 0110, 0111, 1111, 1011, 0011, 0001, 1001, 1101, 0101, 0100>. The update pattern is <{1}, {2}, {1}, {3}, {1}, {2}, {1}, {4}, {1}, {2}, {1}, {3}, {1}, {2}, {1}, {4}>. Here, the update pattern is independent of the initial state, but depends on SEQ. The update pattern and the cycle of rule 51 ACA are same for both the boundary conditions.

However, cycles can be formed by updating multiple cells simultaneously. Rule 51 ACA with n cells can form a cycle of maximum length 2^{n-m+1} while m cells ($1 \leq m \leq n$) are updated simultaneously. In such case, the same rule of single cell update to get a cycle can be followed with an exception that each entry in the sequence of cells, to be updated, is a set of m cells. Following example illustrates the cycle formation by updating multiple cells.

Example 2 Let us consider, $n = 4$ and $m = 2$. To get an 8 length (2^{n-m+1}) cycle of the ACA, a sequence $SEQ =<\{1, 3\}, \{2, 4\}, \{1, 4\}>$ of cells is formed arbitrarily. Consider that the initial state is 0100. The first and third bits are updated to generate the second state (1110). Similarly, the cells of second and third entries of SEQ are selected to update the second and forth states. Like Example 1, the cells of first set ($\{1, 3\}$) are repeatedly selected to update the odd states. The cells of second set ($\{2, 4\}$) are selected again to update the sixth state. Therefore, a sequence <0100, 1110, 1011, 0001, 1000, 0010, 0111, 1101, 0100> of states is obtained. Here, the update pattern is <$\{1, 3\}, \{2, 4\}, \{1, 3\}, \{1, 4\}, \{1, 3\}, \{2, 4\}, \{1, 3\}, \{1, 4\}$>.

The update rule, designed for rule 51 reversible ACA, guides us to develop Algorithm 1, which finds the update pattern for a cycle of some reversible ACA. The algorithm is independent from the boundary condition. It takes the ACA, cycle length to be designed (2^i), initial state (S) and the number of cells updated in a single step (m), as input. However, with arbitrary ACA and arbitrary initial state, a cycle of given length may not be designed. In such cases, the algorithm finds a cycle which is close in length with the given cycle length. It outputs the update pattern with the cycle length, if cycle can be designed.

The algorithm first forms a sequence of i unique sets arbitrarily. The sets are also designed arbitrarily with m ACA cells per set. The update style of rule 51 reversible ACA is followed to generate the update pattern. If no bit flips during the update of a set of m cells, another set of m cells is searched so that at least one bit flips. If no such set is found, then the algorithm reports that cycle is not possible. While 2^i states are covered but no cycle is formed, the algorithm attempts to form a cycle by generating a very few states.

Algorithm 1 *ReversibleACA*
Input: *R (rule), n (# cells), 2^i (cycle length, $1 \leq i \leq n$), S (initial state), m (# cells updated in each step)*
Output: *Update pattern with cycle length, if cycle is possible*
Step 1: *Form a sequence, SEQ of i unique sets of m ACA cells such that no two neighboring cells are not taken at a time, arbitrarily.*
Step 2: *Load the ACA, configured with R, with S.*
Step 3: *For $k = 1$ to 2^i repeat Step 4 to Step 9.*
Step 4: *If $k = 2^{j-1}$ ($1 \leq j \leq i$), select the jth set of the SEQ.*
   If $k = 2^i$, select the ith set of SEQ.
   If $k = 2^{j-1} + p * 2^j$, where p is some positive integer and $1 \leq j < i$, select the jth set.
Step 5: *Update the ACA cells of the selected set.*
Step 6: *If no cell flips during the update, find a set of m cells so that*

   1. *at least one cell flips, and*

   2. *the generated state is unique.*

111	110	101	100	011	010	001	000	Rule
(7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	
1	0	0	1	0	0	1	1	147

U.P. of 51	1 2 3 4 5	U.P. generated	SEQ <{3,5} ,{2,4}, {1,3}>
3,5	1 0 0 1 0	3,5	
2,4	1 0 1 1 1	1,4	
3,5	0 0 1 1 1	3,5	
1,3	0 0 0 1 0	1,3	
3,5	1 0 1 1 0	3,5	
2,4	1 0 0 1 1	2,4	
3,5	1 1 0 0 1	3,5	
1,3	1 1 1 0 0	1,3	
	0 1 0 0 0	1,4	
	1 1 0 1 0	2	
	1 0 0 1 0		

Fig. 3: Generation of cycle for rule 147 ACA. At most two cells are updated simultaneously.

Otherwise, goto Step 9.
Step 7: *If no such set is found in Step 6, goto Step 14.*
Step 8: *Update the ACA cells according to the set, designed in Step 6.*
Step 9: *Print the ACA cells that are updated to generate the next state of k.*
Step 10: *If no cycle is formed, identify the bits of $2^i + 1$ state which differ from the initial state, S.*
    Otherwise, go to Step 15.
Step 11: *Update the ACA cells to flip the identified bits.*
Step 12: *If few cells flip, print those cells. Update the nearest cells of the rest bits one-by-one or more than one at a time, so that the S is returned back within few steps.*
Step 13: *If no cycle is formed, goto Step 14, otherwise goto Step 15.*
Step 14: *Print 'Cycle is not possible', and exit.*
Step 15: *Print the length of cycle.*

Following example illustrates the execution of Algorithm 1.

Example 3 Let us consider, $R = 147$, $n = 5$, cycle length = 8 (2^3), $S = 10010$ and $m = 2$. Formation of cycle following Algorithm 1 is shown in Figure 3. Firstly, a sequence of 3 sets $SEQ =<\{3, 5\}, \{2, 4\}, \{1, 3\}>$ is formed arbitrarily so that no two adjacent cells are the member of any set (Step 1). The ACA

are configured with rule 147 in null boundary condition. To get the next state of 10010 (initial state), the third and fifth cells are updated (Steps 4 and 5). In Figure 3, update pattern of rule 51 ACA is noted on the left side of the states, and the update pattern generated by the algorithm is shown on the right side. To update the second state (similarly sixth state), according to the update pattern of rule 51 ACA, the set $\{2, 4\}$ is selected. Since no cell flips here, another set $\{1, 4\}$ is searched (Step 6). After generation of 8 states, cycle is not formed. So, another 2 states are generated to form a cycle (Steps 10 – 12). Therefore, length of cycle is 10.

V   Conclusion

The reversibility in 1-dimensional ACA (asynchronous cellular automata) has been addressed in this paper. The ACA cells are updated independently. An action taken by a cell to change its state is considered here as atomic. Depending upon their update during state transition, update pattern is defined. To facilitate the design of reversible ACA, CA rules are classified as reversible and irreversible. The irreversible rules can not configure reversible ACA with any set of update patterns. However, the reversibility of ACA depends on both – the rule and update patterns. A set of irreversible rules for both the boundary conditions are identified. The paper has finally reported an algorithm to get an update pattern for a cycle of ACA.

References

[AP72]   S. Amoroso and Y. N. Patt. Decision procedures for surjectivity and injectivity of parallel maps for tesselation structures. *J. Comput. Syst. Sci.*, 6:448–464, 1972.

[CMZ93]   R. Cori, Y. Metivier, and W. Zielonka. Asynchronous mappings and asynchronous cellular automata. *Inf. Comput.*, 106:159–202, 1993.

[DS06]   Sukanta Das and Biplab K. Sikdar. Classification of CA rules targeting synthesis of reversible cellular automata. In *Proc. of International Conference on Cellular Automata for Research and Industry*, pages 68–77. ACRI, September 2006.

[IB84]   T. Ingerson and R. Buvel. Structure in asynchronous cellular automata. *Physica D*, 10:59–68, 1984.

[LPA$^+$02] Jia Lee, Ferdinand Peper, Susumu Adachi, Kenichi Morita, and Shinro Mashiko. Reversible computation in asynchronous cellular automata. In *Proc. of the Third International Conference on Unconventional Models of Computation*, pages 220–229. Springer-Verlag, London, 2002.

[Neh04]   Chrystopher L. Nehaniv. Asynchronous automata networks can emulate any synchronous automata network. *International Journal of Algebra and Computation (IJAC)*, 14:719–739, 2004.

[SMD11]   Anindita Sarkar, Anindita Mukherjee, and Sukanta Das. Reversibility in asynchronous cellular automata. *Communicated to Complex Systems*, March 2011.

[Tof77]   T. Toffoli. Computation and construction universality of reversible cellular automata. *J. Comput. System Sci.*, 15:213–231, 1977.

[vN66]   John von Neumann. *The theory of self-reproducing Automata*. Univ. of Illinois Press, 1966.

[Wol86] S. Wolfram. *Theory and applications of cellular automata*. World Scientific, Singapore, 1986.

[Zie87] Wieslaw Zielonka. Notes on finite asynchronous automata. *Theoretical Informatics and Applications*, 21:99–135, 1987.

On 1-resilient, radius 2 elementary CA rules

E. Formenti[1][†] and K. Imai[2][‡] and B. Martin[1][§] and J-B. Yunès[3][¶]

[1]*Université Nice–Sophia Antipolis, Laboratoire I3S, UMR 6070 CNRS, BP 121, F-06903 Sophia Antipolis Cedex.*
[2]*Graduate School of Engineering, Hiroshima University, Japan.*
[3]*Université Paris Diderot & CNRS, Laboratoire d'Informatique et Algorithmique, Fondements et Applications (LI-AFA), Case 7014, F-75205 Paris cedex 13.*

The study of cellular automata rules suitable for cryptographic applications is under consideration. Cellular automata can be used to generate pseudo-random sequences as well as for the design of S-boxes in symmetric cryptography. Boolean functions with good properties like resiliency and non-linearity are usually obtained either by exhaustive search or by the use of genetic algorithms. We propose here to use some recent research in the classification of Boolean functions and to link the study of cellular automata rules to the study of such Boolean functions. We illustrate our approach with 5-variable Boolean functions.

Keywords: Cellular automata, Boolean functions, Pseudo-random generators, Symmetric cryptography.

1   Introduction

Cellular automata (CA) are models of massive parallel computers based on communicating finite state machines; they are used in many applications of computer science. In particular, they are utilised for the generation of binary pseudo-random (PR) sequences in cryptography. This was first proposed by [Wol86a] who suggested that PR sequences generated by a certain rule (numbered 30) could be used for cryptographic purposes as keys for a Vernam-type cipher. This CA rule is also used as one of the PR generators included in Mathematica®. It is now well known that rule 30 has several weaknesses and that it is not resistant to the attacks of [MS91]. One of the reasons is that the rule is not resilient, meaning that the output of rule 30 is not statistically independent of the combination of some subsets of its inputs. This can be proved either by an exhaustive search ([Mar08]) or by a simple application of the Siegenthaler bound which links the number of variables of a Boolean function with the resiliency.

Despite this negative result, several techniques have been used to pursue the study of PR generation by CA. The first natural idea is to enlarge the neighbourhood and to search for rules with good properties of non-linearity and resiliency. [LMS08] investigate the 2^{16} elementary CA rules with four neighbours and give a complete classification of the functions with good cryptographic properties.

In the present work, we recall how to generate binary PR sequences with uniform CA. We particularly focus on the search for good updating functions. By good we mean functions which fulfil the resiliency property. To that purpose, we use the classification of Boolean functions of five variables with respect to cryptographic properties done by [BBNP05] which is related to the cosets classification of the Reed-Muller error-correcting code $RM(1,5)$ (recall that a code C is a linear subspace of $\mathbb{F}_2^5$ and for a vector $u \in \mathbb{F}_2^5$, the set $u+C = \{u+x : x \in C\}$ is a *coset* of C and u is called the *coset leader*. The cosets form a partition of $\mathbb{F}_2^5$). The paper by [BBNP05] only gives representatives and the number of Boolean functions fulfilling some cryptographic properties in the equivalence class. More precisely, their classification tells that if there are resilient functions, they have to be in the equivalence class which also contain non-resilient functions. We thus explore these equivalence classes to find out good Boolean functions which can be used as CA rules for generating PR sequences. We also propose to extend those CA rules of radius 2 into Boolean functions of nine variables by selecting the CA rules which preserve the resiliency property. The fact that resiliency is preserved by the iteration of a rule is not true in general. PR sequences generated by these rules are then submitted to the Diehard test suite.

The paper is organised as follows: Section 2 defines the notions of pseudo-random generator, cellular automata and Boolean functions; it also presents some equivalence properties of the Fourier-Hadamard (or Walsh-Hadamard) transform, which is widely used in this study. In Section 3, we propose a study of the radius 2, elementary CA for selecting resilient rules and propose an approach to avoid an exhaustive search among all the 2^{2^5} possible rules. For the rules we have selected, we propose in Section 4 some tests of the randomness for the sequences which can be generated by radius 2, elementary CA.

2   Definitions and notations

This section recalls some basic notation and facts on pseudo-randomness, CAs and Boolean functions.

2.1   *Pseudo-Randomness*

In [Wol02], three mechanisms responsible for random behaviour in systems are described: (1) *Randomness from physics* like brownian motion; (2) *Randomness from the initial conditions* which is studied by chaos theory; and (3) *Randomness by design*, also called pseudo-randomness. Many algorithms generate PR sequences. The behaviour of the system is fully determined by knowing the seed and the algorithm used. They are quicker methods than getting "true" randomness from the environment, inaccessible for computers.

The applications of randomness have led to many different methods for generating random data. These methods may vary as to how unpredictable or statistically random they are, and how quickly they can generate random sequences. Before the advent of computational PR sequences, generating large amount of sufficiently random numbers (important in statistics and physical experimentation) required a lot of work. Results would sometimes be collected and distributed as random number tables.

In the sequel, we will consider *pseudo-random generators* (PRG). This corresponds to a deterministic algorithm which "stretches" a short truly random sequence (the *seed*) into a polynomially longer sequence that appear to be "random" (although they are not). In other words, although the output of a PRG is not really random, it is infeasible to tell the difference. It turns out that pseudorandomness and computational complexity are linked in a fundamental way (see [Gol99] for further details). More practically, this corresponds to the behaviour of random number generators implemented in operating systems. In this case,

the short truly random sequence corresponds to the pseudo-device /dev/random and the output of the PRG to the pseudo-device /dev/urandom for producing more random bits of weaker quality.

2.2   Cellular automata

One-dimensional binary CAs (also called *elementary*) consist of a line of cells taking their states among binary values. For practical implementations, the number of cells is finite. There are two cases: a CA has *periodic boundary conditions* if the cells are arranged in a ring and it has *null boundary conditions* when both extremal cells are continuously fixed to zero. All the cells are finite state machines with an updating function which gives the new state of the cell according to its current state and the current state of its nearest neighbours.

In [Wol86a], it was proposed to use CA to produce PR sequences. He considered one-dimensional binary CA with l cells ($l = 2N + 1$ for $N \in \mathbb{N}$). For a CA, the values of the cells at time $t \geq 0$ are updated synchronously by a Boolean function f with $n = r_1 + r_2 + 1$ variables by the rule $x_i(t + 1) = f(x_{i-r_1}(t), \ldots, x_i(t), \ldots, x_{i+r_2}(t))$. Elementary CA are such that $r_1 = r_2 = 1$. For a fixed t, the sequence of the values $x_i(t)$ for $1 \leq i \leq 2N + 1$, is the *configuration* at time t. It is a mapping $c : [\![1, l]\!] \to \mathbb{F}_2$ (the finite field with two elements) which assigns a state of $\mathbb{F}_2$ to each cell. The initial configuration ($t = 0$) $x_1(0), \ldots, x_l(0)$ is the *seed* of the generator, the sequence $(x_N(t))_t$ is the *output sequence* and, when $r_1 = r_2 = r$, the number r denotes the *radius* of the rule. The *Wolfram numbering* associates a rule number to any one of the 256 elementary CA; it takes the binary expansion of a rule number as the truth table of a 3-variable Boolean function.

2.3   Boolean functions

A Boolean function is a mapping from $\mathbb{F}_2^n$ into $\mathbb{F}_2$. In the sequel, additions in $\mathbb{Z}$ (resp. $\mathbb{F}_2$) will be denoted by $+$ and Σ (resp. $\oplus$ and $\bigoplus$), products by $\times$ and $\prod$ (resp. $.$ and $\prod$). When there is no ambiguity, we denote by $+$ the addition of binary vectors. If x and y are binary vectors, their inner product is $x \cdot y = \sum_{i=1}^n x_i y_i$. A very handy representation of Boolean function is the *algebraic normal form* (ANF):

Definition 1 *A Boolean function f with n variables is represented by a binary polynomial in n variables, called algebraic normal form:* $f(x) = \bigoplus_{u \in \mathbb{F}_2^n} a_u (\prod_{i=1}^n x_i^{u_i})$ $a_u \in \mathbb{F}_2$, u_i *is the i-th projection of u.*

Example 1 *If we consider rule (30), its ANF is $x_1 \oplus x_2 \oplus x_3 \oplus x_2 x_3$ or, more concisely, 1+2+3+23.*

The *degree of the ANF* or *algebraic degree* of f corresponds to the number of variables in the longest term $x_1^{u_1} \ldots x_n^{u_n}$ in the ANF of f. This makes sense thanks to the existence and uniqueness of the ANF. The *Hamming weight* $w_H(f)$ of f is the number of $x \in \mathbb{F}_2^n$ such that $f(x){=}1$. The *Hamming weight* $w_H(x)$ of $x \in \mathbb{F}_2^n$ counts the number of 1-valued coordinates in x. f is *balanced* if $w_H(f) = w_H(1 \oplus f) = 2^{n-1}$.

Definition 2 *Two Boolean functions f and g with n variables are* equivalent *iff*

$$f(x) = g\left((x \cdot A) \oplus a\right) \oplus \left(x \cdot B^T\right) \oplus b, \quad \forall x \in \mathbb{F}_2^n \tag{1}$$

where A is a non-singular binary $n \times n$ matrix, b a binary constant, a and $B \in \mathbb{F}_2^n$.

Let us mention one important tool in the study of Boolean functions. The *Fourier-Hadamard transform* is a linear mapping which maps a Boolean function f to the real-valued function $\widehat{f}(u) = \sum_{x \in \mathbb{F}_2^n} f(x)(-1)^{u \cdot x}$, which describes the *spectrum* of the latter. When applied to $f_\chi(x) = (-1)^{f(x)}$ (the

sign function) the Fourier-Hadamard transform is the *Walsh transform*: $\widehat{f_\chi}(u) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus u \cdot x}$.
Using the fact that $f_\chi(u) = 1 - 2f(u)$, the Fourier-Hadamard transform is:

$$\widehat{f}(u) = \frac{1}{2} \sum_{x \in \mathbb{F}_2^n} (-1)^{u \cdot x} - \frac{1}{2}\widehat{f_\chi}(u) \quad , \tag{2}$$

since $\widehat{f}(u) = \sum_{x \in \mathbb{F}_2^n}(-1)^{u \cdot x}\left(\frac{1 - f_\chi(u)}{2}\right) = \frac{1}{2}\sum_{x \in \mathbb{F}_2^n}(-1)^{u \cdot x} - \frac{1}{2}\sum_{x \in \mathbb{F}_2^n} f_\chi(u)(-1)^{u \cdot x}$. And, by
definition, $\widehat{f_\chi}(u) = \sum_{x \in \mathbb{F}_2^n} f_\chi(u)(-1)^{u \cdot x}$. Using Eq. 2 and, as stated by [Car11], we obtain that
$\widehat{f_\chi}(u) = 2^n \delta_0 - 2\widehat{f}(u)$, where δ_0 denotes the *Dirac symbol* defined by $\delta_0(u) = 1$ if u is the null vector
and $\delta_0(u) = 0$ otherwise.

For two equivalent Boolean functions f and g with n variables, the following property holds:

$$\widehat{f_\chi}(u) = (-1)^{a \cdot A^{-1}(u^t + B^T) + b} \widehat{g_\chi}((u \oplus B)(A^{-1})^T) \ . \tag{3}$$

This property is used by [BBNP05] for counting the number of functions satisfying some cryptographic
properties. The Walsh transform allows us to study the *correlation-immunity* of a function.

Definition 3 *A Boolean function f in n variables is* correlation-immune of order k *$(0 < k < n)$ if, given
any n i.i.d. binary random variables $x_1, \cdots, x_n$ according to a uniform Bernoulli distribution, then
the random variable $Z = f(x_1, \ldots, x_n)$ is independent from any random vector $(x_{i_1}, x_{i_2}, \ldots, x_{i_k})$,
$1 \le i_1 < \cdots < i_k < n$. When f is correlation immune of order k and balanced, it is k-resilient.*

In [XM88], a spectral characterisation of resilient functions was given:

Theorem 1 *A Boolean function f in n variables is k-resilient iff it is balanced and $\widehat{f}(u) = 0$ for all
$u \in \mathbb{F}_2^n$ s.t. $0 < w_H(u) \le k$. Equivalently, f is k-resilient iff $\widehat{f_\chi}(u) = 0$ for all $u \in \mathbb{F}_2^n$ s.t. $w_H(u) \le k$.*

Remark that Theorem 1 concerns both transforms (refer to [Car11] for further details).

Theorem 2 (Siegenthaler Bound) *For a k-resilient $(0 \le k < n - 1)$ Boolean function with n variables,
there is an upper bound for its algebraic degree d: $d \le n - k - 1$ if $k < n - 1$ and $d = 1$ if $k = n - 1$.*

Theorem 2 shows that the algebraic degree of a 1-resilient function with 3 variables is at most one, i.e. the
Boolean function must be linear to be resilient. Linear functions are avoided in cryptography.

2.4 *Some properties of the Fourier-Hadamard transform*

Computing the Fourier-Hadamard transform   We use the Fourier-Hadamard transform from [ER82]
called the *Walsh or Sequency Ordered Transform* $(WHT)_w$. This transform is used to study the CA rules
in order to find the best (non-linear) rules for generating PR sequences. To check the rules, we use the
fast transform algorithm whose time complexity is $O(n \log n)$. It receives as an input an array F of size
2^n which contains the images by the t iterates of the local rule f of all the configurations of n cells (i.e.
$f^t(0), f^t(1), \ldots, f^t(2^n - 1)$). Walsh(F, 0, n) outputs $\widehat{F}$ as $\widehat{f^t}(2^n - 1), \ldots, \widehat{f^t}(0)$.

```
Walsh(F, start, n)
    half = n div 2
    for index = start to start + half - 1 do
        mem = F[index]
```

```
        F[index] -= F[index + half]
        F[index + half] += mem
    endfor
    if half > 1 then
        Walsh(F, start, half)
        Walsh(F, start + half, half)
    endif
end
```

Application to CA rules   We proceed step by step with increasing values of t, which counts the number of times the local rule with 5 variables, supposed to be 1-resilient, is iterated on an initial configuration. In this way, we consider the natural extension of $f : \mathbb{F}_2^5 \to \mathbb{F}_2$ to $f : \mathbb{F}_2^{n+4} \to \mathbb{F}_2^n$ where:

$$f(x_0, \ldots, x_{n+4}) = (y_2, \ldots, y_n) \quad \text{such that } y_j = f(x_{j-2}, x_{j-1}, x_j, x_{j+1}, x_{j+2}), j \in [\![2, n]\!]$$

Using the extended f, one can define the t-th iterate of f which is a function $f^t : \mathbb{F}_2^{4t+1} \to \mathbb{F}_2$ in the natural way. We compute next the maximum absolute value of the Fourier-Hadamard transform of the tth-iterate of f at all the points u of Hamming weight 1 and we select the rules with a minimum spectral value. The computation is repeated with increasing values of t until we can identify rules with flat spectral or relatively small values which are slowly growing in function of t.

Iterates and Fourier-Hadamard transform   In this section we isolate some transformations which preserves resiliency (i.e. the spectral values of the Fourier-Hadamard transform) upon iterations.

We first introduce the *reverse operator* $\Phi : \mathbb{F}_2^m \to \mathbb{F}_2^m$, $\Phi((v_1, \ldots, v_m)) = (v_m, \ldots, v_1)$.

Definition 4 *Let $f : \mathbb{F}_2^{2m+1} \to \mathbb{F}_2$ be the local function of a CA. Then, $f_R(x_{-m}, \ldots, x_0, \ldots, x_m) = f(x_m, \ldots, x_0, \ldots, x_{-m})$ is the* reflection *of f.*

Another basic transformation is given by $\Psi(x) = 1 \oplus x$ for $x \in \mathbb{F}_2$. It corresponds to the negation of the variable and is used for designing the conjugation and the conjugation-reflection introduced in [Wol86b, p. 492]. With some abuse of notation, Ψ is extended to sequences of Boolean variables: for $u = (u_1, u_2, \ldots, u_n)$ with $u_i \in \mathbb{F}_2$, $\Psi(u) = (\Psi(u_1), \Psi(u_2), \ldots, \Psi(u_n))$. Moreover, remark that $\Psi^{-1} = \Psi$.

Definition 5 *Let $f : \mathbb{F}_2^{2m+1} \to \mathbb{F}_2$ be the local function of a CA. Then $f_N(x_{-m}, \ldots, x_0, \ldots, x_m) = \Psi \circ f(\Psi(x_m, \ldots, x_0, x_{-m}))$ is the* negation *of f.*

One can see that for any $t \in \mathbb{N}$, $f^t \circ H = H \circ f_R^t$ for $H = \Psi$ or $H = \Phi$, this property will be useful later.

Lemma 1 *Let $\Xi : \mathbb{F}_2^{2m+1} \to \mathbb{F}_2^{2m+1}$ be 1:1 and $f : \mathbb{F}_2^{2m+1} \to \mathbb{F}_2$ a CA. Then, $w_H(f) = w_H(f \circ \Xi)$.*

Proof: Obvious. □

Proposition 1 shows that resiliency is preserved by the reflection when the local rule is iterated.

Proposition 1 *Let $f : \mathbb{F}_2^{2m+1} \to \mathbb{F}_2$ be the local function of a CA. For any $t \in \mathbb{N}$, let $0 < k \le 2mt + 1$. Then, f_R^t is k-resilient iff f^t is k-resilient.*

Proof: The transformation Φ is bijective. Hence, by Lemma 1, we have $w_H(f_R) = w_H(f \circ \Phi) = w_H(f)$. Since f is balanced, $w_H(f_R) = w_H(\Psi \circ f)$. Now, applying Lemma 1 to $\Psi \circ f$ and using last equation, it holds $w_H(\Psi \circ f) = w_H(\Psi \circ f \circ \Phi) = w_H(\Psi \circ f_R)$. Let $a = B = (0, 0, \ldots, 0)$, $b = 0$ and A the reverse identity matrix. Remark that A is non-singular, then, by using Eq. 3, one obtains $\widehat{(f_R^t)}_\chi(u) = \widehat{f_\chi^t}(u \cdot (A^{-1})^T) = \widehat{f_\chi^t}(u \cdot A) = \widehat{f_\chi^t}(A \cdot u)$ which entails

$$\widehat{f_R^t}(u) = \widehat{f^t}(A \cdot u) \ . \tag{4}$$

Now, assume that f^t is k-resilient. Remark that $w_H(A \cdot u) = w_H(u)$ for any u, therefore, by Theorem 1, if $\widehat{f^t}(u) = 0$ for $0 < w_H(u) \le k$, then, by Eq. 4, $\widehat{f_R^t}(u) = 0$ too. For the converse, just remark that A^2 is the identity transformation and then, by Eq. 4, one finds $\widehat{f_R^t}(\Phi(u)) = \widehat{f^t}(u)$. Therefore if $\widehat{f_R^t}(\Phi(u)) = 0$, we have $\widehat{f^t}(u) = 0$. Since Φ is a bijection we have the thesis. $\qquad\square$

Lemma 2 *Let $f : \mathbb{F}_2^{2m+1} \to \mathbb{F}_2$ be the local function of a CA. For any $t \in \mathbb{N}$, f_N^t is balanced iff f^t is balanced.*

Proof: Assume f^t balanced for some $t \in \mathbb{N}$. By definition of f_N^t, $w_H(f_N^t) = w_H(\Psi \circ f^t \circ \Psi)$. Remark that $\Psi \circ f^t$ is a CA; then by Lemma 1, $w_H(\Psi \circ f^t \circ \Psi) = w_H(\Psi \circ f^t)$. Since f^t is balanced, $w_H(\Psi \circ f^t) = w_H(f^t)$. Finally, observing that Ψ^2 is the identity and by Lemma 1 again, it holds $w_H(f^t) = w_H(\Psi^2 \circ f^t) = w_H(\Psi^2 \circ f^t \circ \Psi) = w_H(\Psi \circ f_N^t)$. For the converse, assume that f_N^t is balanced for some $t \in \mathbb{N}$. Then, $w_H(f_N^t) = w_H(\Psi \circ f_N^t) = w_H(\Psi^2 \circ f^t \circ \Psi) = w_H(f^t \circ \Psi)$. By Lemma 1, $w_H(f^t \circ \Psi) = w_H(f^t)$ and therefore $w_H(f^t) = w_H(f_N^t)$. Again, by Lemma 1, $w_H(\Psi \circ f^t) = w_H(\Psi \circ f^t \circ \Psi) = w_H(f_N^t)$. Hence $w_H(f^t) = w_H(\Psi \circ f^t)$. $\qquad\square$

Proposition 2 *Let $f : \mathbb{F}_2^{2m+1} \to \mathbb{F}_2$ be the local function of a CA. For any $t \in \mathbb{N}$, let $0 < k \le 2mt + 1$. Then, f^t is k-resilient iff f_N^t is k-resilient.*

Proof: Fix $k \in \mathbb{N}$ as in the hypothesis. By Lemma 2, it suffices to prove that $\widehat{f_N}(u) = h(u) \cdot \widehat{f}(u)$ for any $u \in \mathbb{F}_2^{2m+1}$ such that $0 < w_H(u) \le k$ and $h : \mathbb{F}_2^{2m+1} \to \mathbb{R}^+$. Let $A = \mathrm{Id}$, $a = (1, 1, \ldots, 1)$, $b = 1$ and $B = (0, 0, \ldots, 0)$. Then, by using Eq. 3, one obtains $\widehat{(f_N^t)}_\chi(u) = (-1)^{1+a \cdot u}\widehat{f_\chi^t}(u)$ for any $u \in \mathbb{F}_2^{2mt+1}$ with $0 < w_H(u) \le 2mt + 1$. This entails $\widehat{f_N^t}(u) = (-1)^{1+a \cdot u}\widehat{f^t}(u)$. $\qquad\square$

Consider the equivalence relation $\mathcal{R}$ on CA rules such that $f \mathcal{R} g$ iff $g = f_R$ or $g = f_N$ or $g = f_{RN}$. According to [CFMM97], there are $2^{2^m}(6 + 2^{2^m})$ distinct $\mathcal{R}$-classes. Propositions 1 and 2 say that all elements in a class have the same resiliency and hence only one element per class should be tested for studying this property. However the gain obtained by this quotient of the set of local rules is minor. Therefore, ideas for "quickly" spanning the set of interesting local rules are welcome. Section 4 proposes (among other things) to consider affine transformations. Indeed, even if f and its Boolean equivalent, say f_A have the same resilience characteristics, the same is not true, in general, for their iterates.

This is essentially due to the fact that the proofs of this subsection are based on the existence of a bijection ϕ and a transformation τ on the local rules such that for any local rule f, it holds that $\forall t \in \mathbb{N}$, $[\tau(f)]^t \circ \phi = \phi \circ f^t$. This property is not true, in general, when transformations different from the negation or the reflection are considered.

Representative	$\mathcal{N}_{CI(1)}$	$\mathcal{N}_{R(1)}$
12	4840	4120
123	16640	11520
123+14	216 000	133 984
123+14+25	69120	24960
123+145+23	1 029 120	537600
123+145+23+24+35	233 472	96 960

Tab. 1: Number of functions satisfying $CI(1)$ and $R(1)$.

3 Exploring radius 2, 1-resilient elementary CA rules

Unlike 3 and 4-variable Boolean local functions, we will not explore the whole class of radius 2 elementary uniform CA rules. Here, we will use the classification made by [BBNP05]. They propose an efficient algebraic approach to the classification of the affine equivalence classes of the cosets of the first order Reed-Muller error correcting code. Indeed, the study of the properties of Boolean functions is related to the study of Reed-Muller codes. The codewords of the r-th order Reed-Muller code of length 2^n, denoted by $RM(r,n)$ correspond to the truth tables of Boolean functions with degree less or equal to r. [BW72] classified all the 2^{26} cosets of $RM(1,5)$ into 48 equivalence classes under the action of the general affine group $AGL(2,5)$. The method is used to classify with respect to the 48 classes into which the general affine group $AGL(2,5)$ partitions the cosets of $RM(1,5)$. The cryptographic properties considered by [BBNP05] are correlation immunity, resiliency and propagation characteristics as well as their combination.

Tab. 1 is a selection of the representatives of Boolean functions taken out from [BBNP05] which lists the coset leaders and counts the number of Boolean functions in the coset which satisfy 1-resiliency (denoted by $R(1)$) and correlation immunity of first order (denoted by $CI(1)$). In Tab. 1, for a property P, $\mathcal{N}_P$ accounts for the number of Boolean functions in the coset which fulfils P.

Like other authors, we restrict our study to 1-resilient functions since there are only 8 2-resilient Boolean functions with 5 variables. This comes from the following upper bound on the non linearity of f (which is the Hamming distance between f and the class of linear functions): for k-resilient functions of n variables, the non linearity is upper bounded by $2^{n-1} - 2^{k+1}$.

From the original table, we only selected representatives of Boolean functions of degrees 2 and 3 since there is no 1-resilient non-linear Boolean function of degree one. The classification done by [BBNP05] also removes Boolean functions of degree 4 if 1-resiliency is considered. Thus, there are only 6 cosets containing 1-resilient Boolean functions as listed in Tab. 1; 12 is the single representative of functions of degree two and the remaining 5 are all of degree three.

3.1 Finding the rules

From the classification by [BBNP05], representatives of cosets containing Boolean functions fulfilling the property of 1-resiliency were found. In order to complete our program, we have to find which elements in the cosets are 1-resilient. For this, we first explored the elements of the cosets listed in Tab. 1 by considering all the linear combinations of all possible linear/affine functions and by computing the Fourier-Hadamard transform on all those elements in the coset. More precisely, the first step is to generate all elements in the coset. If we denote by $R(x_1, x_2, x_3, x_4, x_5)$ the coset leader, we consider elements of the form $R(x_1, x_2, x_3, x_4, x_5) \oplus (ax_1) \oplus (bx_2) \oplus (cx_3) \oplus (dx_4) \oplus (ex_5) \oplus h$ for a, b, c, d, e, h Boolean,

Coset	1-resilient functions
12	3c3c3cc3 3c3cc33c 3cc33c3c 3cc3c3c3 5a5a5aa5 5a5aa55a 5aa55a5a
	5aa5a5a5 66666699 66669966 66996666 66999999 69696996 69699669
	69966969 69969696 96696969 96699696 96966996 96969669 99666666
	99669999 99996699 99999966 a55a5a5a a55aa5a5 a5a55aa5 a5a5a55a
	c33c3c3c c33cc3c3 c3c33cc3 c3c3c33c
123	66696996 66699669 66966969 66969696 69666699 69669966 69996666
	69999999 96666666 96669999 96996699 96999966 99696969 99699696
	99966996 99969669
123+14	66695aa5 6669a55a 66965a5a 6696a5a5 696655aa 6966aa55 969955aa
	9699aa55 99695a5a 9969a5a5 99965aa5 9996a55a
123+14+25	∅
123+145+23	1eb4663c 1eb499c3 e14b663c e14b99c3
123+145+23+24+35	∅

Tab. 2: 1-resilient Boolean functions in the cosets.

spanning the 2^6 elements of the coset. Then, for each element, we compute the Fourier-Hadamard transform; we next only select the balanced Boolean functions and finally the Boolean functions which are 1-correlation immune among the balanced Boolean functions. That is, among the balanced Boolean functions, all functions with zero spectral values at points whose binary decomposition has a Hamming weight of 1. This first step was done with Mathematica and gave us Tab. 2. We adopted a hexadecimal notation for representing the truth table of the Boolean functions instead of the usual decimal notation. And, from Lemma 1, we do not need to specify the most signifiant bit position anymore.

Reading Tab. 2, we notice that two cosets seem not to contain 1-resilient functions, although listed in [BBNP05] table. The reason for this is that we only explored the cosets and not the equivalence class. Recall that the table by [BBNP05] classifies the 48 equivalence classes of $RM(1,5)$ under the action of $AGL(2,5)$. At first, to check the validity of our approach, we generated the coset elements and not the Boolean functions which could be obtained by the action of $AGL(2,5)$ and which can be generated using Eq. (1). The size of the set of functions to explore is thus quite small. We run the fast transform algorithm on a set containing 6.2^6 elements which has to be compared with the whole set with 2^{32} elements. If we had taken into account the action of $AGL(2,5)$, we should have explored 6 classes among the 48 equivalence classes (a ratio of 1/8) on the whole set, which might be further reduced using results from Section 2.4.

3.2 Testing the iterates

Results from Section 3.1 are used to select rules susceptible of preserving 1-resiliency when they are iterated as a CA local rule, with the same method as in Section 2.4. More precisely, from the set of elementary, radius 2 rules (with a generic element denoted by f) , we consider the natural extension of $f : \mathbb{F}_2^5 \to \mathbb{F}_2$ to $f : \mathbb{F}_2^{n+4} \to \mathbb{F}_2^n$ (with $n > 0$) where: $f(x_1, \ldots, x_{n+4}) = (y_1, \ldots, y_n)$ such that $y_j = f(x_j, x_{j+1}, x_{j+2}, x_{j+3}, x_{j+4}), j \in [\![1, n]\!]$. Using the extended f, one can define the t-th iterate of f which is a function $f^t : \mathbb{F}_2^{4t+1} \to \mathbb{F}_2$. We next test the second iterate for selecting rules preserving 1-resiliency. In other words, we compute the maximum absolute value of the Fourier-Hadamard transform of the tth-iterate of f at all the points u of Hamming weight 1 and we select the balanced rules with a zero Fourier Hadamard transform values at those points (by Theorem 1).

For every f of Tab. 2, we built f^2 and tested its 1-resiliency property. This property is easily checked on the Fourier-Hadamard spectrum $\widehat{f^2}$: a balanced f^2 is 1-resilient if $\forall u \in \mathbb{F}_2^9$ with $w_H(u) = 1$, $\widehat{f^2}(u) = 0$. The spectrum has been computed by the algorithm defined in subsection 2.4 and implemented by a C program. The behaviour of our program was tested by recovering the truth table of the function by computing the inverse transform and by comparing its output with computer algebra systems (sage and mathematica). The results are available in Tab. 3 and shows that few functions (exactly 4 of them) of coset 12 are not 1-resilient, that every function of coset 123 and coset 123+14 preserves 1-resiliency, and that no function of coset 123+145+23 are 1-resilient after 2 iterations. Currently, our experiments are limited to the second iterate since the search of functions which preserve the resiliency upon iterations (greater than the second iterate) requires a long computation time. More complete results in this direction will be presented in the forthcoming long version of this paper.

	0x3C3C3CC3	yes	0x3C3CC33C	no	0x3CC33C3C	no
	0x3CC3C3C3	yes	0x5A5A5AA5	yes	0x5A5AA55A	yes
	0x5AA55A5A	yes	0x5AA5A5A5	yes	0x66666699	yes
	0x66669966	yes	0x66996666	yes	0x66999999	yes
	0x69696996	yes	0x69699669	yes	0x69966969	yes
Coset 12	0x69969696	yes	0x96696969	yes	0x96699696	yes
	0x96966996	yes	0x96969669	yes	0x99666666	yes
	0x99669999	yes	0x99996699	yes	0x99999966	yes
	0xA55A5A5A	yes	0xA55AA5A5	yes	0xA5A55AA5	yes
	0xA5A5A55A	yes	0xC33C3C3C	yes	0xC33CC3C3	no
	0xC3C33CC3	no	0xC3C3C33C	yes		
	0x66696996	yes	0x66699669	yes	0x66966969	yes
	0x66969696	yes	0x69666699	yes	0x69669966	yes
Coset 123	0x69996666	yes	0x69999999	yes	0x96666666	yes
	0x96669999	yes	0x96996699	yes	0x96999966	yes
	0x99696969	yes	0x99699696	yes	0x99966996	yes
	0x99969669	yes				
	0x66695AA5	yes	0x6669A55A	yes	0x66965A5A	yes
Coset 123+14	0x6696A5A5	yes	0x696655AA	yes	0x6966AA55	yes
	0x969955AA	yes	0x9699AA55	yes	0x99695A5A	yes
	0x9969A5A5	yes	0x99965AA5	yes	0x9996A55A	yes
	0x1EB4663C	no	0x1EB499C3	no	0x2D7855F0	no
	0x2D78AA0F	no	0x44EE3C66	no	0x44EEC399	no
Coset 123+145+23	0x4B1ECC69	no	0x77220FAA	no	0x7722F055	no
	0x88DD0FAA	no	0x88DDF055	no	0xB4E13396	no
	0xBB113C66	no	0xBB11C399	no	0xD28755F0	no
	0xD287AA0F	no	0xE14B663C	no	0xE14B99C3	no

Tab. 3: 1-resilient Boolean functions after 2 iterations.

4   PRNG testing

The quality of pseudo-randomness that can be generated from the Boolean functions mentioned above has been evaluated by using the Diehard test suite. It is a widely used tool, especially by cryptographers. The Diehard test suite, developed by Marsaglia from the Florida State University, consists of 17 different tests which have become something which could be considered as a "benchmarking tool" for PR number generators (see [Mar85]). It is meant to evaluate if a stream of numbers is a good PR sequence. It is not necessary to explain how Diehard really works and we refer the reader to [Mar95] for further details. But basically, Diehard uses Kolmogorov-Smirnov normality test to quantifiy the distance between the distribution of a given data set and the uniform distribution; and as the documentation says:

> Each Diehard test is able to provide probability values (p-value) which should be uniformly distributed on $[0, 1)$ if the sequence is made of truly independent bits. Those p-values are obtained by $p = F(X)$ where F is the assumed distribution of the sample random variable X–often normal. But that assumed F is just an asymptotic approximation, for which the fit will be worse in the tail of the distribution. Thus, we should not be surprised with occasional p-values close to 0 or 1. When a stream really fails, one gets p-values of 0 or 1 to six or more places. Otherwise, for each test, its p-value should lie in the interval $(0.025, 0.975)$.

So in order to test our data, we designed a `C` program in which we included the Diehard functions that were slightly modified to fit well with our needs. That is to directly use the results of the CA as a PRG. The 17 different and independent statistical tests require about 16 Mbyte of PR values in binary format.

Our goal was to generate different number sequences from the CA and test them against Diehard. Two different tests were made.

4.1   *Randomness preservation*

In this section we describe the experimentation we made to test if a CA "preserves" the randomness through its dynamics. For this experiment, we consider a CA, whose transition function is $f : \mathbb{F}_2^5 \to \mathbb{F}_2$. Given such a CA, we set up an initial sequence of bits $(b_i)_{i \geq 0}$ that we extract from the `/dev/random` pseudo-device of a MacOSX system[i]. Then we compute the sequence of bits $(b_i')_{i \geq 0}$ such that $\forall i \leq 0$, $b_i' = f(b_{5i}, b_{5i+1}, b_{5i+2}, b_{5i+3}, b_{5i+4})$. To ensure some statistical soundness, for a single CA we build 30[ii] of such sequences from the same entropic source (each sequence being 16 Mbyte long as required by Diehard).

The measure, illustrated in Fig. 1, shows all the distributions of the indicators produced by each single sequence passing all the Diehard tests. And it can be observed that the p-values are well distributed for every data pack. Indeed, there are no accumulation points near zero or one.

This means that the input to the tests is made of independent bits. Thus, we can deduce that these functions are good at preserving the randomness of a given source. Or, in other terms, if we feed a CA with a truly random sequence (obtained by the entropy collector of the BSD kernel) as an input configuration and let the CA run, the output configuration is still PR, according to the Diehard test suite.

---

[i] The entropy collector of the BSD kernel family is considered as a pretty good source of random numbers and MacOSX is built on top of a BSD kernel.

[ii] The repetition of 30 independent experiments comes from statistics. Indeed sample sizes of at least 30 are for many tests considered as "large" and allows a better statistical treatment.
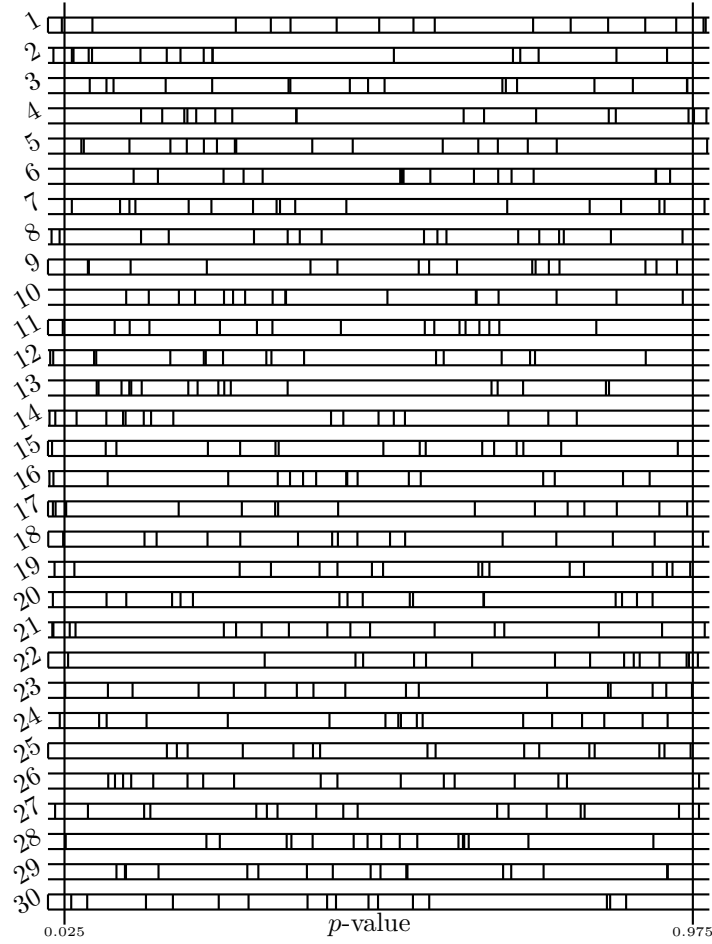
Fig. 1: `0x3C3C3CC3`: distribution of the p-values for each data pack. p-values between the two lines (at 0.025 and 0.975) mean that the corresponding statistical test was successful.
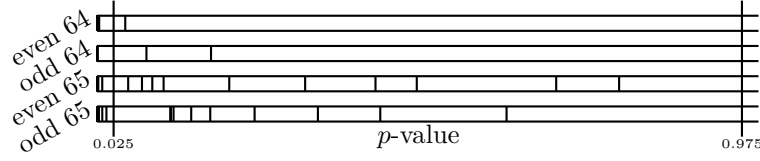
Fig. 2: Distribution of the p-values for the ring CA with rule `0x3C3C3CC3`. p-values between the two lines (at 0.025 and 0.975) mean that the corresponding statistical test was successful, which is not the case for even 64 and odd 64 (all the p-values are almost zero) and barely for even 65 and odd 65.
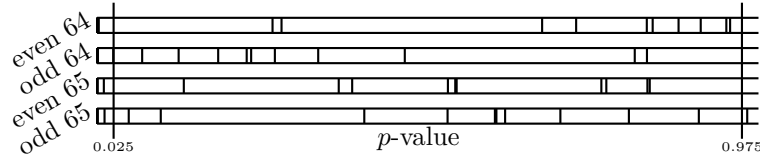
Fig. 3: Distribution of the p-values for the ring CA with rule `0x69999999`. p-values between the two lines (at 0.025 and 0.975) mean that the corresponding statistical test was successful.

4.2   Random number generation

Much more classically, these tests were built to evaluate the possible generation of a good PR sequence by CAs. While it is well known that radius 1 elementary CAs are not suitable for generating PR sequences, it is not impossible to build good PR sequence from simple CAs. As we already tested if the radius-2 functions are good to preserve the randomness of a random source, it would be interesting to consider them as PRNG. So, we tried something very similar to [STCS02].

We set up two rings of cells. Although Wolfram used a ring of 127 cells and Preneel (1993) suggested a ring of 1024 cells to ensure a better quality (both used a slightly different mechanisms for random bit extraction), we use perimeters 64 and 65 as done in [STCS02]. The initial configuration of these rings is of Hamming weight 1. We let the CA iterate about 2 million times. Then, from each configuration obtained, we extract two 32-bits words: the "even" (resp. "odd"), word is built with the state of the first 32 "even" (resp. "odd") cells. The sequences of these "even" (resp. "odd") words constitute two different sequences of 16 Mbyte.

Then, we use Diehard to produce p-values for each test as illustrated in Fig. 2. For the CA with rule `0x3c3c3cc3`, we conclude that this PRNG is not satisfactory. But we were able to find some CAs (like the one with rule `0x69999999` given as an example in Fig. 3) which were able to give much better results against Diehard tests. This suggests that it may be possible to obtain a good PRNG from such a CA.

Acknowledgements

References

[BBNP05]   An Braeken, Yuri Borissov, Svetla Nikova, and Bart Preneel. Classification of boolean functions of 6 variables or less with respect to some cryptographic properties. In *Automata, Languages and Programming*, volume 3580 of *Lecture Notes in Computer Science*, pages 61–61. Springer Berlin / Heidelberg, 2005.

[BW72]   E. Berlekamp and L. Welch. Weight distribution of the cosets of the $(32, 6)$ Reed-Muller code. *IEEE Trans. Inf. Theory*, 18:203–207, 1972.

[Car11]   C. Carlet. Boolean functions for cryptography and error-correcting codes. Technical report, University of Paris 8, 2011.

[CFMM97]   G. Cattaneo, E. Formenti, L. Margara, and G. Mauri. Transformations of the one-dimensional cellular automata rule space. *Parallel Comput.*, 23:1593–1611, November 1997.

[ER82]   D. Elliott and K. Rao. *Fast transforms, algorithms, analysis, applications*. Academic press, 1982.

[Gol99]   O. Goldreich. Pseudorandomness. *Notices of the AMS*, 46:1209–1216, Sep 1999.

[LMS08]   P. Lacharme, B. Martin, and P. Solé. Pseudo-random sequences, boolean functions and cellular automata. In *Proceedings of Boolean Functions and Cryptographic Applications*, 2008.

[Mar85]   G. Marsaglia. A current view of random number generators. In *Computer Sciences and Statistics*, pages 3–10, 1985.

[Mar95]   G. Marsaglia. Diehard. http://www.stat.fsu.edu/pub/diehard/, 1995.

[Mar08]   Bruno Martin. A walsh exploration of elementary ca rules. *J. Cellular Automata*, 3(2):145–156, 2008.

[MS91]   Willi Meier and Othmar Staffelbach. Analysis of pseudo random sequences generated by cellular automata. In *Proceedings of the 10th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'91, pages 186–199, Berlin, Heidelberg, 1991. Springer-Verlag.

[STCS02]   Barry Shackleford, Motoo Tanaka, Richard J. Carter, and Greg Snider. Fpga implementation of neighborhood-of-four cellular automata random number generators. In *Proceedings of the 2002 ACM/SIGDA tenth international symposium on Field-programmable gate arrays*, FPGA '02, pages 106–112, New York, NY, USA, 2002. ACM.

[Wol86a]   Stephen Wolfram. Cryptography with cellular automata. In Hugh Williams, editor, *Advances in Cryptology CRYPTO 85 Proceedings*, volume 218 of *Lecture Notes in Computer Science*, pages 429–432. Springer Berlin / Heidelberg, 1986.

[Wol86b]   Stephen Wolfram. *Theory and applications of cellular automata*. World Scientific, Singapore, 1986.

[Wol02]   Stephen Wolfram. *A New Kind of Science*. Wolfram Media, 2002.

[XM88]   G.-Z. Xiao and J. L. Massey. A spectral characterization of correlation-immune combining functions. *IEEE Trans. on Information Theory*, 34:569–571, 1988.

On the set of Fixed Points of the Parallel Symmetric Sand Pile Model[†]

Kévin Perrot[1][‡] and Thi Ha Duong Phan[2][§] and Trung Van Pham[2][¶]

[1]*LIP (UMR 5668 - CNRS - Université de Lyon - ENS de Lyon) - 46 allé d'Italie 69364 Lyon Cedex 7, France*
[2]*Institute of Mathematics, VAST - 18 Hoang Quoc Viet Road, Cau Giay,10307, Hanoi, Vietnam*

Sand Pile Models are discrete dynamical systems emphasizing the phenomenon of *Self-Organized Criticality*. From a configuration composed of a finite number of stacked grains, we apply on every possible positions (in parallel) two grain moving transition rules. The transition rules permit one grain to fall to its right or left (symmetric) neighboring column if the difference of height between those columns is larger than 2. The model is nondeterministic and grains always fall downward. We propose a study of the set of fixed points reachable in the Parallel Symmetric Sand Pile Model (PSSPM). Using a comparison with the Symmetric Sand Pile Model (SSPM) on which rules are applied once at each iteration, we get a continuity property. This property states that within PSSPM we can't reach every fixed points of SSPM, but a continuous subset according to the lexicographic order. Moreover we define a successor relation to browse exhaustively the sets of fixed points of those models.

Keywords: Discrete Dynamical System, Sand Pile Model, Fixed point

1   Introduction

Sand Pile Models were introduced in 1988 ([BTW88]) to highlight *Self-Organized Criticality* (SOC). SOC characterizes dynamical systems having critical attractors, *i.e.*, systems that evolve toward a stable state from which small perturbations have uncontrolled consequences on the system. This property is straightforward to figure out in the scope of sand pile models : consider a flat table on which we add grains one by one. After a moment, the amount of grains will look like a circular cone which base diameter will continue to grow as we add grains one by one. Some grain additions create avalanches, chain reactions involving numerous grain falls. Some avalanches stop quickly, others continue until they reach the table top. Now remark that whatever the size of the pile is, there will always be one more single grain addition which will increase the base diameter of the cone. So the tiniest possible perturbation —

one single grain addition — can create an unbounded avalanche. This example illustrates the SOC of sand pile models.

There are many variants of sand pile models. All of them consider local grain moving transitions, applied in sequential or parallel mode (one rule application at each iteration or as many rule applications as possible at each iteration), starting from a finite number of stacked grains. The first model, introduced in [CK93], considers one rule applied sequentially : if the difference of height between columns i and $i+1$ is larger than two, then one grain falls from column i to column $i+1$. The set of reachable configurations has a lattice structure and some other interesting properties, see [CMP02] and [LMMP01]. Furthermore its set of reachable configurations can be generated efficiently (see [MM11], [MR10] and [Mas09]). Applying the rule in parallel on every possible column leads to a completely different description of the model, see [DL98]. We can also add one more rule, symmetric to the previous one : if the difference of height between columns i and $i-1$ is larger than two, then one grain can fall from column i to column $i-1$. This leads to SSPM (symmetric sand pile model), studied in [Pha08] and [FMP07].

In [FPPT10], the authors studies PSSPM, the parallel variant of SSPM, and they proved that the form of fixed points of the two models are the same. In this paper, we investigate the set of all fixed points of PSSPM, taking into account their position. We provide a deterministic procedure to reach the extremal (leftmost and rightmost) fixed points of PSSPM according to the total lexicographic order, and prove that any fixed point between these two extremal fixed points reachable in SSPM is also reachable in PSSPM. We also define a successor relation $\triangleleft$ which gives a straightforward way of computing the set of fixed points of PSSPM.

In [RDMDP06] the authors suggest to add rules to get grains also moving forward and backward, to get closer to real life sand piles. [CLM$^+$04] is a survey on sand pile models. An interesting generalization of sand pile models is sand automata, which are powerful enough to simulate cellular automata, see [DGM09], [CFM07] and [CF03].

In this paper, n is a given nonnegative integer.

2   Parallel Symmetric Sand Pile Model

In the theory of discrete dynamical systems, a model is defined by its set of configurations and its transition rule(s). We say that a configuration b is *reachable* from a configuration a if b is obtained from a by a sequence of transitions. In the scope of sand piles, we are interested in the set of configurations reachable from a finite number of stacked grains.

Notation 1 *A configuration c is a sequence of nonnegative integers, with only finitely many positive values. We use an underlined number to denote the position 0 of a sequence. For example $c = (1, 4, \underline{3}, 2, 1)$ is the configuration such that $c_{-2} = 1, c_{-1} = 4, c_0 = 3, c_1 = 2, c_2 = 1$ and for all $i \notin [\![-2; 2]\!], c_i = 0$.*

We now give formal definitions of SSPM and PSSPM.

Definition 1 *SSPM is a discrete dynamical system defined by:*

- *Initial configuration: $(\underline{n})$.*

- *Local left vertical rule $\mathcal{L}$: $(\dots, a_{i-1}, a_i, \dots) \to (\dots, a_{i-1} + 1, a_i - 1, \dots)$ if $a_{i-1} + 2 \leq a_i$.*

- *Local right vertical rule $\mathcal{R}$: $(\dots, a_i, a_{i+1}, \dots) \to (\dots, a_i - 1, a_{i+1} + 1, \dots)$ if $a_i \geq a_{i+1} + 2$.*

> • *Global rule: we apply once the $\mathcal{L}$ rule, or once the $\mathcal{R}$ rule.*

SSPM is a non deterministic and sequential model. PSSPM is defined similarly with the rules applies in parallel on each column:

Definition 2 *PSSPM is a discrete dynamical dynamical system defined with the same initial configuration and local rules as SSPM, and the following global rule:*

> • *Global rule: we apply $\mathcal{L}$ and $\mathcal{R}$ in parallel on every possible column. We apply at most one of the two rules on each column.*

PSSPM is also a non deterministic model, for example from the initial configuration $(\underline{5})$ one has to choose whether applying $\mathcal{L}$ or $\mathcal{R}$ on column 0.

Once the model (SSPM or PSSPM) is fixed, we denote $a \rightarrow b$ when configuration a reduces in one step to configuration b according to the global transition rule. $\rightarrow^*$ denotes the transitive closure of $\rightarrow$. We formally define the sets of *reachable* configurations as:

Notation 2 *SSPM(n)=$\bigcup\{a|(\underline{n}) \rightarrow^* a\}$ is the set of reachable configurations from the initial configuration $(\underline{n})$ by applying SSPM rules.*

PSSPM(n)=$\bigcup\{a|(\underline{n}) \rightarrow^* a\}$ is the set of reachable configurations from the initial configuration $(\underline{n})$ by applying PSSPM rules.

SSPM = $\bigcup\limits_{n \in \mathbb{N}}$ SSPM(n) and PSSPM = $\bigcup\limits_{n \in \mathbb{N}}$ PSSPM(n).

In both models, one can note that any configuration c reachable from the initial configuration $(\underline{n})$ verifies $\sum\limits_{i} c_i = n$ and for some j, $\cdots \leq a_{j-2} \leq a_{j-1} \leq a_j \geq a_{j+1} \geq a_{j+2} \geq \ldots$. This last observation leads to the fact that within PSSPM, there is at most one column j on which a choice between $\mathcal{L}$ and $\mathcal{R}$ happens (such a column j must verify $a_{j-1} < a_j$ and $a_j > a_{j+1}$).

On figure 1 we present in PSSPM the complete set of reachable configurations from $(\underline{5})$. A reachable configuration from which no transition can be applied is a *fixed point*.

A trivial — nevertheless motivating — result is that the set of reachable configurations in PSSPM is a subset of reachable configurations in SSPM:

Proposition 1 *PSSPM $\subsetneq$ SSPM.*

Proof: PSSPM $\subseteq$ SSPM is obvious. Let us show that PSSPM(5) $\subsetneq$ SSPM(5) which leads to the result. Using SSPM global transition rule, $(\underline{5}) \rightarrow (\underline{4}, 1) \rightarrow (\underline{3}, 2) \rightarrow (\underline{3}, 1, 1) \rightarrow (\underline{2}, 2, 1) \rightarrow (1, \underline{1}, 2, 1)$, so $(1, \underline{1}, 2, 1) \in SSPM(5)$. On figure 1 we can see that using PSSPM parallel rule application, $(1, \underline{1}, 2, 1) \notin PSSPM(5)$. □

The set of fixed points of PSSPM is strictly included in the set of fixed points of SSPM (note that it does not hold in the one sided case, where SPM and PSPM have exactly the same fixed points). The following section concentrates on the properties of former compared to latter.
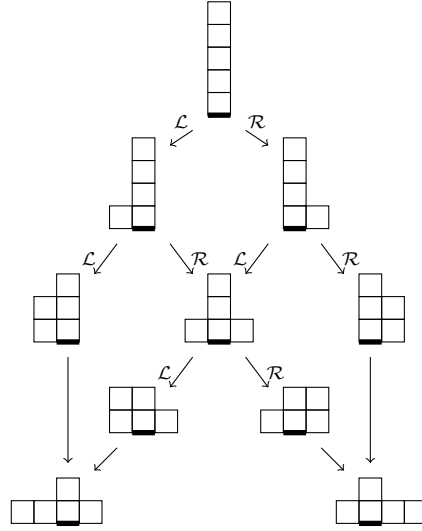
Fig. 1: The set of reachable configurations in PSSPM starting from the initial configuration $(\underline{5})$. A bold line denotes column 0. Edges are labelled according to the choice $\mathcal{L}$ or $\mathcal{R}$, whenever there is one. Two fixed points are reachable from $(\underline{5})$: $(1, 1, \underline{2}, 1)$ and $(1, \underline{2}, 1, 1)$.

3 Fixed points of PSSPM

We propose a study of the set of fixed points of PSSPM. We give a deterministic procedure to reach the rightmost and leftmost fixed points of PSSPM(n), respectively corresponding to the smallest and greatest configurations according to the lexicographic order. We also prove that every fixed point of SSPM(n) between the smallest and greatest fixed point of PSSPM(n) are reachable in PSSPM(n). As a consequence, the set of PSSPM(n) fixed points inherits a kind of continuity property.

The *transition diagram* of PSSPM(n) is the edge-labeled directed multigraph $G_n = (V_n, E_n)$ where $V_n = $ PSSPM(n) is the set of reachable configurations from the initial configuration $(\underline{n})$, and $E_n \subseteq V_n \times V_n \times \{\mathcal{L}, \mathcal{R}\}$ such that $(a, b, \alpha) \in E_n$ if and only if $a \to b$ according to PSSPM rules where we choose (recall that there is at most one choice) to apply the α rule (when there is no choice from a to b, both $(a, b, \mathcal{L})$ and $(a, b, \mathcal{R})$ belong to E_n). Figure 1 is the transition diagram of PSSPM(5), where multiple edges are replaced by a single unlabeled edge.

From a configuration a, we consider the two configurations obtained according to the choices $\mathcal{L}$ and $\mathcal{R}$. Then, we let those two configurations evolve using the same choice at each step. We obtain two sequences of configurations, and we will see that they stay very close, in other words they represent very similar paths within the transition diagram. We introduce a formal notation, $\mathcal{L}(a)$, standing for the configuration obtained by choosing the top grain to fall to the left if possible (if it is not possible, the top grain falls to the right):

Notation 3 *Let a be a configuration such that $a \in V_n$ for some fixed integer n. $\mathcal{L}(a)$ is the configuration defined as:*

1. *if $\exists b$ such that $(a, b, \mathcal{L}) \in E_n$ then $\mathcal{L}(a) = b$,*

2. *else if* $\exists\, b$ *such that* $(a, b, \mathcal{R}) \in E_n$ *then* $\mathcal{L}(a) = b$,

3. *else* $\mathcal{L}(a) = a$.

$\mathcal{R}(a)$ *is defined similarly.*

Let $\omega = \omega_1 \ldots \omega_k$ *be a word over the alphabet* $\{\mathcal{L}, \mathcal{R}\}$, $\omega(a)$ *is the configuration defined inductively as* $\omega(a) = \omega_2 \ldots \omega_k(\omega_1(a))$.

The idea will be to consider a configuration a of PSSPM(n) for a fixed integer n and the two configurations $\mathcal{R}(a)$ and $\mathcal{L}(a)$. Those two configurations are intuitively similar each other. Then we will see that for every word ω over the alphabet $\{\mathcal{L}, \mathcal{R}\}$, the configurations $\omega(\mathcal{R}(a))$ and $\omega(\mathcal{L}(a))$ are also similar according to the relation $\overset{*}{\triangleleft}$ defined below. This is the key argument of our study, stated in Proposition 2. Finally, we use known results about SSPM and further developments to show that when we reach fixed points, the configurations are very similar (see $\triangleleft$ defined below). This leads to Theorem 1, relating the set of fixed points reachable in PSSPM(n) to that reachable in SSPM(n).

Definition 3 *Let* $\Delta(a,b)$ *be the sequence of differences between configurations* a *and* b, $\Delta_i(a,b) = a_i - b_i$. *We define a notion of similarity or closeness between configurations, denoted by the following relations:*

$$
\begin{aligned}
a \triangleleft b &\iff \Delta(a, b) \in 0^* -10^* 10^* \\
a \overset{*}{\triangleleft} b &\iff \Delta(a, b) \in (0^* -10^* 10^*)^*
\end{aligned}
$$

where -1 *is a minus one value. As a convention* $\epsilon = 0^\omega$, *so that* $a = b$ *implies* $a \overset{*}{\triangleleft} b$.

The reader should note that $\overset{*}{\triangleleft}$ is not the reflexive transitive closure of $\triangleleft$, it is just a kind of *non-strict* variant of $\triangleleft$.

The following lemma states the similarity of the configurations obtained when we follow very close paths in the transition diagram of PSSPM(n). The weak relation $\overset{*}{\triangleleft}$ is used to compare obtained configurations all along the evolution toward fixed points. We will see in Proposition 3 that the relation between fixed points can be strengthened into $\triangleleft$.

Proposition 2 *Let* $a \in PSSPM(n)$. *For all* $\omega \in \{\mathcal{L}, \mathcal{R}\}^*$,

$$
\omega(\mathcal{R}(a)) \overset{*}{\triangleleft} \omega(\mathcal{L}(a))
$$

We first present a technical lemma used to avoid some impossible cases in the proof of Proposition 2.

Lemma 1 (technical) *Consider a sequence in PSSPM(n).*

$$
c^1 \to c^2 \to \cdots \to c^k
$$

If there exists a column i *such that*

1. i *remains one of the highest columns i.e.,* $\forall\, 1 \leq t \leq k,\ c_i^t = \max_j c_j^t$

2. $c_i^1 \leq c_{i+1}^1 + 2$ *(resp.* $c_{i-1}^1 + 2 \geq c_i^1$*)*

Then $\forall\, 1 \leq t \leq k,\ c_i^t \leq c_{i+1}^t + 2$ *(resp.* $c_{i-1}^t + 2 \geq c_i^t$*).*

Proof: We proceed by induction on the iterations. The base case is verified according to the second hypothesis. The top column i can't receive any grain during the iterations under consideration so the height difference with column $i+1$ (resp. $i-1$) can only be increased by 1 if i doesn't lose a grain and $i+1$ (resp. $i-1$) loses a grain. In any other case the height difference doesn't increase. So if the height difference is at most 1, then it can't be increased to a difference greater than 2. If the height difference is 2, then column i loses a grain so the height difference doesn't increase.   □

Proof of Proposition 2: We proceed by induction on the length of ω. The base case is obvious : either there is no choice from a to $\mathcal{L}(a)$ and $\mathcal{R}(a)$, and hence $\mathcal{L}(a) = \mathcal{R}(a)$ implies $\mathcal{R}(a) \overset{*}{\lhd} \mathcal{L}(a)$ or there is a choice on column i and hence

- $\mathcal{R}(a)_{i-1} = \mathcal{L}(a)_{i-1} - 1$

- $\mathcal{R}(a)_i = \mathcal{L}(a)_i$

- $\mathcal{R}(a)_{i+1} = \mathcal{L}(a)_{i+1} + 1$

- $\forall j \notin \{i-1, i, i+1\},\ \mathcal{R}(a)_j = \mathcal{L}(a)_j$

so $\mathcal{R}(a) \overset{*}{\lhd} \mathcal{L}(a)$. By induction hypothesis, we are considering two configurations $b = \omega_1 \ldots \omega_{k-1}(\mathcal{R}(a))$ and $c = \omega_1 \ldots \omega_{k-1}(\mathcal{L}(a))$ such that $b \overset{*}{\lhd} c$ and we will now prove that $\omega_k(b) \overset{*}{\lhd} \omega_k(c)$.

For the sake of clarity, we denote d (resp. e) the configuration such that $b \overset{\omega_k}{\to} d$ (resp. $c \overset{\omega_k}{\to} e$).

We do an induction on the columns, and construct $\Delta(d,e)$ from our knowledge on $\Delta(b,c)$, from left to right according to the behavior of each column i in b and c. Considering the rule application on column i of a configuration h gives us three informations:

- does column $i-1$ receive a grain from its right neighbor, denoted $\overleftarrow{h}_{i-1} \in \{0,1\}$;

- does column i give a grain to one of its neighbors, denoted $\overline{h}_i \in \{0,1\}$;

- does column $i+1$ receive a grain from its left neighbor, denoted $\overrightarrow{h}_{i+1} \in \{0,1\}$.

In order to conclude, we will use the fact that

$$\text{for all } j,\ \Delta_j(d,e) = \Delta_j(b,c) + (\overleftarrow{b}_j - \overleftarrow{c}_j) - (\overline{b}_j - \overline{c}_j) + (\overrightarrow{b}_j - \overrightarrow{c}_j)$$

At each step of the induction, we will "update" $\Delta(b,c)$ with the three informations we get and see that it has always the form $(0^* -10^*10^*)^*$. We denote $\Delta^i(b,c)$ the sequence $\Delta(b,c)$ updated up to index i, defined at each index j as

$$\Delta^i_j(b,c) = \begin{cases} \Delta_j(b,c) + (\overleftarrow{b}_j - \overleftarrow{c}_j) - (\overline{b}_j - \overline{c}_j) + (\overrightarrow{b}_j - \overrightarrow{c}_j) = \Delta_j(d,e) & \text{if } j < i \\ \Delta_j(b,c) - (\overline{b}_i - \overline{c}_i) + (\overrightarrow{b}_i - \overrightarrow{c}_i) & \text{if } j = i \\ \Delta_j(b,c) + (\overrightarrow{b}_i - \overrightarrow{c}_i) & \text{if } j = i+1 \\ \Delta_j(b,c) & \text{if } j > i+1 \end{cases}$$

For initialization, there obviously exists an index s such that for all $j \leq s$, there is no grain and hence no rule application on j both in b and c. Therefore $\Delta^s(b, c) = \Delta(b, c) \in (0^* -10^*10^*)^*$.

Let us now eventually prove that for any i, $\Delta^{i-1}(b, c) \in (0^* -10^*10^*)^*$ implies $\Delta^i(b, c) \in (0^* -10^*10^*)^*$. This will complete the proof of the lemma, since there exists an index t such that for all $j \geq t$, there is no grain and hence no rule application on j both in b and c. Therefore $\Delta^t(b, c) = \Delta(d, e)$.

We prove that $\Delta^{i-1}(b, c) \in (0^* -10^*10^*)^*$ implies $\Delta^i(b, c) \in (0^* -10^*10^*)^*$ in three stages: left part, central part and right part. The central part is the set of columns where we apply different local rules in b and c (we will see that there is at most one column in the central part). The left (resp. right) part is the set of columns where grains can only fall to the left (resp. right) both in b and c. The proofs for the left and right parts are symmetric. The central part is more involved and uses lemma 1.

- left part.
  We consider an index i which may be fired to the left or not fired. Since it is not fired to the right, $\overrightarrow{b}_{i+1} - \overrightarrow{c}_{i+1} = 0$ and every index in this part verifies that $\Delta_i^{i-1}(b, c) = \Delta_i(b, c)$. There are 4 cases, some of them are symmetric:

  - i fired to the left in both b and c, then $\Delta^i(b, c) = \Delta^{i-1}(b, c)$.
  - i not fired in both b and c, again $\Delta^i(b, c) = \Delta^{i-1}(b, c)$.
  - i fired to the left in b, not fired in c. Then we have the following changes in $\Delta^i(b, c)$:

  $$\begin{cases} \overleftarrow{b}_{i-1} - \overleftarrow{c}_{i-1} = 1 \\ \overline{b}_i - \overline{c}_i = 1 \\ \overrightarrow{b}_{i+1} - \overrightarrow{c}_{i+1} = 0 \end{cases} \quad \text{hence} \quad \begin{cases} \Delta_{i-1}^i(b, c) = \Delta_{i-1}^{i-1}(b, c) + 1 \\ \Delta_i^i(b, c) = \Delta_i^{i-1}(b, c) - 1 \\ \text{elsewhere there is no change} \end{cases}$$

  but the rule application on i involves that $b_{i-1} + 2 \leq b_i$ and $c_{i-1} + 2 > c_i$. There are 3 different cases according to the values of $(b_i - b_{i-1})$, $\Delta_{i-1}(b, c)$ and $\Delta_i(b, c)$: (for any other set of values we haven't i fired to the left in b and i not fired in c)
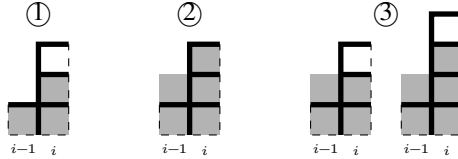
  b is pictured with bold lines, c is pictured in grey. If the difference of height between $i - 1$ and i is greater than 3 in b then it is greater or equal to 2 in c. We recall that $b \overset{*}{\lhd} c$.

  ① $\Delta_{i-1}(b, c) = 0$ and $\Delta_i(b, c) = 1$.
  By induction hypothesis $\Delta^{i-1}(b, c) \in (0^* -10^*10^*)^*$, so we can deduce from the equality $\Delta_i^{i-1}(b, c) = \Delta_i(b, c)$ that $\Delta^{i-1}(b, c)$ around index i is

  $$(\ldots, -1, \ldots, \underset{i}{1}, \ldots, -1, \ldots)$$

  where the right -1 may not exist. Therefore, after applying the changes (adding 1 at index $i - 1$ and subtracting 1 at index i) we still have $\Delta^i(b, c) \in (0^* -10^*10^*)^*$.

② $\Delta_{i-1}(b,c) = -1$ and $\Delta_i(b,c) = 0$.

By induction hypothesis $\Delta^{i-1}(b,c) \in (0^* -10^*10^*)^*$, and we also need that $\Delta^{i-2}(b,c) \in (0^* -10^*10^*)^*$ which is clear according to the base case. We can deduce from the equalities $\Delta^{i-2}_{i-1}(b,c) = \Delta_{i-1}(b,c)$ and for the same reason $\Delta^{i-2}_i(b,c) = \Delta_i(b,c)$ that $\Delta^{i-2}(b,c)$ around index $i-1$ is

$$(\dots, 1, \dots, \underset{i-1}{-1}, \underset{i}{0}, \dots, 1, \dots)$$

where the left 1 may not exist. The part on the right of $i-1$ is not altered by the induction step from $i-2$ to $i-1$, therefore $\Delta^{i-1}(b,c)$ around index i is

$$(\dots, -1, \dots, \underset{i}{0}, \dots, 1, \dots)$$

(it can't be equal to 0^ω for the right 1 is still there). Therefore, after applying the changes (adding 1 at index $i-1$ and subtracting 1 at index i) we still have $\Delta^i(b,c) \in (0^* -10^*10^*)^*$.

③ $\Delta_{i-1}(b,c) = -1$ and $\Delta_i(b,c) = 1$.

The argument is the same as in the case ①.

– i not fired in b, fired to the left in c. This case is symmetric to the previous one.

• central part.

Let us first prove by contradiction that there is at most one column which is fired using different local rules in b and c. We name u and v ($u < v$) the two columns. There are two cases:

– In b, u fires to the left and v fires to the right. Then in c, u fires to the right and v fires to the left. This is impossible since c is an increasing then decreasing sequence.

– In b, both u and v fires to the left. Then the height difference between b_{u-1} and b_v is at least 4. Since $b \overset{*}{\vartriangleleft} c$ the differences between b and c are at most 1 which makes impossible the case where $c_u - c_{v+1} \geq 2$ (necessary condition for u to fire to the right in c).

We now consider the influence of the index i where b and c have opposite behaviors. Let us take $\omega_k = \mathcal{L}$ and consider that i is fired to the left in b and to the right in c (other cases are symmetric). We have the following changes in $\Delta^i(b,c)$:

$$\begin{cases} \overset{\leftarrow}{b}_{i-1} - \overset{\leftarrow}{c}_{i-1} = 1 \\ \overline{b}_i - \overline{c}_i = 0 \\ \vec{b}_{i+1} - \vec{c}_{i+1} = -1 \end{cases} \quad \text{hence} \quad \begin{cases} \Delta^i_{i-1}(b,c) = \Delta^{i-1}_{i-1}(b,c) + 1 \\ \Delta^i_i(b,c) = \Delta^{i-1}_i(b,c) \\ \Delta^i_{i+1}(b,c) = \Delta^{i-1}_{i+1}(b,c) - 1 \\ \text{elsewhere there is no change} \end{cases}$$

but the rule application on i involves that $b_{i-1} + 2 \leq b_i$, $c_{i-1} + 2 > c_i$ (which prevents index i in c to follow the choice $\mathcal{L}$) and $c_i \geq c_{i+1} + 2$. There are 3 cases which can be pictured exactly as in the left part.

① $\Delta_{i-1}(b,c) = 0$ and $\Delta_i(b,c) = 1$.

In this case, $b_{i+1} \leq c_{i+1}$. Since column i in c is fired to the left, $c_i \geq c_{i+1} + 2$, hence $b_i \geq b_{i+1} + 3$ because there is one more grain at i in b.

Also, $\Delta_i(b,c) \neq 0$ so there is one iteration during which a firing of index i has been performed in an ancestor of c and not in the corresponding ancestor of b (in which the height difference between i and $i+1$ was lesser than 2), or there is one iteration during which index i received a grain in an ancestor of b but not in the corresponding ancestor of c (in this case, i became and remains the highest column in the chain leading to b and there exist an iteration where i is not fired, so that it became the only highest, hence the height difference between i and $i+1$ was lesser than 2).

The conditions of lemma 1 are verified and $b_i \geq b_{i+1} + 3$, this case is impossible.

② $\Delta_{i-1}(b,c) = -1$ and $\Delta_i(b,c) = 0$.

By induction hypothesis $\Delta^{i-1}(b,c) \in (0^* {-}10^*10^*)^*$, so we can deduce from the fact that i can't receive any grain (it is obviously one of the top columns of b and c) that $\Delta_i^{i-1}(b,c) = \Delta_i(b,c)$. Moreover, $\Delta_j^{i-1}(b,c)$ for $j > i$ is still equal to $\Delta_j(b,c)$. Let us recall that $b \overset{*}{\lhd} c$ and $\Delta_{i-1}(b,c) = -1$. As a consequence, $\Delta^{i-1}(b,c)$ around index i is

$$(\ldots, -1, \ldots, \underset{i}{0}, \ldots, 1, \ldots)$$

Therefore, after applying the changes (adding 1 at index $i-1$, subtracting 1 at index $i+1$), we still have $\Delta^i(b,c) \in (0^* {-}10^*10^*)^*$.

③ $\Delta_{i-1}(b,c) = -1$ and $\Delta_i(b,c) = 1$.

For the same reason as above, we prove using lemma 1 that this case is impossible.

- right part.

    This part is symmetric to the left part.

We proved that $\Delta^{i-1}(b,c) \in (0^* {-}10^*10^*)^*$ implies $\Delta^i(b,c) \in (0^* {-}10^*10^*)^*$, which concludes the proof that $d \overset{*}{\lhd} e$, which in turn completes the proof of this lemma. □

This lemma states that trying to follow the same transitions conserves the relation $\overset{*}{\lhd}$. It provides a deterministic procedure to reach the extremal fixed points of PSSPM(n):

Notation 4 *We use the symbols $\leq_{lex}$ and $\geq_{lex}$ to denote the lexicographic order over configurations. Note that $a \overset{*}{\lhd} b \Rightarrow a \leq_{lex} b$ and $a \lhd b \Rightarrow a <_{lex} b$.*

Corollary 1 *The maximal — leftmost — (resp. minimal — rightmost —) fixed point of PSSPM(n) according to the lexicographic order is reached when one chooses at every step the $\mathcal{L}$ rule (resp. $\mathcal{R}$ rule).*

Proof: By induction on Proposition 2 and since $a \overset{*}{\lhd} b \Rightarrow a \leq_{lex} b$, we have for all $k \in \mathbb{N}$ and all $w \in \{\mathcal{L}, \mathcal{R}\}^k$ that $\mathcal{L}^k((\underline{n})) \leq w((\underline{n}))$. □

We will now see how the relation $\overset{*}{\lhd}$, used strictly, allows one to browse exhaustively the set of reachable fixed points of SSPM(n) and PSSPM(n).

Proposition 3 *For all fixed a of PSSPM(n) except its leftmost (maximal according to $\leq_{lex}$), there exists a unique fixed point b of PSSPM(n) such that $a \lhd b$.*

Proof: There exists a word u such that $u((\underline{n})) = a$ and from Proposition 2, since a is not the greatest fixed point of PSSPM(n), by incrementally changing letters $\mathcal{R}$ into $\mathcal{L}$ in u until reaching a configuration different from a, we will eventually find a configuration b such that $a \overset{*}{\lhd} b$ and $a \neq b$.

Let us now prove that $a \lhd b$ and that there is no other fixed point c such that $a \lhd c$. By a result from [FMP07] and [Pha08] a fixed point of SSPM (hence of PSSPM) can be cut into two parts which are fixed points of SPM. By a result from [CK93] a fixed point of SPM is a stair (each difference of height is 1) with at most one plateau (two consecutive columns with the same number of grain). As a consequence of those two results, there are at most three plateaus in a (there may be one on the top, which we cut) at positions $(x, x+1)$ for the left plateau, $(y, y+1)$ for the top plateau and $(z, z+1)$ for the right plateau (see figure 2). We have $a \overset{*}{\lhd} b$ and $a \neq b$ so there exists at least one couple of positions (i, j), with $i < j$, such that $a_i = b_i - 1$ and $a_j = b_j + 1$. Let us now see that we can't have more than one such couple of positions, which will prove that $\Delta(a, b) \in 0^* -10^*10^*$. There are 4 positions where we can remove a grain and still respect the plateaus requirement to be a PSSPM fixed point on: x, y, $y+1$ and $z+1$ (if there is no top plateau, we can still remove the top grain), and there are 2 positions where we can add a grain: $x+1$ and z. But if we add a grain at z, we have to remove a grain at a position greater than z (recall that $a \overset{*}{\lhd} b$). The only possible candidate position is $z+1$, leading to a configuration which is not a fixed point since the difference of height between z and $z+1$ becomes greater than 2. Therefore we can only add a grain at $x+1$. Now, where can we remove a grain: only on $z+1$ if there is a plateau at $(z, z+1)$ (otherwise there are two plateaus on the left or right side which is not a SPM fixed point), and only on the rightmost top column if there is no right plateau. This proves that $a \lhd b$ and b is unique.
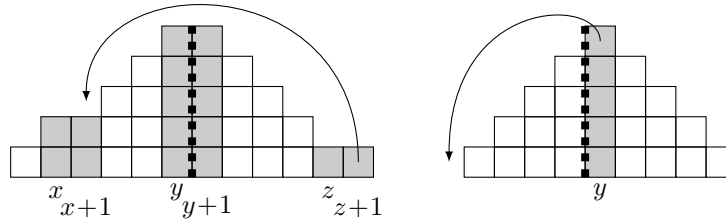
Fig. 2: For any non-maximal fixed point a, there exists a unique fixed point b such that $a \lhd b$.   □

Theorem 1 *Let*

$$\pi_0 <_{lex} \pi_1 <_{lex} \cdots <_{lex} \pi_{k-1} <_{lex} \pi_k$$

be the sequence of all fixed points of PSSPM(n) ordered lexicographically. Then this sequence has the following strong relation:

$$\pi_0 \lhd \pi_1 \lhd \cdots \lhd \pi_{k-1} \lhd \pi_k$$

Moreover, for any fixed point π of SSPM(n) such that $\pi_0 \leq_{lex} \pi \leq_{lex} \pi_k$, there exists an index i, $0 \leq i \leq k$, such that $\pi_i = \pi$.

Proof: n^2 is an upper bound to the number of iterations from the configuration $(\underline{n})$ to a fixed point using PSSPM rules (at each step a grain loses some height). Therefore, the set of fixed points of PSSPM(n) is equal to $\bigcup\limits_{\omega \in \{\mathcal{L},\mathcal{R}\}^{n^2}} \omega((\underline{n}))$ because trying every possibility leads to reaching every possible fixed point.

Starting from the word $s^0 = \mathcal{R}^{n^2}$ and changing one by one the letters $\mathcal{R}$ into $\mathcal{L}$, we get a sequence of words $(s^0, s^1, \ldots, s^{n^2})$ such that for all k, the size of the word s^k is n^2 and the number of occurrences of $\mathcal{L}$ in s^k is k. From Proposition 2, for all $k < n^2$ we have $s^k((\underline{n})) \overset{*}{\lhd} s^{k+1}((\underline{n}))$. There are two possibilities:

- $s^k((\underline{n})) = s^{k+1}((\underline{n}))$.

- $s^k((\underline{n})) \neq s^{k+1}((\underline{n}))$.

In the second case, both configurations are fixed points of PSSPM(n) and from Proposition 3 we have $s^k \lhd s^{k+1}$. This gives a simple procedure to construct the set of fixed points of PSSPM(n) from π_0 to π_k and proves the first part of the theorem (the procedure is described below).

From the SSPM(n) fixed point characterization described in [FMP07] and [Pha08] (presented in the proof of Proposition 3), even if the complete set of reachable fixed points are not the same, the fixed points of SSPM(n) and PSSPM(n) between the smallest and greatest fixed points of PSSPM(n) are the same (the authors of [FMP07] and [Pha08] use exactly the same construction as the one described in the proof of Proposition 3, see figure 2). The fact that PSSPM$(n) \subseteq$ SSPM(n) completes the proof of the second part of the theorem. $\qquad \square$

The proofs of Proposition 3 and Theorem 1 provide a simple algorithm to browse the set of fixed points of PSSPM(n). First compute the minimal (rightmost $\pi^{\mathcal{R}}$) and maximal (leftmost $\pi^{\mathcal{L}}$) fixed points starting from $(\underline{n})$ by following always the same choice ($\mathcal{R}$ to get the minimal configuration, and $\mathcal{L}$ to get the maximal one). Then starting from $\pi^{\mathcal{R}}$, construct the unique fixed point π_1 such that $\pi^{\mathcal{R}} \lhd \pi_1$, as explained on figure 2. From π_1, construct the unique fixed point π_2 such that $\pi_1 \lhd \pi_2$, etc... Until you get $\pi^{\mathcal{L}}$. From what precedes, this deterministic procedure browses exhaustively the set of fixed points of PSSPM(n).

4   Conclusion

We have studied the set of fixed points of PSSPM(n) and compared it to the set of fixed points of SSPM(n) using the natural lexicographic order. We proved the intuitive fact that the greatest fixed point can be reached using always the choice $\mathcal{L}$, and that the smallest fixed point can be reached using always the choice $\mathcal{R}$. More interestingly, we showed that every fixed point reachable in SSPM(n) between the lowest and the greatest fixed points of PSSPM(n) is also reachable in PSSPM(n). This is a kind of continuity property: the set of fixed points reachable in PSSPM(n) is an "interval" of the set of fixed points reachable in SSPM(n).

Further work may concentrate on finding a bound on the maximal and minimal non-empty columns in the set of fixed points of PSSPM(n) which is an open question. The bound $\lfloor \sqrt{2n} \rfloor$ proved in [Pha08] holds for PSSPM(n) but it is not satisfying since proposition 1 states that there are strictly less fixed points in PSSPM(n) than in SSPM(n).

Acknowledgements

References

[BTW88]   P. Bak, C. Tang, and K. Wiesenfeld. Self-organized criticality. *Phys. Rev. A*, 38(1):364–374, 1988.

[CF03]   Julien Cervelle and Enrico Formenti. On sand automata. In Helmut Alt and Michel Habib, editors, *STACS*, volume 2607 of *Lecture Notes in Computer Science*, pages 642–653. Springer, 2003.

[CFM07]   Julien Cervelle, Enrico Formenti, and Benoît Masson. From sandpiles to sand automata. *Theor. Comput. Sci.*, 381(1-3):1–28, 2007.

[CK93]   Eric Goles Ch. and Marcos A. Kiwi. Games on line graphs and sand piles. *Theor. Comput. Sci.*, 115(2):321–349, 1993.

[CLM⁺04]   Eric Goles Ch., Matthieu Latapy, Clémence Magnien, Michel Morvan, and Ha Duong Phan. Sandpile models and lattices: a comprehensive survey. *Theor. Comput. Sci.*, 322(2):383–407, 2004.

[CMP02]   Eric Goles Ch., Michel Morvan, and Ha Duong Phan. Sandpiles and order structure of integer partitions. *Discrete Applied Mathematics*, 117(1-3):51–64, 2002.

[DGM09]   Alberto Dennunzio, Pierre Guillon, and Benoît Masson. Sand automata as cellular automata. *Theor. Comput. Sci.*, 410:3962–3974, September 2009.

[DL98]   Jérôme Olivier Durand-Lose. Parallel transient time of one-dimensional sand pile. *Theor. Comput. Sci.*, 205(1-2):183–193, 1998.

[FMP07]   Enrico Formenti, Benoît Masson, and Theophilos Pisokas. Advances in symmetric sandpiles. *Fundam. Inform.*, 76(1-2):91–112, 2007.

[FPPT10]   E. Formenti, V. T. Pham, T. H. D. Phan, and T. T. H. Tran. Fixed point form of the parallel symmetric sand pile model. *preprint*, 2010.

[LMMP01]   Matthieu Latapy, Roberto Mantaci, Michel Morvan, and Ha Duong Phan. Structure of some sand piles model. *Theor. Comput. Sci.*, 262(1):525–556, 2001.

[Mas09]   Paolo Massazza. A cat algorithm for sand piles. *Pure Mathematics and Applications*, 19:147–158, 2009.

[MM11]   Roberto Mantaci and Paolo Massazza. From linear partitions to parallelogram polyominoes. In *Proceedings of the 15th international conference on Developments in language theory*, DLT'11, pages 350–361, Berlin, Heidelberg, 2011. Springer-Verlag.

[MR10]   Paolo Massazza and Roberto Radicioni. A cat algorithm for the exhaustive generation of ice piles. *RAIRO - Theor. Inf. and Applic.*, 44(4):525–543, 2010.

[Pha08]   Thi Ha Duong Phan. Two sided sand piles model and unimodal sequences. *ITA*, 42(3):631–646, 2008.

[RDMDP06] Dominique Rossin, Enrica Duchi, Roberto Mantaci, and Ha Duong Phan. Bidimensionnal sand pile and ice pile models. In *GASCOM 2006*, Dijon, France, 2006.

.

Bifurcations in Boolean Networks

Chris J. Kuhlman[1,3] and Henning S. Mortveit[1,2†] and David Murrugarra[2] and V. S. Anil Kumar[1,3]

[1]*Network Dynamics and Simulation Science Laboratory, Virginia Tech*
[2]*Department of Mathematics, Virginia Tech*
[3]*Department of Computer Science, Virginia Tech*

This paper characterizes the attractor structure of synchronous and asynchronous Boolean networks induced by bi-threshold functions. Bi-threshold functions are generalizations of standard threshold functions and have separate threshold values for the transitions $0 \rightarrow 1$ (up-threshold) and $1 \rightarrow 0$ (down-threshold). We show that synchronous bi-threshold systems may, just like standard threshold systems, only have fixed points and 2-cycles as attractors. Asynchronous bi-threshold systems (fixed permutation update sequence), on the other hand, undergo a bifurcation. When the difference Δ of the down- and up-threshold is less than 2 they only have fixed points as limit sets. However, for $\Delta \geq 2$ they may have long periodic orbits. The limiting case of $\Delta = 2$ is identified using a potential function argument. Finally, we present a series of results on the dynamics of bi-threshold systems for families of graphs.

Keywords: Boolean networks, graph dynamical systems, synchronous, asynchronous, sequential dynamical systems, threshold, bi-threshold, bifurcation

---

1 Introduction

A standard Boolean *threshold function* $\boldsymbol{t}_{k,m} \colon \{0,1\}^m \longrightarrow \{0,1\}$ is defined by

$$\boldsymbol{t}_{k,m}(x_1,\ldots,x_m) = \begin{cases} 1, & \text{if } \sigma(x_1,\ldots,x_m) \geq k \quad \text{and} \\ 0, & \text{otherwise,} \end{cases} \tag{1.1}$$

where $\sigma(x_1,\ldots,x_m) = \big|\{1 \leq i \leq m \mid x_i = 1\}\big|$. This class of functions is a common choice in modeling biological systems [Kauffman (1969); Karaoz et al. (2004)], and social behaviors (e.g., joining a strike or revolt, adopting a new technology or contraceptives, spread of rumors and stress, and collective action), see, e.g., [Granovetter (1978); Bulger et al. (1989); Macy (1991); Centola and Macy (2007); Watts (2002); Kempe et al. (2003)].

A *bi-threshold function* is a function $\boldsymbol{t}_{i,k^\uparrow,k^\downarrow,m} \colon \{0,1\}^m \longrightarrow \{0,1\}$ defined by

$$\boldsymbol{t}_{i,k^\uparrow,k^\downarrow,m}(x_1,\ldots,x_m) = \begin{cases} \boldsymbol{t}_{k^\uparrow,m}, & \text{if } x_i = 0, \\ \boldsymbol{t}_{k^\downarrow,m}, & \text{if } x_i = 1 \,. \end{cases} \tag{1.2}$$

---

†Email: `Henning.Mortveit@vt.edu` (corresponding author)

Here i denotes a designated argument – later it will be the vertex or cell index. We call $k^\uparrow$ the *up-threshold* and $k^\downarrow$ the *down-threshold*. When $k^\uparrow = k^\downarrow$ the bi-threshold function coincides with a standard threshold function. Note that unlike the standard threshold function in (1.1) which is symmetric, the bi-threshold function is *quasi-symmetric* (or outer-symmetric) – with the exception of index i, it only depends on its arguments through their sum.

In this paper we consider synchronous and asynchronous *graph dynamical systems* (GDSs), see [Mortveit and Reidys (2007); Macauley and Mortveit (2009)], of the form $\mathbf{F}\colon \{0,1\}^n \longrightarrow \{0,1\}^n$ induced by bi-threshold functions. These are natural extensions of threshold GDSs and capture threshold phenomena exhibiting hysteresis properties. Bi-threshold systems are also prevalent in social systems where each individual can change back-and-forth between two states; Schelling states: "Numerous social phenomena display cyclic behavior ...", see (Schelling, 1978, p. 86). Among his examples is whether pick-up volleyball games will continue through an academic semester or die (e.g., individuals regularly choosing to play or not play). One can also look at public health concerns such as obesity, where an individual's back-and-forth decisions to diet or not—which are peer influenced, [Christakis and Fowler (2007)], and therefore can be at least partially described by thresholds—are so commonplace that it has a name: "yo-yo dieting" [Atkinson et al. (1994)]. When $k^\uparrow > k^\downarrow$, a vertex that transitions from state 0 to state 1 is more likely to remain in state 1 than what would be the case in a standard threshold GDS. For the state transitions from 1 to 0 the situation is analogous. This suggests that the cost to change back to state 0 is great or that a change to state 0 will occur only if the conditions that gave rise to the $0 \to 1$ transition significantly diminish. A company that acquires and later divests itself of a competitor is such an example. Examples where $k^\downarrow \geq k^\uparrow$ are commonplace. For example, [Schelling (1978)] states that he often witnesses people who start to cross the street against traffic lights, but will return to the curb if they observe an insufficient number of others following behind. Overshooting, whereby a group of individuals take some action, and within a short time period, a subset of these pull back from it, is also of interest to the sociology community [Bischi and Merlone (2009)] and is characterized by $k^\downarrow \geq k^\uparrow$.

It is convenient to introduce the quantity $\Delta = k^\downarrow - k^\uparrow$. The first of our main results (Theorem 3.1) characterizes limit cycle structure of synchronous bi-threshold GDS (also known as as Boolean networks). Building on the proof for threshold functions in Goles and Olivos (1981), we prove that only fixed points and periodic orbits of length 2 can occur for each possible combination of $k^\uparrow$ and $k^\downarrow$. Since we re-use parts of their proof, and also since their proof only appears in French, a condensed English translation is included in the appendix on page 83. The situation is very different for asynchronous bi-threshold GDSs where a vertex permutation is used for the update sequence. Our second main result states that when $\Delta < 2$, only fixed points can occur as limit cycles. However, for $\Delta \geq 2$ there are graphs for which arbitrary length periodic orbits can be generated. The case $\Delta = 2$ is identified using a potential function argument and represents a (2-parameter) *bifurcation* in a discrete system, a phenomenon that to our knowledge is novel. We also include a series of results for bi-threshold dynamics on special graph classes. These offer examples of asynchronous bi-threshold GDSs with long periodic orbits, and may also serve as building blocks in construction and modeling of bi-threshold systems with given cycle structures.

Paper organization. We introduce necessary definitions and terminology for graph dynamical systems in Section 2. The two main theorems are presented in Sections 3.1 and 3.2. Our collection of results on dynamics for graph classes like trees and cycle graphs follow in Section 4 before we conclude in Section 5.

2   Background and Terminology

In the following we let X denote an undirected graph with vertex set $v[X] = \{1, 2, \ldots, n\}$ and edge set $e[X]$. To each vertex v we assign a state $x_v \in K = \{0, 1\}$ and refer to this as the *vertex state*. Next, we let $n[v]$ denote the sequence of vertices in the 1-neighborhood of v sorted in increasing order and write

$$x[v] = (x_{n[v](1)}, x_{n[v](2)}, \ldots, x_{n[v](d(v)+1)})$$

for the corresponding sequence of vertex states. Here $d(v)$ denotes the degree of v. We call $x = (x_1, x_2, \ldots, x_n)$ the *system state* and $x[v]$ the *restricted state*. The dynamics of vertex states is governed by a list of *vertex functions* $(f_v)_v$ where each $f_v \colon K^{d(v)+1} \longrightarrow K$ maps as

$$x_v(t+1) = f_v\big(x(t)[v]\big) \,.$$

In other words, the state of vertex v at time $t+1$ is given by f_v evaluated at the restricted state $x[v]$ at time t. An *update mechanism* governs how the list of vertex functions assemble to a *graph dynamical system* map (see e.g. Mortveit and Reidys (2007); Macauley and Mortveit (2009))

$$\mathbf{F} \colon K^n \longrightarrow K^n$$

sending the system state at time t to that at time $t+1$.

For the update mechanism we will here use *synchronous* and *asynchronous* schemes. In the former case we obtain Boolean networks where

$$\mathbf{F}(x_1, \ldots, x_n) = (f_1(x[1]), \ldots, f_n(x[n])) \,.$$

This sub-class of graph dynamical systems is sometimes referred to as *generalized cellular automata*. In the latter case we will consider permutation update sequences. For this we first introduce the notion of *X-local functions*. Here the X-local function $F_v \colon K^n \longrightarrow K^n$ is given by

$$F_v(x_1, \ldots, x_n) = (x_1, x_2, \ldots, f_v(x[v]), \ldots, x_n) \,.$$

Using $\pi = (\pi_1, \ldots, \pi_n) \in S_X$ (the set of all permutations of $v[X]$) as an update sequence, the corresponding asynchronous (or sequential) graph dynamical system map $\mathbf{F}_\pi \colon K^n \longrightarrow K^n$ is given by

$$\mathbf{F}_\pi = F_{\pi(n)} \circ F_{\pi(n-1)} \circ \cdots \circ F_{\pi(1)} \,. \tag{2.1}$$

We also refer to this class of asynchronous systems as (permutation) *sequential dynamical systems* (SDSs). The X-local functions are convenient when working with the asynchronous case. In this paper we will consider graph dynamical systems induced by bi-threshold functions, that is, systems where each vertex function is given as

$$f_v = f_{v, k_v^\uparrow, k_v^\downarrow} := \boldsymbol{t}_{v, k_v^\uparrow, k_v^\downarrow, d(v)+1} \,.$$

The phase space of the GDS map $\mathbf{F} \colon K^n \longrightarrow K^n$ is the directed graph with vertex set K^n and edge set $\{(x, \mathbf{F}(x)) \mid x \in K^n\}$. A state x for which there exists a positive integer p such that $\mathbf{F}^p(x) = x$ is a *periodic point*, and the smallest such integer p is the *period* of x. If $p = 1$ we call x a *fixed point* for $\mathbf{F}$. A state that is not periodic is a *transient state*. Classically, the *omega-limit set* of x, denoted by $\omega(x)$, is the accumulation points of the sequence $\{\mathbf{F}^k(x)\}_{k \geq 0}$. In the finite case, the omega-limit set is the unique periodic orbit reached from x under $\mathbf{F}$.

Example 2.1 *To illustrate the above concepts, take $X = \mathrm{Circ}_4$ as graph (shown in Figure 1), and choose thresholds $k^\uparrow = 1$ and $k^\downarrow = 3$. For the synchronous case we have we have for example $\mathbf{F}(1,0,0,1) = (0,1,1,0)$. Using the update sequence $\pi = (1,2,3,4)$ we obtain $\mathbf{F}_\pi(1,0,0,1) = (0,0,1,0)$. The phase spaces of $\mathbf{F}_\pi$ and $\mathbf{F}$ are shown in Figure 1. Notice that $\mathbf{F}_\pi$ has cycles of length 3, while the maximal cycle length of $\mathbf{F}$ is 2.*
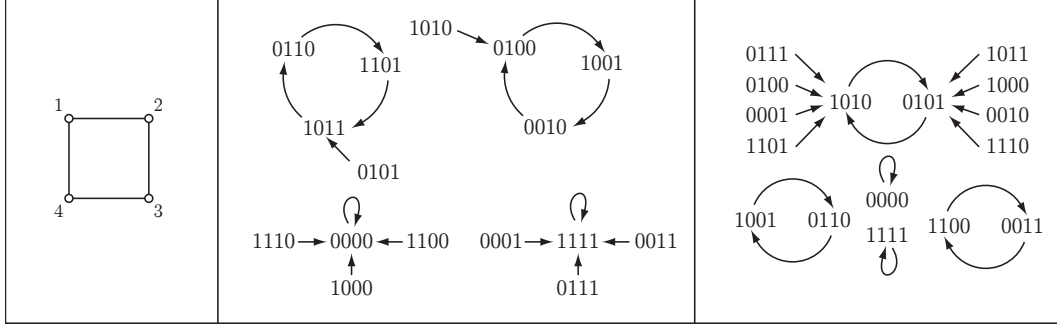
Fig. 1: The graph $X = \mathrm{Circ}_4$ (left), and the phase spaces of $\mathbf{F}_\pi$ (middle) and $\mathbf{F}$ (right) for Example 2.1.

We remark that graph dynamical systems generalize concepts such as cellular automata and Boolean networks, and can describe a wide range of distributed, nonlinear phenomena.

3 ω-Limit Set Structure of Bi-Threshold GDS

This section contains the two main results on dynamics of synchronous and asynchronous bi-threshold GDSs.

3.1   Synchronous Bi-Threshold GDSs

Let $K = \{0, 1\}$ as before, let $A = (a_{ij})$ be a real-valued symmetric matrix, let $(k_i^\uparrow)_{i=1}^n$ and $(k_i^\downarrow)_{i=1}^n$ be vertex-indexed sequences of up- and down-thresholds, and define the function $\mathbf{F} = (f_1, \ldots, f_n)\colon K^n \longrightarrow K^n$ by

$$f_i(x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } x_i = 0 \text{ and } \sum_{j=1}^n a_{ij}x_j \geq k_i^\uparrow \\ 0 & \text{if } x_i = 1 \text{ and } \sum_{j=1}^n a_{ij}x_j < k_i^\downarrow \\ x_i & \text{otherwise.} \end{cases} \tag{3.1}$$

The following theorem is a generalization of Theorem A.1 (see appendix) to the case of bi-threshold functions.

Theorem 3.1 *If $\mathbf{F}$ is the synchronous GDS map over the complete graph of order n with vertex functions as in Equation (3.1), then for all $x \in K^n$, there exists $s \in \mathbb{N}$ such that $\mathbf{F}^{s+2}(x) = \mathbf{F}^s(x)$.*

The proof builds on the arguments of the proof from Goles and Olivos (1981) for standard threshold functions (see page 83 of the appendix). Note that we can use Lemma A.2 in its original form, but

for Lemma A.3 changes are needed to adapt for bi-threshold functions. The position is marked **[Cross-reference for bi-threshold systems]** in the the proof of Lemma A.3 on page 85. Before starting the proof of the theorem above, we first introduce the notion of *bands* and give a result on their structural properties. This is essential in the extension of the original result.

As in the proof of Goles and Olivos (1981) in the appendix, let $z_i \in S$ and assume that $\gamma_i \geq 3$ (the period of the ith component of z). We set

$$\text{supp}(z_i) = \{l \in \{0, 1, 2, \ldots, T - 1\} : z_l = 1\};$$

and use their partition $\mathcal{C} = \{C_0, C_1, C_2, \ldots, C_p\}$. By the assumption $\gamma_i \geq 3$, we are guaranteed that $p \geq 1$. The bi-threshold functions require a more careful structural analysis of the elements of $\mathcal{C}$ than in the case of standard threshold functions. We say that $C \in \mathcal{C}$ is of *type ab* if $C = (l, l+2, l+4, \ldots, k)$ and $z_{l-1} = a$ and $z_{k+1} = b$ where all indices are modulo T. Here we write $m_{ab} = m_{ab}(\mathcal{C})$ for the number of elements of $\mathcal{C}$ of type ab.

We claim that $m_{01} = m_{10}$. Before we prove this, observe first that the sequence $\big(z_i(0), z_i(1), \ldots, z_i(T-1)\big)$ can be split into contiguous (modulo T) sub-sequences (*bands*) whose states contain only isolated 0s, where the end points have state 1, and where bands are separated by sub-sequences of lengths ≥ 2 whose state consist entirely of 0s. By the construction of $\mathcal{C}$, each element $C \in \mathcal{C}$ must be fully contained in a single band. Our claim above is now a direct consequence of the following lemma:

Lemma 3.2 *A band either (i) contains no element C of type* 01 *or* 10*, or (ii) contains precisely one element C of type* 01 *and precisely one element C' of type* 10*.*

Proof: Fix a band B and let $C \in \mathcal{C}$ be the partition containing the first element of B. There are now two possibilities. In the first case, C also contains the final element of B. Then C has type 00, and any other partition element contained in B is necessarily of type 11. In the second case, C terminates before the end of B. The configuration at the end of C must then be as

$$(\ldots 00 \vert 1\_1\_1\_ \ldots \_1\underset{C'}{\underline{101}} \ldots 1 \vert 00 \ldots)$$
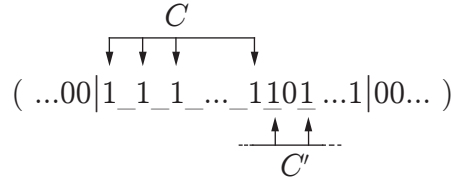
and C is of type 01. The element C' containing the index after the last element of C either goes all the way to the end of B, in which case it is of type 10, or it terminates before that in which case the situation is as in the diagram above and C' is of type 11. By repeated application of this argument, the band B is eventually exhausted with an element C'' of type 10. All other elements of $\mathcal{C}$ within B not included in the sequence of partitions C, C' and so on, must be of type 11, and the proof is complete. □

Corollary 3.3 $m_{01}(\mathcal{C}) = m_{10}(\mathcal{C})$

Proof (Theorem 3.1): Claim: If $\gamma_i \geq 3$ for $z_i \in S$ then $\sum_{j=1}^{n} L(z_i, z_j) < 0$.

We can write

$$\sum_{i=1}^{n} L(z_i, z_j) = \sum_{k=0}^{p}\Big(\sum_{j=1}^{n} a_{ij} \sum_{l\in C_k} \big(z_j(l+1) - z_j(l-1)\big)\Big) = \sum_{k=0}^{p} \Psi_{ik} \, ,$$

where

$$\Psi_{ik} = \sum_{j=1}^{n} a_{ij} \sum_{l\in C_k} \big(z_j(l+1) - z_j(l-1)\big) = \sum_{j=1}^{n} a_{ij} z_j(l_k + 2q_k + 1) - \sum_{j=1}^{n} a_{ij} z_j(l_k - 1) \, .$$

We need to consider Ψ_{ik} for the four types of partition elements. As in the original proof, note that $\Psi_{i0} = 0$.

C_k **is of type** 00**:** in this case $z_i(l_k - 1) = 0$, $z_i(l_k) = 1$, $z_i(l_k + 2q_k + 1) = 0$ and $z_i(l_k + 2q_k + 2) = 0$, which is only possible if

$$\sum_{j=1}^{n} a_{ij} z_j(l_k - 1) \geq k_i^{\uparrow} \quad \text{and} \quad \sum_{j=1}^{n} a_{ij} z_j(l_k + 2q_k + 1) < k_i^{\uparrow} \, ,$$

which implies that $\Psi_{ik} < 0$.

C_k **is of type** 11**:** this case is completely analogous to the 00 case, and again we conclude that $\Psi_{ik} < 0$.

C_k **is of type** 10**:** here $z_i(l_k - 1) = 1$, $z_i(l_k) = 1$, $z_i(l_k + 2q_k + 1) = 0$ and $z_i(l_k + 2q_k + 2) = 0$. This implies that

$$\sum_{j=1}^{n} a_{ij} z_j(l_k - 1) \geq k_i^{\downarrow} \quad \text{and} \quad \sum_{j=1}^{n} a_{ij} z_j(l_k + 2q_k + 1) < k_i^{\uparrow} \, ,$$

leading to $\Psi_{ik} < k_i^{\uparrow} - k_i^{\downarrow}$.

C_k **is of type** 01**:** this case is essentially the same as the 10 case, but here $\Psi_{ik} < k_i^{\downarrow} - k_i^{\uparrow}$.

Using the above four cases, we now have

$$\sum_{j=0}^{n} L(z_i, z_j) = \sum_{k=0}^{p} \Psi_{ik} < 0 + m_{00} \cdot 0 + m_{11} \cdot 0 + m_{10}(k_i^{\uparrow} - k_i^{\downarrow}) + m_{01}(k_i^{\downarrow} - k_i^{\uparrow}) = 0 \, ,$$

where the last equality follows by Corollary 3.3. Clearly, this leads to the same contradiction as in the proof of Theorem A.1. □

An immediate consequence of Theorem 3.1 is the following:

Corollary 3.4 *A synchronous bi-threshold GDS may only have fixed points and 2-cycles as limit sets.*

3.2 Asynchronous Bi-Threshold GDSs

Theorem 3.5 *Let X be a graph, let $\pi \in S_x$ and let $(f_v)_v$ be bi-threshold functions all satisfying $\Delta(v) = k_v^\downarrow - k_v^\uparrow \leq 1$. The sequential dynamical system map $\mathbf{F}_\pi$ only has fixed points as limit sets.*

As before, the graph X is finite. Note also that the per-vertex thresholds $k^\uparrow$ and $k^\downarrow$ need not be uniform for the graph.

Proof: The proof uses a potential function based on a construction in Barrett et al. (2006), but see also Goles-Chacc et al. (1985). For a given state $x \in K^n$ we assign to each vertex the potential

$$P(v, x) = \begin{cases} k_v^\downarrow, & x_v = 1 \\ d(v) + 2 - k_v^\uparrow, & x_v = 0 \ . \end{cases}$$

Note that the quantity $d(v) + 2 - k_v^\uparrow$ is the smallest number of vertex states in the local state $x[v]$ that must be zero to ensure that x_v remains in state zero. Similarly, an edge $e = \{v, v'\}$ is assigned the potential

$$P(e = \{v, v'\}, x) = \begin{cases} 1, & x_v \neq x_{v'} \\ 0, & x_v = x_{v'} \ . \end{cases}$$

For book-keeping, we let $n_i = n_i(v; x)$ denote the number of vertices adjacent to v in state i for $i = 0, 1$ and note that $n_0 + n_1 = d$. The *system potential* $P(x)$ at the state x is the sum of all the vertex and all the edge potentials. For the theorem statement it is clearly sufficient to show that each application of a vertex function that leads to a change in a vertex state causes the system potential to drop.

   Consider first the case where x_v is mapped from 0 to 1 which implies that $n_1 \geq k_v^\uparrow$. Since a change in system potential only occurs for vertex v and edges incident with v, we may disregard the other potentials when determining this change. Denoting the system potential before and after the update by P and P', we have $P = d + 2 - k_v^\uparrow + n_1$ and $P' = k_v^\downarrow + n_0$ which implies that

$$P' - P = k_v^\downarrow + n_0 - d - 2 + k_v^\uparrow - n_1 = k_v^\downarrow + k_v^\uparrow - 2n_1 - 2$$
$$\leq -(k_v^\uparrow - k_v^\downarrow) - 2 = \Delta(v) - 2 \ ,$$

and this is strictly negative whenever $\Delta = k^\downarrow - k^\uparrow \leq 1$. Similarly, for the transition where x_v maps from 1 to 0 one must have $n_1 + 1 \leq k_v^\downarrow - 1$ or $n_1 \leq k_v^\downarrow - 2$. In this case we have

$$P' - P = [d + 2 - k_v^\uparrow + n_1] - [k_v^\downarrow + n_0] = 2n_1 + 2 - k_v^\downarrow - k_v^\uparrow$$
$$\leq 2k_v^\downarrow - 4 + 2 - k_v^\uparrow - k_v^\downarrow = \Delta(v) - 2$$

as before, concluding the proof. □

3.3 Bifurcations in Asynchronous GDS

A natural question now is what happens in the case where $\Delta = k^\downarrow - k^\uparrow = 2$ since periodic orbits are no longer excluded by the arguments in the proof above. The following proposition shows that there are graphs and choices of $k^\uparrow$ and $k^\downarrow$, such that $\Delta = 2$, for which there are periodic orbits of arbitrary length.

Proposition 3.6 *The bi-threshold GDS map over $X = \mathrm{Circ}_n$ with update sequence $\pi = (1, 2, 3, \ldots, n)$, thresholds $k^\uparrow = 1$ and $k^\downarrow = 3$, has cycles of length $n - 1$.*

Proof: We claim that the state $x = (0, 0, \ldots, 0, 1, 0)$ is on an $(n-1)$-cycle. Straightforward computations give that the single 1-state is shifted one position to the left upon each application of $\mathbf{F}_\pi$ until the state $y = (0, 1, 0, \ldots, 0)$ is reached. The image of this state is $z = (1, 0, 0, \ldots, 0, 0, 1)$ which is easily seen to map to x. The smallest number of iterations required to return to the original state x is $n - 1$, producing a cycle as claimed.   □

In other words, by taking Δ as a parameter, we see that the bi-threshold sequential dynamical system undergoes a bifurcation at $\Delta = 2$.

4   Dynamics of Bi-Threshold GDSs

4.1   Graph Unions

From Proposition 3.6, we see that for $X = \mathrm{Circ}_n$ with threshold $k^\uparrow = 1$ and $k^\downarrow = 3$ at each vertex, we obtain an $(n-1)$-cycle for the update sequence $\pi = (1, 2, \ldots, n)$. The following proposition demonstrates how we can combine graphs to obtain larger cycle sizes for bi-threshold SDSs with arbitrarily nonuniform $k^\uparrow, k^\downarrow$. In particular, the result applies to the case where we combine Circ_n graphs where $p = n - 1$ is prime.

Proposition 4.1 *For $i = 1, 2$ let X_i be a graph for which the bi-threshold GDS with update sequence π_i has a cycle in phase space of length c_i. Let $u_i \in \mathrm{v}[X_i]$, and let X be the graph obtained as the disjoint union of X_1 and X_2 plus additionally the vertex $w \notin \mathrm{v}[X_1], \mathrm{v}[X_2]$ with the edges $\{u_1, w\}$ and $\{u_2, w\}$. Moreover, let all thresholds of vertices in X_1 and X_2 be as before, and assign threshold $k^\uparrow = 3$ to w. The bi-threshold SDS map over X with update sequence $\pi = (\pi_1 | \pi_2 | w)$ [juxtaposition] has a cycle of length $\mathrm{lcm}(c_1, c_2)$.*

Proof: Let vertex w have $k^\uparrow = 3$, so that w will never transition to state 1 from state 0. Let $x = (x_1 | x_2 | x_w)$ be the state over X constructed from states x_1 and x_2 on the respective c_i-cycle over X_1 and X_2 with $x_w = 0$. The only vertices whose connectivity, and therefore induced vertex function, are affected by the addition of w are u_1 and u_2. But the state transitions for u_1 and u_2 are unaffected because each is predicated on $\sigma(x[u_1])$ and $\sigma(x[u_2])$, respectively, and these latter two quantities are not altered by the state of w because that state is fixed at 0 by construction. Hence, the phase space of X contains a cycle of length $\mathrm{lcm}(c_1, c_2)$ as claimed.   □

Thus, for $k^\uparrow = 1$ and $k^\downarrow = 3$, there exists a circle graph and permutation π that will produce a cycle in phase space of length three or greater, and multiple circle graphs can be combined to produce graphs with large orbit cycles without modifying the thresholds of vertices in X_1 and X_2.

4.2   Trees

Propositions 3.6 and 4.1 show how periodic orbits of length > 2 arise over graphs that contain cycles. This section investigates bi-threshold SDS maps where X is a tree.

To start, we first recall the notion of κ-equivalence of permutations from Macauley and Mortveit (2009, 2008). Two permutations $\pi, \pi' \in S_X$ are *κ-equivalent* if the corresponding *induced acyclic orientations* O_π and $O_{\pi'}$ of X are related by a sequence of source-to-sink conversions. Here, the orientation O_π
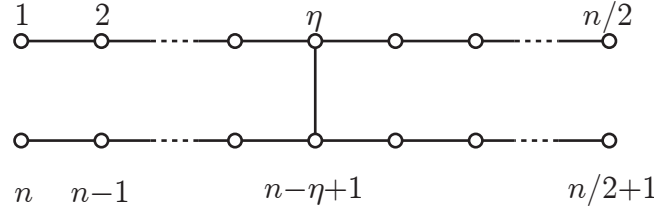
Fig. 2: The tree H_n used in the proof of Proposition 4.2.

is obtained from π by orienting each edge $\{v, v'\} \in \mathrm{e}[X]$ as (v, v') if v precedes v' in π and as (v', v) otherwise. This is an equivalence relation, and it is shown in Macauley and Mortveit (2009) that (i) for a tree the number of κ-equivalence classes is $\kappa(X) = 1$, and (ii) that $\mathbf{F}_\pi$ and $\mathbf{F}_{\pi'}$ have the same periodic orbit structure (up to digraph isomorphism/topological conjugation) whenever π and π' are κ-equivalent. As a result, *we only need to consider a single permutation update sequence* to study the possible periodic orbit structures of permutation SDS maps over a tree X.

The following result shows that there can be cycles of length 3 or greater for permutation SDS over a tree.

Proposition 4.2 *For any integer $c \geq 3$ there is a tree X on $n = 4c - 6$ vertices such that bi-threshold permutation SDS maps over X with thresholds $k^\uparrow = 1$ and $k^\downarrow = 3$ have periodic orbits of length c.*

Proof: An H-tree on $n = 4\beta + 2$ vertices, denoted by H_n, has vertex set $\{1, 2, \ldots, n\}$ and edge set

$$\{\eta, n - \eta + 1\} \cup \big\{\{i, i+1\}, n/2 + \{i, i+1\} \mid 1 \leq i \leq n/2 - 1\big\},$$

where $\eta = \beta + 1$ and $\beta \geq 1$. The graph H_n is illustrated in Figure 4.2.

Set $\beta = c - 2$ so that $n = 4\beta + 2$ and $\eta = \beta + 1$. We take $X = H_n$ as the graph and assign thresholds $(k^\uparrow, k^\downarrow) = (1, 3)$ to all vertices. By the comment preceding Proposition 4.1, we may simply use $\pi = (1, 2, 3, \ldots, n)$ as update sequence since all permutations give cycle equivalent maps $\mathbf{F}_\pi$.

For the initial configuration, set the state of each vertex v in the range $(n/2) + 1 \leq v \leq n - \eta + 1$ (bottom right branch) to 1 and set all other vertex states to 0 so that

$$x(0) = (0, 0, \ldots, 0, \underbrace{1, 1, \ldots, 1,}_{\text{start at vertex } (n/2) + 1} 0, 0, \ldots 0)$$

The number of vertices in a contiguous vertex range with state 1 will always be η; there may be one or two such groups in a system state. The image of $x(0)$ is

$$x(1) = (0, 0, \ldots, 0, \underbrace{1, 1, \ldots, 1,}_{\text{start at vertex } \eta} 0, 0, \ldots 0) \, ,$$

where now the first $\eta - 1$ vertices are in state 0, the next η vertices are in state 1, and the remaining vertices—all those along the bottom arm—are in state 0, as follows. Along the top arm, vertices 1

through $\eta - 1$ will remain in state 0 because all nodes and their neighbors are in state 0. Vertex η, the state of the vertex incident to the crossbar on the top arm, will change to 1 because its neighbor along the crossbar is in state 1. For the given permutation, then, each subsequent vertex v_i in the range $\eta + 1$ through $n/2$ will change to state 1 because $x_{v_{i-1}} = 1$ and $k^\uparrow = 1$. For the bottom arm, vertex $(n/2) + 1$ will change from state 1 to state 0 because $\sigma(x[v_{(n/2)+1}]) = 2 < k^\downarrow$. For the same reason, each vertex v_i in the range $(n/2) + 2$ to $n - \eta + 1$ will transition to state 0. Vertices from $n - \eta + 2$ through n will remain in state 0.

The next state is

$$x(2) = (0, 0, \ldots, 0, \underbrace{1, 1, \ldots, 1}_{\text{start at vertex } \eta-1}, 0, 0, \ldots 0, \underbrace{1, 1, \ldots, 1}_{\text{start at vertex } n-\eta+1}) \,,$$

where, for the top arm, the first $\eta - 2$ vertices are in state 0, the next η vertices are in state 1, and the last vertex on the top arm is in state 0. That is, the set of 1's along the top arm has shifted one vertex left, as follows. Let the set of vertices in the top arm in state 1 (in $x(1)$) be denoted v_i through $v_{i+\eta}$. Vertex v_{i-1} will transition $0 \to 1$ because $x_{v_i} = 1$. Vertex v_i will remain in state 1 because $\sigma(x[v_i]) = 3 = k^\downarrow$. Likewise v_{i+1} through $v_{i+\eta-1}$ will remain in state 1. However, $v_{i+\eta}$ will transition to state 0 because $\sigma(x[v_{i+\eta}]) = 2 < k^\downarrow$. We refer to this behavior as a *left-shift* (the analogous shift to the right is a *right-shift*). For the bottom arm, the η vertices (labels $(n/2) + 1$ through $n - \eta$) remain in state 0. Vertex $n - \eta + 1$ transitions to state 1 because the neighbor along the crossbar is in state 1. Subsequently, vertices $n - \eta + 2$ through n transition to state 1, in turn, according to π.

The next state is

$$x(3) = (0, 0, \ldots, 0, \underbrace{1, 1, \ldots, 1}_{\text{start at vertex } \eta-2}, 0, 0, \ldots 0, \underbrace{1, 1, \ldots, 1}_{\text{start at vertex } n-\eta}) \,,$$

where the set of η vertices in state 1 in the top arm has shifted left, and the set of η vertices in state 1 in the bottom arm has shifted left. The shifting process embodied in the transition from state $x(2)$ to $x(3)$—where there is a group of vertices in state 1 in each of the top and bottom arms—can happen a total of $(\eta - 2)$ times. The state after these $(\eta - 2)$ transitions is

$$x(\eta) = (\underbrace{1, 1, \ldots, 1}_{\text{start at vertex } 1}, 0, 0, \ldots 0, \underbrace{1, 1, \ldots, 1}_{\text{start at vertex } n-2\eta+3}, 0, 0, \ldots, 0) \,.$$

The image of $x(\eta)$ is $x(0)$, the initial state. There are $2 + (\eta - 2) + 1$ state transitions, and we have a limit cycle of length $c = \eta + 1$. $\square$

Of course, the proof does not guarantee that c is the minimal periodic orbit size, nor that H_n is the minimal order tree with a periodic orbit of this length. Additionally, there may be multiple periodic orbits of length c. The following proposition expands on this in the case where $c \geq 5$: there exists a tree of smaller order than H_n that also admits a c-cycle, namely the Y-trees.

Proposition 4.3 *For any integer $c \geq 3$ there is a tree on $n = 3c - 2$ vertices such that bi-threshold permutation SDS maps over this tree with thresholds $k^\uparrow = 1$ and $k^\downarrow = 3$ have periodic orbits of length c.*
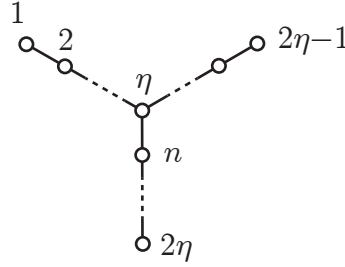
Fig. 3: The tree Y_n used in the proof of Proposition 4.3.

Proof: The proof is analogous to the case of the H-tree. We take as the graph the Y-tree on $n = 3\beta + 1$ vertices (see Figure 3) with $\beta \geq 1$, which has vertex set $\{1, 2, \ldots, n\}$ and, setting $\eta = \beta + 1$, edge set

$$\big\{\{i, i+1\} \mid 1 \leq i \leq 2\eta - 2\big\} \cup \big\{\{i, i+1\} \mid 2\eta \leq i \leq (n-1)\big\} \cup \{\eta, n\} \ .$$

Let $c \geq 3$ with $n = 3c - 2$ so that $X = Y_n$ (and $c = \beta + 1$). We assign thresholds $(k^\uparrow, k^\downarrow) = (1, 3)$ to all vertices and use update sequence $\pi = (1, 2, 3, \ldots, n)$ as before. As the initial configuration, set the states of the β vertices v in the range $\eta \leq v \leq 2\eta - 2$ (all vertices in the upper right branch except $2\eta - 1$) to 1, and set all other vertex states to 0 to form

$$x(0) = (0, 0, \ldots, 0, \underbrace{1, 1, \ldots, 1}_{\text{start at vertex } \eta}, 0, 0, \ldots 0) \ .$$

The image of $x(0)$ is

$$x(1) = (0, 0, \ldots, 0, \quad \underbrace{1, 1, \ldots, 1,}_{\text{start at vertex } (\eta - 1)} \quad 0, 0, \ldots 0, 1) \ ,$$

where now the first $\eta - 2$ vertices are in state 0, the next β vertices are in state 1, and the remaining vertices—except for vertex n—are in state 0. In the upper two branches, the initial set of β nodes in state 1 *shifts left* for the same reasons described in the proof of Proposition 4.2. The last vertex, n, will change to 1 because it is adjacent to vertex η, which has state 1.

The image of $x(1)$ is

$$x(2) = (0, 0, \ldots, 0, \quad \underbrace{1, 1, \ldots, 1,}_{\text{start at vertex } (\eta - 2)} \quad 0, 0, \ldots 0, 1, 1) \ ,$$

where the β nodes in state 1 beginning at vertex $\eta - 2$ have shifted left and vertex $n - 1$ transitions to 1 because vertex n is in state 1. Vertex n remains in state 1 because $\sigma(x[v_n]) = 3$.

The mechanics of the last state transition (the left shift of β vertices and nodes transitioning to state 1 in the lower branch) repeats itself a total of $\beta - 2$ times, at which point the state is

$$x(\beta - 1) = (0, 1, \ldots, 1, \quad \underbrace{0, 0, \ldots, 0,}_{\text{start at vertex } (\eta + 1)} \quad \underbrace{1, 1, \ldots, 1}_{\text{start at vertex } (n - \beta + 2)} \quad) \ ,$$
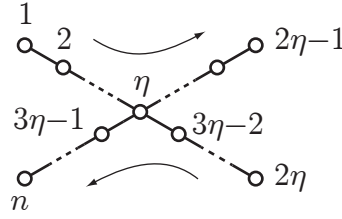
Fig. 4: The tree X_n used in the proof of Proposition 4.4 (arrows indicate vertex labeling order).

where the only vertex in the lower vertical branch in state 0 is 2η, the leaf node.

Noting that vertex η remains in state 1 on the next transition because $\sigma(x[v_\eta]) = 3$, all vertices in the upper right branch transition to 1. Vertex 2η also transitions to 1, giving

$$x(\beta) = (1, 1, \ldots, 1) \ .$$

The next state can be verified to be $x(0)$, thus completing the cycle. The cycle length is therefore $c = \beta + 1$ as stated. □

Interestingly, there is no H-tree nor Y-tree that generates a maximum orbit of size 2 for thresholds $(k^\uparrow, k^\downarrow) = (1, 3)$. However, so-called X-trees (defined below) admit cycles of any size $c \geq 1$.

Proposition 4.4 *For any integer $c \geq 2$ there is a tree X on $n = 4c - 3$ vertices such that bi-threshold permutation GDS maps over X with thresholds $k^\uparrow = 1$ and $k^\downarrow = 3$ have periodic orbits of length c. For $c = 1$, there is a tree X on $n = 5$ vertices that has periodic orbits of length 1 (fixed points).*

Proof: An X-tree on $n = 4\beta + 1$ vertices with $\beta \geq 1$ has vertex set $\{1, 2, \ldots, n\}$ and edge set as illustrated in Figure 4. Here $\eta = \beta + 1$ is the unique vertex of degree 4. Note first that for any n the all-zero state over X_n is a fixed point.

We treat the case $c = 2$ separately; use $X = X_5$, $\pi = (1, 2, 3, 4, 5)$, and $(k^\uparrow, k^\downarrow) = (1, 3)$. It can easily be verified that $x(0) = (0, 1, 1, 0, 0)$ is mapped to $x(1) = (1, 1, 0, 1, 1)$ which in turn is mapped to $x(0)$, constituting a 2-cycle.

Fix $c \geq 3$, set $n = 4c - 3$ and then $c = \beta + 1$, take as the graph $X = X_n$ with thresholds $(k^\uparrow, k^\downarrow) = (1, 3)$ for all vertices, and let $\pi = (1, 2, 3, \ldots, n)$.

Define the initial configuration $x(0)$ by assigning the β vertices v with $\eta \leq v \leq 2\eta - 2$ (all vertices in the upper right branch except $2\eta - 1$) to 1 and set all other vertex states to 0, that is,

$$x(0) = (0, 0, \ldots, 0, \underbrace{1, 1, \ldots, 1}_{\text{start at vertex } \eta}, 0, 0, \ldots 0) \ .$$

The image of $x(0)$ is

$$x(1) = (0, 0, \ldots, 0, \quad \underbrace{1, 1, \ldots, 1}_{\text{start at vertex } (\eta - 1)}, \quad 0, 0, \ldots 0, \quad \underbrace{1, 1, \ldots, 1}_{\text{start at vertex } 3\eta - 2}) \ ,$$

where now the first $\eta - 2$ vertices are in state 0, the next β vertices are in state 1, and the remaining vertices in branch 2 are in state 0. In branch 3, only the vertex neighboring vertex η transitions to state 1, while all vertices in branch 4 transition to state 1 because η is in state 1.

State $x(2)$ is generated by a left-shift of the β contiguous states that are 1 in branches 1 and 2, and by a left-shift of the $\beta + 1$ contiguous state-1 vertices in branches 3 and 4, that is,

$$x(2) = (0, 0, \ldots, 0, \underbrace{1, 1, \ldots, 1}_{\text{start at vertex } (\eta - 2)}, 0, 0, \ldots 0, \underbrace{1, 1, \ldots, 1}_{\text{start at vertex } 3\eta - 3}, 0) \ .$$

From $x(1)$ there are $\beta - 2$ such transitions that result in the state

$$x(\beta - 1) = (0, \underbrace{1, 1, \ldots, 1}_{\text{start at vertex } 2}, 0, 0, \ldots 0, \underbrace{1, 1, \ldots, 1}_{\text{start at vertex } 3\eta - \beta}, 0, 0, \ldots, 0) \ .$$

The next transition results in all vertices in branches 1 and 2 in state 1 since η remains in state 1. The contiguous set of $\beta + 1$ vertices in branches 3 and 4 shift left, giving

$$x(\beta) = (1, 1, \ldots, 1, \underbrace{0, 0, \ldots, 0}_{\text{start at vertex } 3\eta}) \ .$$

The image of $x(\beta)$ is $x(0)$, and, since β is the smallest positive time step with this property, we have established the presence of a periodic orbit of length $c = \beta + 1$. □

Finally, we consider a special class of bi-threshold SDSs on trees with $k^\uparrow = 1$ and $k^\downarrow = k^\downarrow(v) = d(v) + 1$ for each vertex v. Note that the down-threshold for each vertex depends on its degree as indicated by the index v in $k^\downarrow(v)$. We show that such bi-threshold SDS maps always have fixed points. In such systems, the state of a vertex v switches from 0 to 1 if it has at least one neighbor in state 1, and from 1 to 0 if it has at least one neighbor in state 0. This is an interesting contrast to the classes of bi-threshold SDSs on trees discussed above which have large limit cycles.

Let X be a tree. We choose some arbitrary vertex $r \in v[X]$ as its root, and partition X into levels $X_0, X_1, \ldots, X_D$ with respect to r such that $X_0 = \{r\}$, and for any $i \geq 0$, we let X_{i+1} be the set of vertices adjacent to vertices in set X_i, but not in the set $\cup_{j<i} X_j$. We sometimes refer to X_i as level-i set. Let D be the number of levels. We can also define a parent-child relationship relative to this rooted tree, and denote $p(v)$ as the parent of vertex $v \neq r$. In our arguments below, we use any permutation π of $v[X]$, which consists of all the vertices in X_i before those in X_{i-1} for each i. Our result is based on the following property.

Lemma 4.5 *Consider a bi-threshold SDS* $\mathbf{F}_\pi$ *on a tree* X *with an arbitrary root* r *and permutation* π *as defined above where* $k^\uparrow = 1$ *and* $k^\downarrow(v) = d(v) + 1$ *for each vertex* v. *Let* x *be any state vector and* $x' = \mathbf{F}_\pi(x)$. *For each vertex* v *other than the root, we have* $x'_v = x_{p(v)}$.

Proof: Our proof is by induction on the levels, starting from the highest, i.e., X_D. For the base case, consider a leaf $v \in X_D$. We have four cases: $x_v = x_{p(v)} = 1$, $x_v = 0, x_{p(v)} = 1$, $x_v = 1, x_{p(v)} = 0$ and $x_v = x_{p(v)} = 0$. It is easy to verify that in the first two cases, we have $x'_v = 1$ and in the latter two cases, we have $x'_v = 0$, since vertex v is updated before $p(v)$ in π. Therefore, the statement of the lemma holds in the base case for all vertices $v \in X_D$.

Next, consider a vertex v in some level X_j, $j < D$. If v is a leaf in X_j, the lemma follows by exactly the same argument as in the base case. Therefore, consider the case v is not a leaf. Let $w_1, \ldots, w_c$ denote its children. Since level $j + 1$ vertices are updated before those in level j in π, by induction, we have $x'_{w_i} = x_v$ for each w_i. Again, we have a case similar to the base case: when vertex v is updated, it has the same values as its children, and therefore, takes on the state of $p(v)$. Thus, the lemma follows.   □

This property immediately gives us the following:

Corollary 4.6 *Let X be a tree. Let $\pi \in S_x$ and let $(f_v)_v$ be bi-threshold functions satisfying $k^\uparrow = 1$ and $k^\downarrow(v) = d(v) + 1$ for each vertex v. Any SDS map $\mathbf{F}_\pi$ only has fixed points as limit sets.*

Proof: Without loss of generality, we take π to be the permutation in Lemma 4.5. By applying Lemma 4.5, it is easy to verify that for any state vector x, all the vertices in levels 0 and 1 have the same state value in $F(x)$, namely x_r. By induction on i, it is easy to verify that for any $i \geq 1$, all vertices in levels $0, \ldots, i$ have the same state value (of x_r) in $F^i(x)$. The statement follows since all permutations for a tree give cycle equivalent SDS maps.   □

5 Summary and Conclusion

This paper has analyzed the structure of ω-limit sets of bi-threshold GDS. Unlike the synchronous case, bi-threshold SDS maps can have long periodic orbits, and this is characterized in terms of the difference of the up- and down-thresholds. We also analyzed certain classes of trees. The following is a list of questions and conjectures for possible further research.

5.1 *Embedding and Inheritance of Dynamics*

A fundamental question in the study of GDSs is the following: if a graph X has a graph X' as an induced subgraph, what are the relations between the dynamics over the two graphs? Here one has to assume that the vertex function, and update sequences if applicable, are appropriately related. For example, is there a projection from the phase space of the GDS over X to the one over X'?

In initial computational experiments we studied the dynamics for bi-threshold GDS over trees obtained from, e.g. H-trees by adding a collection of edges - results indicate that there are several classes of outcomes. While this is hardly a surprise, there are clear patterns in how edges are added and the dynamics that result. For example, some classes of edge additions give trees that have long periodic orbits just as in the case of H-trees. For other classes of edge additions, however, the addition of even a single edge causes all periodic orbits of size ≥ 2 to disappear. Further insight into the mechanisms involved could shed light on the the fundamental question above.

5.2 *Minimality of Trees with Given Periodic Orbit Sizes*

Our results above on the existence of trees admitting bi-threshold SDS with given periodic orbit sizes are not necessarily minimal. For a given $c \geq 1$ there is an X-tree with a periodic orbit of length c, but there may be a smaller tree (or graph in general) which admits periodic orbits of size c as well. While we have obtained some insight on this via sampling, no firm results have been established.

Note. For all computational experiments involving dynamics of SDS maps over graphs in this paper we used a variant of InterSim (Kuhlman et al., 2011).

Acknowledgements

A Limit Cycle Structure for Standard Threshold Cellular Automata

This appendix section contains a condensed version of the proof from Goles and Olivos (1981) for standard threshold functions. We have incorporated their proof for two reasons. First, only a portion of the original proof needs to be adapted to cover bi-threshold systems, and in this way the paper becomes self-contained. Second, the original proof only appears in French, and we here provide an English version.

Let $K = \{0, 1\}$, let $A = (a_{ij})_{i,j=1}^n$ be a real symmetric matrix, let $\theta = (\theta_1, \ldots, \theta_n) \in \mathbb{R}^n$, and let $\mathbf{F} = (f_1, \ldots, f_n) \colon K^n \longrightarrow K^n$ be the function defined coordinate-wise by

$$f_i(x_1, \ldots, x_n) = \begin{cases} 0, & \text{if } \sum_{j=1}^n a_{ij}x_j < \theta_i \\ 1, & \text{otherwise} . \end{cases} \tag{A.1}$$

Theorem A.1 *For all $x \in K^n$, there exists $s \in \mathbb{N}$ such that $\mathbf{F}^{s+2}(x) = \mathbf{F}^s(x)$.*

The proof of this theorem is based on two lemmas which are given below. Note first that since K^n is finite, for each $x \in K^n$ there exist $s, T \in \mathbb{N}$ (they will generally depend on x) with $T > 0$ such that

$$\mathbf{F}^{s+T}(x) = \mathbf{F}^s(x) \quad \text{and} \quad \mathbf{F}^{s+r}(x) \neq \mathbf{F}^s(x)$$

for all $0 < r < T$. Here s is the transient length of the state x. Next define the $n \times T$ matrix $X(x, T) = (\mathbf{F}^s(x), \ldots, \mathbf{F}^{s+T-1}(x))$ by

$$X(x, T) = \begin{pmatrix} z_1(0) & \ldots & z_1(T-1) \\ \vdots & \ldots & \vdots \\ z_n(0) & \ldots & z_n(T-1) \end{pmatrix} ,$$

where $\mathbf{F}^s(x) = z = (z_1(0), \ldots, z_n(0))$ and $\mathbf{F}^{s+T-1}(x) = (z_1(T-1), \ldots, z_n(T-1))$. In other words, z denotes the first periodic point reached from x (after s steps) and its period is T. The columns of $X(x, T)$ are the T successive periodic points of the cycle containing z.

In general we have

$$\mathbf{F}^{s+l}(x) = (z_1(l), \ldots, z_n(l)) \text{ for } 0 \leq l \leq T - 1 .$$

Since

$$\mathbf{F}^s(x) = \mathbf{F}^{s+T}(x) = \mathbf{F}(z_1(T-1), \ldots, z_n(T-1))$$

we have $z_i(0) = f_i(z_1(T-1), \ldots, z_n(T-1))$, and from $\mathbf{F}^{s+l+1}(x) = \mathbf{F}(\mathbf{F}^{s+l}(x))$ we have

$$z_i(l+1) = f_i(z_1(l), \ldots, z_n(l)) \text{ for } l = 0, \ldots, T-2.$$

We will call z_i the ith row of the matrix $X(x, T)$ and let γ_i denote the smallest divisor of T such that $z_i(l + \gamma_i) = z_i(l)$ for $l \in \{0, \ldots, T-1\}$, and will say that γ_i is the period of the component z_i. Clearly, we have $z_i(l+T) = z_i(l)$ for $i \in \{1, 2, \ldots, n\}$ and all $l \in \{0, \ldots, T-1\}$. Let $S = \{z_1, \ldots, z_n\}$ be the set of rows of $X(x, T)$. We define the operator $L \colon S \times S \to \mathbb{R}$ by

$$L(z_i, z_j) = a_{ij} \sum_{l=0}^{T-1} (z_j(l+1) - z_j(l-1)) z_i(l) \,,$$

with indices taken modulo T.

Lemma A.2 *The operator L has the following properties:*

(i) $L(z_i, z_j) + L(z_j, z_i) = 0$ *for* $i, j \in \{1, \ldots, n\}$ *(anti-symmetry).*

(ii) If $\gamma_i \le 2$ *then* $L(z_i, z_j) = 0$ *for* $j \in \{1, \ldots, n\}$.

Proof: For (i), since $a_{ij} = a_{ji}$, we have

$$L(z_i, z_j) + L(z_j, z_i) = a_{ij} \sum_{l=0}^{T-1} \big([z_i(l) z_j(l+1) - z_i(l-1) z_j(l)]$$
$$+ [z_i(l+1) z_j(l) - z_i(l) z_j(l-1)] \big) \,,$$

which clearly evaluates to zero due to periodicity. For part (ii), if $\gamma_i = 1$ then the row z_i is constant and $L(z_i, z_j) = 0$. If $\gamma_i = 2$ then the value of z_i alternates as

$$z_i(0), z_i(1), z_i(0), z_i(1), \ldots, z_i(0), z_i(1)$$

across the ith row, and the terms in $L(z_i, z_j)$ cancel in pairs. $\square$

Let $z_i \in S$ and suppose in the following that $\gamma_i \ge 3$. We set

$$\mathrm{supp}(z_i) = \{l \in \{0, \ldots, T-1\} : z_i(l) = 1\} \,,$$

and write $\mathcal{I}(l) = \{l, l+2, l+4, \ldots, l-4, l-2\}$. Next, set

$$C_0 = \begin{cases} \varnothing, & \text{if there is no } l_0 \in \{0, \ldots, T-1\} \text{ such that } \mathcal{I}(l_0) \subset \mathrm{supp}(z_i) \\ \mathcal{I}(l_0), & \text{otherwise.} \end{cases}$$

We define C_1 as the set

$$C_1 = \{l_1 + 2s \in \mathrm{supp}(z_i) : s = 0, 1, \ldots, q_1\} \,,$$

where l_1 is the smallest index not in C_0 satisfying $z_i(l_1 - 2) = 0$ and q_1 satisfies $z_i(l_1 + 2q_1 + 2) = 0$. For $k \ge 2$ we define the sets C_k by

$$C_k = \{l_k + 2s \in \mathrm{supp}(z_i) : s = 0, 1, \ldots, q_k\} \,,$$

where $l_k = l_{k-1} + r_k \pmod{T} \notin \{l_1, \ldots, l_{k-1}\}$ is the smallest index for which $z_i(l_k - 2) = 0$ and q_k satisfies $z_i(l_k + 2q_k + 2) = 0$.

Since $\gamma_i \geq 3$ (assumption), there always exists $l_1 \in \operatorname{supp}(z_i)$ for which $z_i(l_1 - 2) = 0$. This allows us to build the collection of sets $\mathcal{C} = \{C_0, \ldots, C_p\}$. By construction, $\mathcal{C}$ is a partition of $\operatorname{supp}(z_i)$. The following lemma provides the final piece needed in the proof of the main result.

Lemma A.3 *For $z_i \in S$ and with $\gamma_i \geq 3$ we have*

$$\sum_{j=1}^{n} L(z_i, z_j) < 0 .$$

Proof: Using the partition $\mathcal{C}$ of $\operatorname{supp}(z_i)$, we have

$$\sum_{j=1}^{n} L(z_i, z_j) = \sum_{j=1}^{n} a_{ij} \sum_{l \in \operatorname{supp}(z_i)} (z_j(l+1) - z_j(l-1)) \cdot 1$$

$$= \sum_{j=1}^{n} a_{ij} \sum_{k=0}^{p} \sum_{l \in C_k} (z_j(l+1) - z_j(l-1)) = \sum_{k=0}^{p} \sum_{j=1}^{n} a_{ij} \sum_{l \in C_k} (z_j(l+1) - z_j(l-1))$$

$$= \sum_{k=0}^{p} \Psi_{ik} ,$$

where we have introduced

$$\Psi_{ik} = \sum_{j=1}^{n} a_{ij} \sum_{l \in C_k} (z_j(l+1) - z_j(l-1)) . \tag{A.2}$$

If $C_0 = \varnothing$ then $\Psi_{i0} = 0$, and if $C_0 = \{l_0, l_0 + 2, \ldots, l_0 - 2\}$ we have

$$\sum_{l \in C_0} (z_j(l+1) - z_j(l-1)) = 0 .$$

In other words, we always have $\Psi_{i0} = 0$, so we assume $k > 0$ in the following. From the assumption that $\gamma_i \geq 3$, there exists $C_k \neq \varnothing$ such that $C_k = \{l_k, l_k + 2, \ldots, l_k + 2q_k\}$, so we can re-write Ψ_{ik} as

$$\Psi_{ik} = \sum_{j=1}^{n} a_{ij} \sum_{s=0}^{q_k} (z_j(l_k + 2s + 1) - z_j(l_k + 2s - 1))$$

$$= \sum_{j=1}^{n} a_{ij} z_j(l_k + 2q_k + 1) - \sum_{j=1}^{n} a_{ij} z_j(l_k - 1) .$$

[Cross-reference for bi-threshold systems] By the construction of C_k, we have $z_i(l_k + 2q_k + 2) = 0$ and $z_i(l_k) = 1$ which, by the definition of f in (A.1), is only possible if

$$\sum_{j=1}^{n} a_{ij} z_j(l_k + 2q_k + 1) < \theta_i, \quad \text{and} \quad \sum_{j=1}^{n} a_{ij} z_j(l_k - 1) \geq \theta_i . \tag{A.3}$$

This implies that $\Psi_{ik} < 0$ and we conclude that

$$\sum_{j=1}^{n} L(z_i, z_j) = \sum_{k=1}^{p} \Psi_{ik} < 0$$

as required. □

Proof of Theorem A.1: From Lemma A.2 we have that L is anti-symmetric so

$$\sum_{i=1}^{n} \sum_{j=1}^{n} L(z_i, z_j) = 0 \ .$$

However, if we assume that $T \geq 3$, then there is z_i with $\gamma_i \geq 3$ and Lemma A.3 produces the desired contradiction. We conclude that $T \leq 2$. □

References

R. Atkinson, W. Dietz, J. Foreyt, N. Goodwin, J. Hill, J. Hirsch, F. Pi-Sunyer, R. Weinsier, R. Wing, J. Hoofnagle, J. Everhart, V. Hubbard, and S. Yanovski. Weight Cycling. *Journal of the American Medical Association*, 272(15):1196–1202, 1994.

C. L. Barrett, H. B. Hunt III, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, and R. E. Stearns. Complexity of reachability problems for finite discrete sequential dynamical systems. *Journal of Computer and System Sciences*, 72:1317–1345, 2006.

G. Bischi and U. Merlone. Global Dynamics in Binary Choice Models with Social Influence. *J. Math. Sociology*, 33:277–302, 2009.

N. Bulger, A. DeLongis, R. Kessler, and E. Wethington. The Contagion of Stress Across Multiple Roles. *Journal of Marriage and the Family*, 51:175–183, 1989.

D. Centola and M. Macy. Complex Contagions and the Weakness of Long Ties. *American J. Sociology*, 113(3):702–734, 2007.

N. Christakis and J. Fowler. The Spread of Obesity in a Large Social Network Over 32 Years. *N. Engl. J. Med.*, pages 370–379, 2007.

E. Goles and J. Olivos. Comportement periodique des fonctions a seuil binaires et applications. *Discrete Applied Mathematics*, 3:93–105, 1981.

E. Goles-Chacc, F. Fogelman-Soulie, and D. Pellegrin. Decreasing energy functions as a tool for studying threshold networks. *Discrete Applied Mathematics*, 12:261–277, 1985.

M. Granovetter. Threshold Models of Collective Behavior. *American J. Sociology*, 83(6):1420–1443, 1978.

U. Karaoz, T. Murali, S. Letovsky, Y. Zheng, C. Ding, C. R. Cantor, and S. Kasif. Whole-genome annotation by using evidence integration in functional-linkage networks. *Proceedings of the National Academy of Sciences*, 101(9):2888–2893, 2004.

S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22:437–467, 1969.

D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the Spread of Influence Through a Social Network. In *Proc. ACM KDD*, pages 137–146, 2003.

C. Kuhlman, V. Kumar, M. Marathe, H. Mortveit, S. Swarup, G. Tuli, S. Ravi, and D. Rosenkrantz. A General-Purpose Graph Dynamical System Modeling Framework. In *Proceedings of the 2011 Winter Simulation Conference (WSC 2011)*, 2011.

M. Macauley and H. S. Mortveit. On enumeration of conjugacy classes of Coxeter elements. *Proceedings of the American Mathematical Society*, 136(12):4157–4165, 2008. doi: 10.1090/S0002-9939-09-09884-0. math.CO/0711.1140.

M. Macauley and H. S. Mortveit. Cycle equivalence of graph dynamical systems. *Nonlinearity*, 22(2):421–436, 2009. doi: 10.1088/0951-7715/22/2/010. math.DS/0709.0291.

M. Macy. Threshold Effects in Collective Action. *American Sociological Review*, 56:730–747, 1991.

H. S. Mortveit and C. M. Reidys. *An Introduction to Sequential Dynamical Systems*. Universitext. Springer Verlag, 2007. ISBN 978-0-387-30654-4. doi: 10.1007/978-0-387-49879-9.

T. Schelling. *Micromotives and Macrobehavior*. W. W. Norton and Company, 1978.

D. Watts. A Simple Model of Global Cascades on Random Networks. *PNAS*, 99(9):5766–5771, 2002.

Asymptotic distribution of entry times in a cellular automaton with annihilating particles

Petr Kůrka[1][†] and Enrico Formenti[2][‡] and Alberto Dennunzio[23][§]

[1]*Center for Theoretical Study, Academy of Sciences and Charles University in Prague, Jilská 1, CZ-11000 Praha 1, Czechia*
[2]*Laboratoire I3S, Université Nice Sophia Antipolis, 2000, route des Lucioles, Les Algorithmes - bât Euclide B, BP 121, 06903 Sophia Antipolis - Cedex, France*
[3]*Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano–Bicocca, Viale Sarca 336, 20126 Milano (Italy)*

This work considers a cellular automaton (CA) with two particles: a stationary particle 1 and left-going one $\overline{1}$. When a $\overline{1}$ encounters a 1, both particles annihilate. We derive asymptotic distribution of appearence of particles at a given site when the CA is initialized with the Bernoulli measure with the probabilities of both particles equal to $1/2$.

Keywords: Cellular Automata, Particle Systems, Entry Times, Return Times

1 Introduction

Cellular automata are a simple formal model for complex systems. They consist of an infinite number of identical finite automata arranged over a regular lattice (here $\mathbb{Z}$). Each automaton updates its state according to its own state and the one of a fixed set of neighboring automata according to a local rule. All updates are synchronous.

The simplicity of the model contrasts with the great variety of different dynamical behaviors. Indeed, exactly this rich variety of behaviors and the ease of being simulated on computers made CA fortune. Actually, they are used in almost all scientific disciplines ranging from Mathematics to Computer Science and Natural Sciences. In particular, in Biology, Physics and Economics, they can be used as a discrete counterpart (in the sense of time) of interacting particle systems (IPS).

The advantage of modeling IPS by CA is that one can have information not only about limit distributions and particle densities but also on their spatial distribution.

On the other hand, as we have already mentioned, the dynamical behavior of CA is complex and not fully understood and IPS can help to understand the dynamics of some CA whenever it can be described

in terms of particles or signals that move in a neutral background and interact on encounters. The general concept of a signal or particle (in the context of CA) has been elaborated in Formenti and Kůrka (2007).

The simplest kind of particles interaction is the annihilation. The classical example is "Just gliders" studied in Gilman (1987). This system consists of a left-going particle $\overline{1}$ and a right-going particle 1 which annihilate on encounters. If the system starts in a Bernoulli measure with equal probabilities of both particles, then at a specified site both kinds of particles keep appearing with probability one, although their appearance is more and more rare as it has been shown by Kůrka and Maass (2002). Other peculiar particle systems and related CA models have been studied, see for example Fisch (1990).

In the present paper we address the question of how the time of appearance of a particle depends on the age of the system. We work with a simpler system called *asymmetric gliders* consisting of one stationary and one left-going particles annihilating on encounters. We show that the appearance of left-going particles time scales linearly with the age of the system and we derive the limit scaled distribution.

The paper is organized as follows. Section 2 and 3 introduce the symmetric and asymmetric gliders CA, respectively. Results are in Section 3. Since the proofs of the main results require several technical lemmata and specific notations, we grouped them in Section 4. The final section draws some conclusions and give some ideas for future work.

2 Symmetric gliders

Let A be a finite alphabet. A *1D CA configuration* is a function from $\mathbb{Z}$ to A. The *1D CA configuration set* $A^{\mathbb{Z}}$ is usually equipped with the metric d defined as follows

$$\forall c, c' \in A^{\mathbb{Z}}, \ d(c, c') = 2^{-n}, \ \text{where } n = \min\left\{i \geq 0 \ : \ c_i \neq c'_i \text{ or } c'_{-i} \neq c'_{-i}\right\} \ .$$

If A is finite, $A^{\mathbb{Z}}$ is a compact, totally disconnected and perfect topological space (i.e., $A^{\mathbb{Z}}$ is a Cantor space). For any pair $i, j \in \mathbb{Z}$, with $i \leq j$, and any configuration $x \in A^{\mathbb{Z}}$ we denote by $x_{[i,j]}$ the word $x_i \cdots x_j \in A^{j-i+1}$, i.e., the portion of c inside the interval $[i, j]$. In the previous notation, $[i, j]$ can be replaced by $[i, j)$ with the obvious meaning. A *cylinder* of block $u \in A^k$ and position $i \in \mathbb{Z}$ is the set $[u]_i = \{x \in A^{\mathbb{Z}} : x_{[i,i+k-1]} = u\}$. Cylinders are clopen sets w.r.t. the metric d and they form a basis for the topology induced by d.

A *1D CA* is a structure $\langle 1, A, r, f \rangle$, where A is the alphabet, $r \in \mathbb{N}$ is the *radius* and $f : A^{2r+1} \to A$ is the *local rule* of the automaton. The local rule f induces a *global rule* $F : A^{\mathbb{Z}} \to A^{\mathbb{Z}}$ defined as follows,

$$\forall c \in A^{\mathbb{Z}}, \forall i \in \mathbb{Z}, \quad F(c)_i = f(c_{i-r}, \ldots, c_i, \ldots, c_{i+r}) \ .$$

In Gilman (1987), Gilman introduced a CA called *Just Gliders* (or *Symmetric Gliders*) which is formally defined as $\langle 1, \{\overline{1}, 0, 1\}, 1, g \rangle$ where $g : \{\overline{1}, 0, 1\}^3 \to \{\overline{1}, 0, 1\}$ is such that

$$\forall (x, y, z) \in \{\overline{1}, 0, 1\}^3, \quad g(x, y, z) = \begin{cases} 1 & \text{if } x = 1, \ y \geq 0 \text{ and } y + z \geq 0 \\ \overline{1} & \text{if } z = \overline{1}, \ y \leq 0 \text{ and } x + y \leq 0 \\ 0 & \text{otherwise} \ . \end{cases}$$

In this context a symbol 1 (resp. $\overline{1}$) is interpreted as a right-going (resp. left-going) particle and 0 is the neutral background. Figure 1 shows an example of evolution of Just Gliders from a random initial configuration.

Fig. 1: Symmetric gliders.

Consider a Bernoulli measure on $\left\{\overline{1},0,1\right\}^{\mathbb{Z}}$, i.e., a sequence of independent identically distributed random variables $X = (X_i)_{i\in\mathbb{Z}}$ over $\left\{\overline{1},0,1\right\}$ such that $\forall i \in \mathbb{Z}, \mathbb{P}[X_i = \overline{1}] = \mathbb{P}[X_i = 1] = p$, $\mathbb{P}[X_i = 0] = 1 - 2p = q$. Then, for any CA global rule F and any $n \in \mathbb{N}$, $F^n(X)_0$ is also a random variable whose distribution depends on the initial distribution of X.

Definition 1 (Entry time) *For $a \in \left\{\overline{1},0,1\right\}$, the* entry time *into $[a]_0$ (appearance of a particle a) after time n at position 0 is*

$$T_n^a(X) = \min\left\{k \geq 0:\ F^{n+k}(X)_0 = a\right\}\ .$$

Since F commutes with σ, the entry times at any position $s \in \mathbb{Z}$ have the same distribution as T_n^a.

In Gilman (1987), the following result has been proven.

Theorem 1 *Let F be the global function of Just Gliders CA. If $\mathbb{P}[X_i = 1] > \mathbb{P}[X_i = \overline{1}]$ then*

$$\mathbb{P}[\forall n \in \mathbb{N}, \exists k \in \mathbb{N}\ s.t.\ F^{n+k}(X)_0 = \overline{1}] = \mathbb{P}[\forall n \in \mathbb{N},\ T_n^{\overline{1}}(X) < \infty] = 0\ .$$

Then Kůrka and Maass (2002) proved the following

Theorem 2 *Let F be the global function of Just Gliders CA. If $\mathbb{P}[X_i = 1] = \mathbb{P}[X_i = \overline{1}]$ then*

1. *$\mathbb{P}[\forall n \in \mathbb{N}, \exists k \in \mathbb{N}\ s.t.\ F^{n+k}(X)_0 \neq 0] = 1$;*

2. *$\mathbb{P}[\forall n \in \mathbb{N},\ T_n^a(X) < \infty] = 1$ for $a \in \left\{\overline{1},1\right\}$;*

3. *$\forall n \in \mathbb{N},\ \mathbb{P}[T_n^a(X) < \infty] = 1$ for $a \in \left\{\overline{1},1\right\}$;*

4. *$\lim_{n\to\infty} \mathbb{P}[F^n(X)_0 = 0] = 1$.*

3 Asymmetric gliders

In this paper we consider a similar CA that we call *Asymmetric Gliders*, $\langle 1, \left\{\overline{1},0,1\right\}, 1, f\rangle$ and $f : \left\{\overline{1},0,1\right\}^3 \to \left\{\overline{1},0,1\right\}$ is defined as follows

$$\forall(x,y,z) \in \left\{\overline{1},0,1\right\}^3, \quad f(x,y,z) = \begin{cases} 1 & \text{if } y = 1 \text{ and } z \neq \overline{1} \\ \overline{1} & \text{if } y \neq 1 \text{ and } z = \overline{1} \\ 0 & \text{otherwise}\ . \end{cases}$$

Fig. 2: Asymmetric gliders.

The symbol 1 can be interpreted as a stationary particle, $\overline{1}$ is a left-going particle and 0 is the neutral background. It is clear from the definition of f that a particle 1 and a $\overline{1}$ annihilate when they meet (Figure 2). In the sequel, the symbols of A are weighted naturally, namely, 0 with 0, 1 with 1 and $\overline{1}$ with -1. Thus, for example, $1 + \overline{1} = 0$ and $\overline{1} + \overline{1} = -2$ and so on.

Again, we consider a Bernoulli measure on $\left\{\overline{1}, 0, 1\right\}^{\mathbb{Z}}$, i.e., a sequence of independent identically distributed random variables $X = (X_i)_{i \in \mathbb{Z}}$ over $\left\{\overline{1}, 0, 1\right\}$ such that $\forall i \in \mathbb{Z}, \mathbb{P}[X_i = \overline{1}] = \mathbb{P}[X_i = 1] = p$, $\mathbb{P}[X_i = 0] = 1 - 2p = q$. Then, for any CA global rule F and any $n \in \mathbb{N}$, $F^n(X)_0$ is also a random variable whose distribution depends on the initial distribution of X.

Proposition 1 *Let F be the global function of Asymmetric Gliders CA. If $\forall i \in \mathbb{Z}$, $\mathbb{P}[X_i = 1] = \mathbb{P}[X_i = \overline{1}] \leq 1/2$ then*

1. $\lim_{n \to \infty} \mathbb{P}[F^n(X)_0 = 0] = 1$;

2. $\lim_{n \to \infty} \mathbb{P}[T_n^0(X) = 0] = 1$;

3. $\lim_{n \to \infty} \mathbb{P}[T_n^1(X) = \infty] = 1$.

Proof: The following relations between $F^n(X)_0$ and the random variables X_i hold

$$F^n(X)_0 = 1 \quad \Leftrightarrow \quad \forall k \leq n, \sum_{i=0}^{k} X_i > 0$$

$$F^n(X)_0 = \overline{1} \quad \Leftrightarrow \quad \forall k \leq n, \sum_{i=k}^{n} X_i < 0$$

$$F^n(X)_0 = 0 \quad \Leftrightarrow \quad \exists k \leq n, \sum_{i=0}^{k} X_i \leq 0 \text{ and } \exists k \leq n, \sum_{i=k}^{n} X_i \geq 0$$

Since $\sum_{i=0}^{n} X_i$ is a recurrent Markov chain, we get

$$\lim_{n \to \infty} \mathbb{P}[F^n(X)_0 = 1] = \lim_{n \to \infty} \mathbb{P}[F^n(X)_0 = \overline{1}] = 0$$

Therefore, it follows that $\lim_{n \to \infty} \mathbb{P}[F^n(X)_0 = 0] = 1$, $\lim_{n \to \infty} \mathbb{P}[T_n^0(X) = 0] = 1$, and also $\lim_{n \to \infty} \mathbb{P}[T_n^1(X) = \infty] = 1$. □

As a consequence of Proposition 1 as $n \to \infty$, $T_n^0(X) \to 0$ and $T_n^1(X) \to \infty$ in probability. Moreover, since for any $n \in \mathbb{N}$ the set of events such that $T_n^1(X) = \infty$ is contained in the one such that $\lim_{n \to \infty} T_n^1(X) = \infty$, it holds that $\mathbb{P}[\lim_{n \to \infty} T_n^1(X) = \infty] \geq \lim_{n \to \infty} \mathbb{P}[T_n^1(X) = \infty] = 1$, and, hence, $T_n^1(X) \to \infty$ almost surely.

Proposition 2 *Let F be the global function of Asymmetric Gliders CA. If $\forall i \in \mathbb{Z}$, $\mathbb{P}[X_i = 1] = \mathbb{P}[X_i = \overline{1}] = 1/2$ then*

1. $\forall x, \lim_{n \to \infty} \mathbb{P}[T_n^{\overline{1}}(X) > x] = 1$;

2. $\forall n \in \mathbb{N}, \mathbb{E}(T_n^{\overline{1}}) = \infty.$

Theorem 3 *Let F be the global function of Asymmetric Gliders CA. If $\forall i \in \mathbb{Z}$, $\mathbb{P}[X_i = 1] = \mathbb{P}[X_i = \overline{1}] = 1/2$ then*

$$\lim_{n \to \infty} \mathbb{P}\left[\frac{T_n^{\overline{1}}(X)}{n} \leq x\right] = \frac{2}{\pi} \arctan \sqrt{x} \ .$$

In the general case with $p \leq 1/2$ we have $\mathbf{Var}(X) = 2p$ so the time scales by $\sqrt{2p}$. Hence, we can give the following.

Conjecture 1 *Let F be the global function of Asymmetric Gliders CA. If $\forall i \in \mathbb{Z}$, $\mathbb{P}[X_i = 1] = \mathbb{P}[X_i = \overline{1}] \leq 1/2$ then*

$$\lim_{n \to \infty} \mathbb{P}\left[\frac{T_n^{\overline{1}}(X)}{n} \leq x\right] = \frac{2}{\pi} \arctan \sqrt{2px} \ .$$

4   Proof of main results

Notation.   For the sake of simplicity, from now on $T_n^{\overline{1}}$ is denoted T_n whenever no misunderstanding is possible.

First of all, we should precise the definition of what we mean by annihilation of particles 1 and $\overline{1}$.

Definition 2 (Annihilation) *A particle 1 at the position n is* annihilated *with the particle $\overline{1}$ at position $n + k$, if $F^{k-1}(X)_n = 1$ and $F^{k-1}(X)_{n+1} = \overline{1}$.*

Denote by Y_n the number of particles 1 in the interval $[0, n)$ which are not annihilated with any particle $\overline{1}$ in the interval $[0, n)$. Then, $Y_0 = 0$ and $Y_{n+1} = \max\{0, Y_n + X_n\}$, so Y is a Markov chain whose transition probabilities are in Figure 3.

For the probabilities $P_{n,m} = \mathbb{P}[Y_n = m]$ we have $P_{0,0} = 1$ and $P_{n,m} = 0$ for $m > n$. The balance equations for the Markov chain Y give

$$
\begin{align}
P_{n+1,0} &= (1-p) \cdot P_{n,0} + p \cdot P_{n,1} \tag{1} \\
P_{n+1,m} &= p \cdot P_{n,m-1} + (1-2p) \cdot P_{n,m} + p \cdot P_{n,m+1} \text{ for } m > 0 \tag{2}
\end{align}
$$
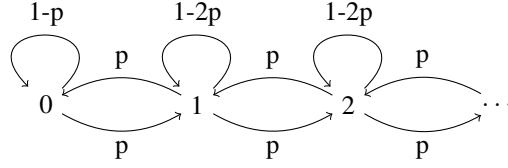
Fig. 3: The Markov chain Y above defined.

For a fixed $n \geq 0$ consider the stochastic process Z such that $Z_0 = Y_n$ and $Z_{m+1} = Z_m + X_{n+m}$. If $Z_0, \ldots Z_m$ are all nonnegative, then Z_m is the number of particles 1 in $[0, n+m)$ which are not annihilated with any particle $\bar{1}$ in $[0, n+m)$. For $m \geq l$ define the entry times for Z as follows

$$S_{m,l} = \min\{t > 0 : \ Z_t = l | Z_0 = m\}$$

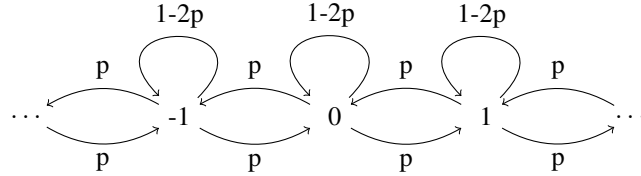and the associated probabilities

$$Q_{m,k} = \mathbb{P}[S_{m+l,l} = k] = \mathbb{P}[S_{m,0} = k] \ .$$

Remark that for $m_2 > m_1 > m_0$ we have $S_{m_2,m_0} = S_{m_2,m_1} + S_{m_1,m_0}$, so $S_{m,0}$ is the sum of m independent random variables which have all the same distribution as $S_{1,0}$. Thus $Q_{0,1} = q$, $Q_{0,2} = 2p^2$, $Q_{1,1} = p$, $Q_{m,1} = 0$ for $m > 1$, $Q_{1,2} = p(1-2p) = pq$, $Q_{2,2} = p^2$ and $Q_{m,k} = 0$ for $m > k$. According to the equilibrium equation of the Markov chain Z one finds

$$\begin{align}
Q_{0,k+1} &= 2p \cdot Q_{1,k} \ \text{for } m = 0 \tag{3} \\
Q_{1,k+1} &= q \cdot Q_{1,k} + p \cdot Q_{2,k} \ \text{for } m = 1 \tag{4} \\
Q_{m,k+1} &= p \cdot Q_{m-1,k} + q \cdot Q_{m,k} + p \cdot Q_{m+1,k} \ \text{for } m > 1 \tag{5}
\end{align}$$

Fig. 4: The Markov chain Z above introduced.

Remark that $T_n = k$ iff $Z_{k+1} = -1$ and $Z_j \geq 0$ for all $j \leq k$. Thus, if $Z_0 = m$, then $T_n = k$ iff $S_{m,-1} = k + 1$. So, for the entry time T_n and the related probabilities $\mathbb{P}[T_n = k | Z_0 = m]$ and $R_{n,k} := \mathbb{P}[T_n = k]$, we have

$$\begin{align}
T_n &= \chi_{[Y_n=m]} \cdot S_{m,-1} \\
\mathbb{P}[T_n = k | Z_0 = m] &= \mathbb{P}[S_{m,-1} = k + 1] = Q_{m+1,k+1} \\
R_{n,k} &= \mathbb{P}[T_n = k] = \sum_{m=0}^{\min(n,k+1)} P_{n,m} \cdot Q_{m+1,k+1}
\end{align}$$

When $\forall i \in \mathbb{Z}$, $\mathbb{P}[X_i = 1] = \mathbb{P}[X_i = \bar{1}] = 1/2$ and $\mathbb{P}[X_i = 0] = 0$ from the definitions of the probabilities P and Q we obtain the following matrices.

$$P = \begin{array}{c|cccccc} & 0 & 1 & 2 & 3 & 4 & \cdots \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & \cdots \\ 1 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & \cdots \\ 2 & \frac{2}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & \cdots \\ 3 & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} & \frac{1}{8} & 0 & \cdots \\ 4 & \frac{6}{16} & \frac{4}{16} & \frac{4}{16} & \frac{1}{16} & \frac{1}{16} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{array}$$

$$Q = \begin{array}{c|cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & \cdots \\ \hline 0 & 0 & \frac{1}{2} & 0 & \frac{1}{8} & 0 & \frac{1}{16} & 0 & \cdots \\ 1 & \frac{1}{2} & 0 & \frac{1}{8} & 0 & \frac{1}{16} & 0 & \frac{5}{128} & \cdots \\ 2 & 0 & \frac{1}{4} & 0 & \frac{1}{8} & 0 & \frac{5}{64} & 0 & \cdots \\ 3 & 0 & 0 & \frac{1}{8} & 0 & \frac{3}{32} & 0 & \frac{9}{128} & \cdots \\ 4 & 0 & 0 & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{array}$$

Next lemmata will give closed formulas for P and Q.

Lemma 1 *If $\forall i \in \mathbb{Z}$, $\mathbb{P}[X_i = 1] = \mathbb{P}[X_i = \bar{1}] = 1/2$ then for all $n, m \in \mathbb{N}$ with $m \leq n$, the following equalities hold on the probabilities $P_{n,m}$:*

$$P_{2n,2m+2} = P_{2n,2m+1} = \binom{2n}{n+m+1} \cdot 2^{-2n} \tag{6}$$

$$P_{2n+1,2m} = P_{2n+1,2m+1} = \binom{2n+1}{n+m+1} \cdot 2^{-2n-1} \tag{7}$$

$$P_{n,m} = \binom{n}{\lceil \frac{n+m}{2} \rceil} \cdot 2^{-n} \tag{8}$$

Proof: We have $P_{0,0} = 1$. By induction, assume that equalities (6) and (7) are true for some value n and for all $m \leq n$. Since from equations (1) and (2), $P_{n+1,0} = (P_{n,0} + P_{n,1})/2$ and $P_{n+1,m} = (P_{n,m-1} + P_{n,m+1})/2$, it follows that (6) and (7) are true for $n+1$ and for all $m \leq n+1$. Thus, (6) and (7) hold, and, as a consequence, equality (8) too. $\qquad\square$

According to Rényi (1970), the following relation holds for the $Q_{m,k}$

$$Q_{1,2k-1} = Q_{0,2k} = \frac{1}{k \cdot 2^{2k-1}} \binom{2k-2}{k-1} = (-1)^{k-1} \cdot \binom{1/2}{k} \tag{9}$$

Lemma 2 *If $\forall i \in \mathbb{Z}$, $\mathbb{P}[X_i = 1] = \mathbb{P}[X_i = \bar{1}] = 1/2$ then for all m with $0 < m \leq k$ the following equalities on the quantities $Q_{m,k}$ hold.*

$$Q_{m,k} = 0, \quad \text{if} \quad \mathbf{mod}_2(k+m) = 1 \tag{10}$$

while

$$Q_{2m,2k} = \frac{m}{k} \binom{2k}{k-m} \cdot 2^{-2k} \tag{11}$$

$$Q_{2m+1,2k+1} = \frac{2m+1}{2k+1} \binom{2k+1}{k-m} \cdot 2^{-2k-1} \tag{12}$$

$$Q_{m,k} = \frac{m}{k} \binom{k}{(k-m)/2} \cdot 2^{-k}, \tag{13}$$

if $\mathbf{mod}_2(k+m) = 0$, where $\mathbf{mod}_2(m)$ is m mod 2.

Proof: Since $q = 0$, if $\mathbf{mod}_2(k + m) = 1$ Equality (10) follows from the definition of $Q_{m,k}$. Using that $Q_{2,k} = 2 \cdot Q_{1,k+1}$, $Q_{m+1,k} = 2 \cdot Q_{m,k+1} - Q_{m-1,k}$ (for $m \geq 1$), and Equation (9), Equalities from (11) to (13), are true for $Q_{1,k}$ and $Q_{2,k}$. The thesis is obtained by proceeding by finite induction on m. $\qquad\square$

Using the expressions found in Lemmata 1 and 2 and substituting them in the definition of $R_{n,k} = \mathbb{P}[T_n = k]$, one can easily find the following.

$$R_{2n,2k} = 2^{-2n-2k-1} \sum_{m=0}^{\min(n,k)} \binom{2n}{n+m} \binom{2k+1}{k-m} \frac{2m+1}{2k+1}$$

$$R_{2n,2k+1} = 2^{-2n-2k-2} \sum_{m=0}^{\min(n,k)} \binom{2n}{n+m+1} \binom{2k+2}{k-m} \frac{m+1}{k+1}$$

$$R_{2n+1,2k} = 2^{-2n-2k-2} \sum_{m=0}^{\min(n,k)} \binom{2n+1}{n+m+1} \binom{2k+1}{k-m} \frac{2m+1}{2k+1}$$

$$R_{2n+1,2k+1} = 2^{-2n-2k-3} \sum_{m=0}^{\min(n,k)} \binom{2n+1}{n+m+1} \binom{2k+2}{k-m} \frac{m+1}{k+1}$$

which can be summed up as follows

$$R_{n,k} = 2^{-n-k-1} \sum_{m=0}^{\min(\lfloor \frac{n}{2} \rfloor, \lfloor \frac{k}{2} \rfloor)} \binom{n}{\lfloor \frac{n}{2} \rfloor + m + \ell_{n,k}} \binom{k+1}{\lfloor \frac{k}{2} \rfloor - m} \frac{2m+1+\mathbf{mod}_2(k)}{k+1}$$

where $\ell_{n,k} = \max\{\mathbf{mod}_2(n), \mathbf{mod}_2(k)\}$.

Finally, we will use the following approximation formula of binomial distribution by the normal distribution.

Theorem 4 (Rényi (1970)) *Let k_n be a sequence of positive integers such that $|2k_n - n| < a\sqrt{n}$ for some constant a. Then*

$$\binom{n}{k_n} = \frac{2^{n+1} \cdot e^{-(2k_n - n)^2/2n}}{\sqrt{2\pi n}} (1 + \mathcal{O}(1/n))$$

and the constant in the remainder $\mathcal{O}(1/n)$ depends only on a.

Proof of Proposition 2:

1. Using the approximation given by Theorem 4 one finds

$$\lim_{n \to \infty} P_{n,m} = \lim_{n \to \infty} \frac{2e^{-m^2/2n}}{\sqrt{2\pi n}} = 0$$

and hence

$$\lim_{n \to \infty} \mathbb{P}[T_n \leq \ell] = \lim_{n \to \infty} \sum_{k=0}^{\ell} \sum_{m=0}^{\min(n,k)} P_{n,m} \cdot Q_{m+1,k+1} = 0$$

2. To prove $\mathbb{E}(T_n) = \infty$ we prove first that $\mathbb{E}(S_{m,1}) = \infty$ for each m. Again, using Theorem 4, one finds

$$Q_{m,k} = \frac{m}{k} \cdot \binom{k}{\lceil \frac{k-m}{2} \rceil} \cdot 2^{-k} \approx \frac{m}{k} \cdot 2^{k+1} \frac{e^{-\frac{m^2}{2k}}}{\sqrt{2\pi k}} \cdot 2^{-k} = \frac{2m \cdot e^{-\frac{m^2}{2k}}}{k\sqrt{2\pi k}} \tag{14}$$

whenever $m < \sqrt{k}$ and $\mathbf{mod}_2(k+m) = 0$. For each ℓ we then have

$$\mathbb{E}(S_{m,0}) = \sum_{k=1}^{\infty} k \cdot Q_{m,k} \geq \sum_{k=\ell}^{\infty} \frac{2m \cdot e^{-\frac{m^2}{2k}}}{\sqrt{2\pi k}} = \infty \ . \tag{15}$$

Since $\mathbb{E}(T_n)$ is a finite linear combination of $\mathbb{E}(S_{0,-1}), \ldots, \mathbb{E}(S_{n,-1})$, we get $\mathbb{E}(T_n) = \infty$ as well.

$\square$

Proof of Theorem 3: Recall that the characteristic function of a discrete distribution with $\mathbb{P}[X = n] = p_n$ is $\varphi(t) = \sum_{n=0}^{\infty} p_n e^{int}$, where i is the imaginary unit. Since by Lemma 2 $Q_{m,k} = 0$ if $\mathbf{mod}_2(k+m) = 1$, for the characteristic function φ of $S_{1,0}$ we obtain

$$\varphi(t) \quad = \quad Q_{1,1}e^{it} + Q_{1,3}e^{3it} + Q_{1,5}e^{5it} + \cdots$$

which, by using Equation (9), turns into

$$\varphi(t) \quad = \quad \binom{1/2}{1} e^{it} - \binom{1/2}{2} e^{3it} + \binom{1/2}{3} e^{5it} - \cdots$$

$$= \quad e^{-it}\left[1 - \left(1 - \binom{1/2}{1} e^{2it} + \binom{1/2}{2} e^{4it} - \cdots \right)\right]$$

$$= \quad e^{-it}(1 - \sqrt{1 - e^{2it}}) \ .$$

Thus, finally we get

$$\lim_{n\to\infty} \varphi\left(\frac{t}{n^2}\right)^n \quad = \quad \lim_{n\to\infty} (1 - \sqrt{1 - e^{2it/n^2}})^n = \lim_{n\to\infty} \left(1 - \frac{\sqrt{-2it}}{n}\right)^n = e^{-\sqrt{-2it}}$$

which is the characteristic function of a random variable with distribution function

$$G(x) = 2(1 - \Phi(1/\sqrt{x}))$$

and density

$$g(x) = G'(x) = \frac{1}{\sqrt{2\pi x^3}} \cdot e^{-1/2x} \ ,$$

where $\Phi(x)$ is the normal distribution function. Figure 5 plots both $G(x)$ and $g(x)$.

Recall that $\mathbb{E}(S_{m,0}) = \infty$ (see Equation 15). Denote by $G_m(x) = \mathbb{P}[S_{m,0} \leq x]$ the distribution function of $S_{m,0}$. Since the characteristic function of $S_{m,0}$ is $\varphi_m(t) = \varphi(t)^m$, we get

$$\lim_{m\to\infty} G_{m+1}(m^2 x) = \lim_{m\to\infty} \mathbb{P}\left[\frac{S_{m,-1}}{m^2} < x\right] = G(x) = 2(1 - \Phi(1/\sqrt{x})) \tag{16}$$
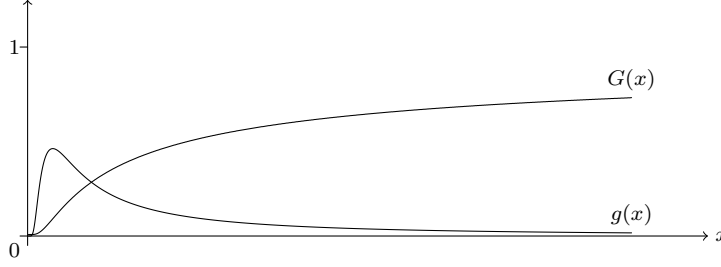
Fig. 5: Asymptotic distribution functions and densities of $\frac{S_{m,0}}{m^2}$.

Denote by $H_n(x) = \mathbb{P}[T_n \leq x]$ the distribution function of T_n. We estimate $H_n(nx) = \mathbb{P}[T_n/n \leq x]$. For a fixed $x > 0$ and $0 < a < b$ we have

$$A_n(a) \quad := \quad \sum_{m=0}^{\lfloor a\sqrt{n}\rfloor} P_{n,m} \cdot G_{m+1}(nx) \leq a\sqrt{n} \cdot P_{n,0} \leq \frac{2a}{\sqrt{2\pi}}(1 + \mathcal{O}(1/\sqrt{n}))$$

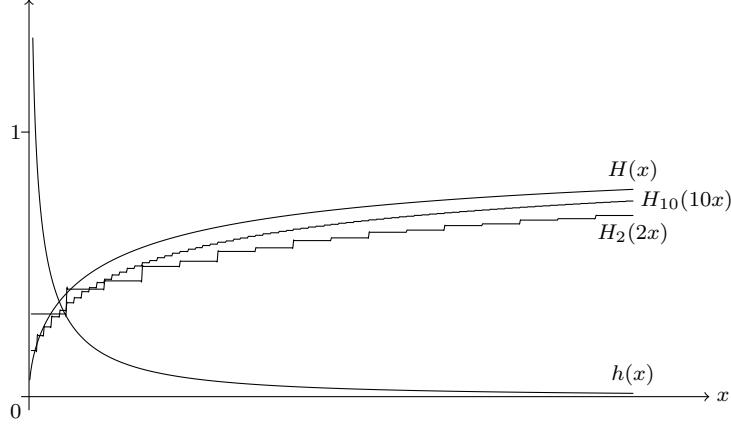$$B_n(b) \quad := \quad \sum_{m=\lceil b\sqrt{n}\,\rceil}^{n} P_{n,m} \cdot G_{m+1}(nx) \leq G_{\lceil b\sqrt{n}\rceil}(nx)$$

so

$$\lim_{n\to\infty} A_n(a) \leq \frac{2a}{\sqrt{2\pi}} \quad \text{and} \quad \lim_{n\to\infty} B_n(b) \leq G(x/b^2) \ . \tag{17}$$

We get

$$H_n(nx) \quad = \quad \sum_{m=0}^{n} P_{n,m} \cdot G_{m+1}(nx)$$

$$= \quad A_n(a) + B_n(b) + \sum_{m=\lceil a\sqrt{n}\rceil}^{\lfloor b\sqrt{n}\rfloor} P_{n,m} \cdot G_{m+1}(nx)$$

$$= \quad A_n(a) + B_n(b) + \sum_{m=\lceil a\sqrt{n}\rceil}^{\lfloor b\sqrt{n}\rfloor} \frac{2e^{-(m+\mathbf{mod}_2(m+n))^2/2n}}{\sqrt{2\pi n}} \cdot (1 + \mathcal{O}(1/n)) \cdot G_{m+1}(nx)$$

Some approximations functions $H_n(nx)$ are plotted in Figure 6. Remark how they quickly converge to $H(x)$. Using variable $y = m/\sqrt{n}$, we get $nx = xm^2/y^2$ and compute $H(x) = \lim_{n\to\infty} H_n(nx)$

$$H(x) \quad = \quad \lim_{n\to\infty} \left(A_n(a) + B_n(b) + \sum_{m=\lceil a\sqrt{n}\rceil}^{\lfloor b\sqrt{n}\rfloor} \frac{2e^{-y^2/2}}{\sqrt{2\pi n}} \cdot G_{m+1}(m^2x/y^2) \right)$$

$$= \quad \lim_{n\to\infty} (A_n(a) + B_n(b)) + \lim_{n\to\infty} \sum_{m=\lceil a\sqrt{n}\rceil}^{\lfloor b\sqrt{n}\rfloor} \frac{2e^{-y^2/2}}{\sqrt{2\pi n}} \cdot G_{m+1}(m^2x/y^2)$$

Fig. 6: Asymptotic distribution functions and densities of $\frac{T_n}{n}$.

and, by using 16, we obtain (recall that $y = m/\sqrt{n} \in [a, b]$)

$$H(x) = \lim_{n \to \infty} (A_n(a) + B_n(b)) + \sqrt{\frac{2}{\pi}} \sum_{y=a}^{b} e^{-y^2/2} \cdot G_{m+1}(m^2 x/y^2)$$

Since $\lim_{a \to 0} \lim_{n \to \infty} A_n(a) = 0$ and $\lim_{b \to \infty} \lim_{n \to \infty} B_n(b) = 0$ (see Inequalities in 17), we have

$$H(x) = \sqrt{\frac{2}{\pi}} \cdot \lim_{a \to 0} \lim_{b \to \infty} \sum_{y=a}^{b} e^{-y^2/2} \cdot G_{m+1}(m^2 x/y^2)$$

$$= \sqrt{\frac{2}{\pi}} \int_0^\infty e^{-y^2/2} \cdot G(x/y^2) \, dy$$

which gives for the density

$$h(x) = H'(x) = \sqrt{\frac{2}{\pi}} \cdot \int_0^\infty e^{-y^2/2} \cdot g(x/y^2) \cdot y^{-2} \, dy$$

$$= \frac{1}{\pi \sqrt{x^3}} \int_0^\infty y \cdot e^{-\frac{y^2}{2}(1+\frac{1}{x})} \, dy = \frac{1}{\pi \sqrt{x}(x+1)}$$

and hence

$$H(x) = \int h(x) dx = \frac{2}{\pi} \arctan \sqrt{x}$$

$\square$

5   Conclusions

In this paper we set up some formal tools that have help to study and to exactly derive the distribution of entry time for CA viewed a particle system consisting of a stationary particle and a left-going particle.

The program is to develop formal tools in order to be able to study more complex situations starting by the symmetric case for example. Another interesting case would consider particles with speed different from 1 or 0 allowing in this way more complex interactions between particles other than annihilation.

References

R. Fisch. The one-dimensional cyclic cellular automaton: A system with deterministic dynamics which emulates an interacting particle system with stochastic dynamics. *Journal of Theoretical Probability*, 3:311–338, 1990.

E. Formenti and P. Kůrka. Subshifts attractors in cellular automata. *Nonlinearity*, 20:105–117, 2007.

R. H. Gilman. Classes of cellular automata. *Ergodic Theory and Dynamical Systems*, 7:105–118, 1987.

P. Kůrka and A. Maass. Stability of subshifts in cellular automata. *Fundamenta Informaticae*, 52(1-3): 143–155, 2002.

A. Rényi. *Probability Theory*. Elsevier, 1970.

Solving Two-Dimensional Binary Classification Problem with Use of Cellular Automata

Anna Piwonska[1][†] and Franciszek Seredynski[2][‡]

[1] *Bialystok University of Technology, Computer Science Faculty, Poland*
[2] *Institute of Computer Science, Polish Academy of Sciences, Poland
and Polish-Japanese Institute of Information Technology, Poland*

This paper proposes a cellular automata-based solution of a two-dimensional binary classification problem. The proposed method is based on a two-dimensional, three-state cellular automaton (CA) with the von Neumann neighborhood. Since the number of possible CA rules (potential CA-based classifiers) is huge, searching efficient rules is conducted with use of a genetic algorithm (GA). Experiments show an excellent performance of discovered rules in solving the classification problem. The best found rules perform better than the heuristic CA rule designed by a human and also better than one of the most widely used statistical method: the k-nearest neighbors algorithm (k-NN). Experiments show that CAs rules can be successfully reused in the process of searching new rules.

Keywords: cellular automata, two-dimensional binary classification problem, genetic algorithm

1 Introduction

CA is a discrete, dynamical system composed of many identical cells arranged in a regular grid, in one or more dimensions Wolfram (2002). A two-dimensional CA considered in the paper consists of rectangular grid of cells. Each cell can take one of a finite number of states and has an identical arrangement of local connections with other cells called a neighborhood, which also includes the cell itself. After determining initial states of all cells (an initial configuration of a CA), states of cells are updated synchronously at discrete time steps, according to a local rule defined on a neighborhood. When a grid is finite, one must assume boundary conditions. The most popular of them are periodic boundary conditions and null boundary conditions. There are many possible variations on this basic CAs concept including other types of rules (e.g. totalistic, probabilistic), other than a rectangular grid (e.g. hexagonal), other neighborhood types (e.g. neighborhood changing in time) and many others.

Despite the fact that CAs have the potential to efficiently perform complex computations, the main problem is a difficulty of designing CAs which would behave in the desired way. One must not only

---

[†]Email: `a.piwonska@pb.edu.pl`
[‡]Email: `sered@ipipan.waw.pl`

select a neighborhood type and size, but most importantly the appropriate rule (or rules). Since the number of possible rules is usually huge, this is the extremely hard task. In some applications of CAs one can design an appropriate rule by hand (e.g. the GKL rule designed in 1978 by Gacs, Kurdyumov and Levin for density classification task Gacs et al. (1978)) or can use partial differential equations describing a given phenomenon Omohundro (1984). However, it is not always possible. In the 90-ties of the last century Mitchell and collaborators proposed to use GAs to find CAs rules able to perform one-dimensional density classification task Mitchell et al. (1993) and the synchronization task Das et al. (1995). The results obtained by Mitchell et al. showed that the GA was able to discover CAs rules demonstrating emergent computational strategies.

The literature shows many examples concerning the concept of generating CAs rules using artificial evolution. For example, Sipper presented results of evolving CAs rules to perform thinning and gap filling in isothetic rectangles Sipper (1997). Breukelaar and Back applied GAs to solve the density classi-fication problem as well as AND and XOR problem in two-dimensional CAs Breukelaar and Back (2004). Swiecicka et al. used GAs to find CAs rules able to solve multiprocessor scheduling problem Swiecicka et al. (2006). Oliveira Jr. and de Oliveira used GA to evolve two-dimensional CAs rules for recognition of handwritten digits Oliveira Jr. and de Oliveira (2008). Piwonska and Seredynski used a GA to find appropriate CAs rules for pattern reconstruction task Piwonska and Seredynski (2010). It is worth point-ing out that other evolutionary techniques were also proposed to find efficient CAs rules, such as genetic programming Andre et al. (1996) and gene expression programming Ferreira (2006).

In a classification problem we wish to determine to which class new observations belong, based on the training set of data containing observations whose class is known. The binary classification deals with only two classes, whereas in a multiclass classification observations belong to one of several classes. The well-known classifiers are neural networks, support vector machines, k-NN algorithm, decision trees and others. The idea of using CAs in the classification problem was described by Maji et al. Maji et al. (2004), Povalej et al. Povalej et al. (2004) and recently by Fawcett Fawcett (2008). Fawcett designed the heuristic rule based on the von Neumann neighborhood (so-called voting rule) and tested its performance on different data sets. This paper proposes a different approach: finding appropriate CAs rules by a GA. The effectiveness of rules discovered by a GA will be compared with the effectiveness of the hand-designed voting rule. Both CA-based approaches will be compared with the k-NN algorithm.

This paper is organized as follows. Section 2 describes two-dimensional CAs used in our approach. Section 3 defines the two-dimensional binary classification problem and describes the proposed CA-based algorithm. Experimental results are presented in Section 4. The last section contains conclusions and future work plans.

2   Two-Dimensional Cellular Automata

A two-dimensional CA consists of a rectangular grid of $N \times M$ cells, each of which can take on k possible states Packard and Wolfram (1985). After determining initial states of all cells (i.e. the initial configuration of a CA), each cell changes its state according to a rule ϕ which depends on states of cells in a neighborhood around it. This is usually done synchronously, although asynchronous mode is used too. Two types of neighborhood are commonly used: the von Neumann neighborhood and the Moore one. The first comprises the four cells orthogonally surrounding the central cell while the other consists of the eight cells around the central cell (Fig. 1).

The evolution of a CA with the von Neumann neighborhood can be described by Eq. 1:

(a)   (b)
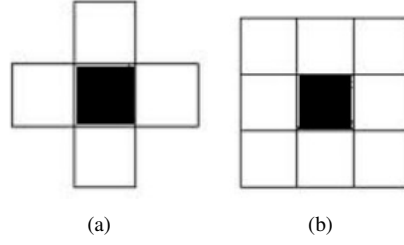
Fig. 1: CA neighborhood: von Neumann (a), Moore (b).

$$a_{i,j}^{(t+1)} = \phi[a_{i,j}^{(t)}, a_{i,j+1}^{(t)}, a_{i+1,j}^{(t)}, a_{i,j-1}^{(t)}, a_{i-1,j}^{(t)}], \tag{1}$$

where $a_{i,j}^{(t)}$ denotes the state of a cell at position i, j in the two-dimensional cellular grid, at time step t.

The evolution of a CA is usually presented by means of so-called "space-time diagrams" displaying grid of cells at subsequent time steps, with each state marked with different color.

3   Binary Classification Problem and Cellular Automata

## 3.1   *Two-Dimensional Binary Classification Problem*

In this paper we deal with the classification problem described in Ishibuchi et al. (1993) in the context of fuzzy rule-based classification system. Let us assume that the data space is the unit square $[0, 1] \times [0, 1]$. Suppose that m data-points $\mathbf{x}_p = (x_{p1}, x_{p2})$, $p = 1, 2, ..., m$ are given as a training set from two classes: class 1 and class 2. That is, the classification of each $\mathbf{x}_p = (x_{p1}, x_{p2})$, $p = 1, 2, ..., m$ is known as one of two classes. The classification problem can be stated as follows. Given m training data find a rule (or "classifier") which divides the data space into two disjoint decision areas (class 1 or 2) such that the class number can be assigned to any new observation.

3.2   *Proposed CA-based Classifier*

The idea of using CAs to solve the binary classification problem is based on the construction of a CA and finding an appropriate rule which can perform the classification task. Since the problem is defined in the two-dimensional space, our CA will also be the two-dimensional. The CA works on a grid of cells, so we must partition our data space $[0, 1] \times [0, 1]$ into a grid. Let us assume that the considered CA has an equal number of cells in each dimension ($N = M$). This means that the data space is divided into $N \times N$ cells and grid lines are placed at nodes: $0, \frac{1}{N}, \frac{2}{N}, \frac{3}{N}, ..., 1$. These nodes determine the division of the interval $[0, 1]$ into N subintervals: $\langle 0, \frac{1}{N} \rangle, \langle \frac{1}{N}, \frac{2}{N} \rangle, ..., \langle \frac{N-1}{N}, N \rangle$.

Our CA is a three-state automaton ($k = 3$). The initial state of each cell is set by training points belonging to this cell and is determined in the following way:

- if there is no point in a cell (an empty cell), then a cell is in state 0 (a cell is marked in grey color),

- if there are points only from class 1 in a cell, then a cell is in state 1 (a cell is marked in white color),
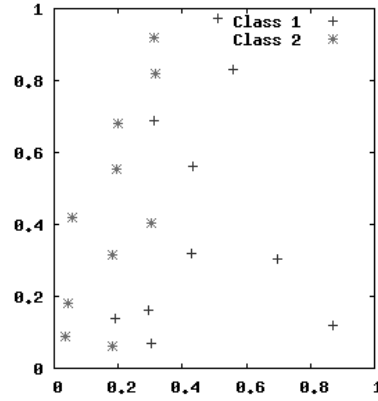
Fig. 2: The exemplary instance of the classification problem.

- if there are points only from class 2 in a cell, then a cell is in state 2 (a cell is marked in black color),

- if there are points from both classes (class 1 and 2) in a cell, then a cell is in state 0 (a cell is marked in grey color).

The interpretation of the above mentioned rules is simply and intuitive. The class of training points determines the state of a cell. If the state of a cell cannot be assigned, a cell is in state 0 (unknown class). This can happen in two situations: either there are no training points in a cell or there are points from both classes in it. Fig. 3 presents the exemplary classification problem: the grid partition in the case of $N = 10$ (Fig. 3(a)) and the corresponding initial configuration of the CA (Fig. 3(b)).

The performance of the CA-based classifier depends on the size of a partition. If a partition is too coarse, the performance of the system may be low (many observations may be misclassified). On the other hand, if a partition is too fine, one can observe the lack of training points in corresponding cells. The similar issue was described in Ishibuchi et al. (1993) in the context of generating fuzzy rules. This problem is illustrated in Fig. 4.

Fig. 4 presents three partitions of considered data space, with the same training set consisting of $m = 100$ points. Cells in state 2 are displayed as black, cells in state 1 are displayed as white and cells in state 0 are displayed as grey. One can see that in spite of the same training set, the CA with $N = 30$ (Fig. 4(c)) has more cells in state 0 than CAs with $N = 10$ (Fig. 4(a)) and $N = 20$ (Fig. 4(b)). This is due to the fact that the CA with $N = 30$ has a lot more cells without training points than CAs with $N = 10$ and $N = 20$. The more empty cells in the initial configuration, the more cells' states need to be properly arranged.

The next step is to determine the boundary conditions. We assume null boundary conditions: border cells have dummy neighbors always in state 0.

After determining initial states of all cells (i.e. the initial configuration of the CA), cells change their states synchronously according to a certain rule which must be found. An appropriate rule transforms, during T time steps, the initial configuration of the CA into the final configuration in which there are no empty cells and for which the correct class number can be assigned to any new observation. Finding an appropriate rule is a key factor for performance of CA-based classifier.

(a) (b)
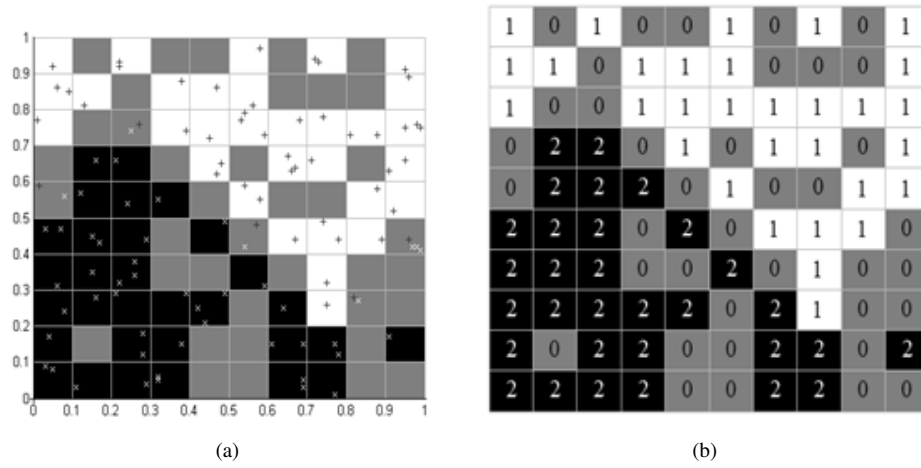
Fig. 3: The exemplary classification problem: the instance of the problem mapped into the CA (a), corresponding initial configuration of the CA (b).

(a) (b) (c)
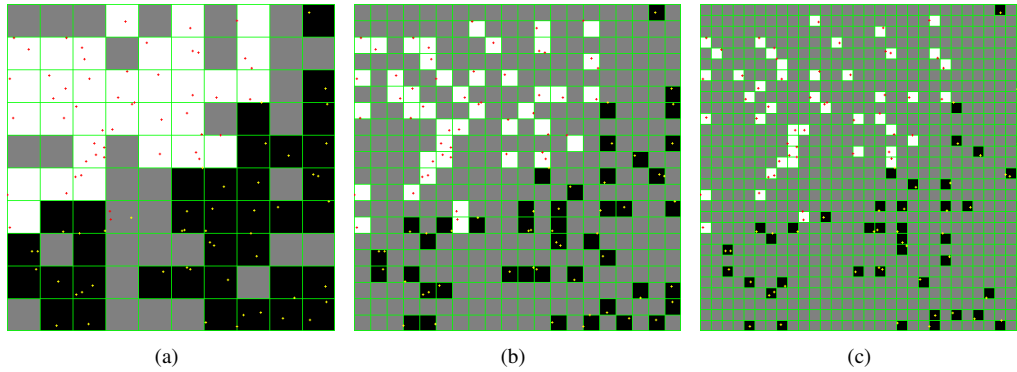
Fig. 4: Three exemplary partitions for: $N = 10$ (a), $N = 20$ (b), $N = 30$ (c).

Let us first consider the heuristic rule for the classification problem designed by Fawcett Fawcett (2008). The rule, called n4V1nonstable, is a non-stable update rule defined on the von Neumann neighborhood with $k = 3$, in which a cell may change its state if the majority changes. According to this rule, the state of a cell at the next time step is determined in the following way:

- if $neigh1 + neigh2 = 0$, then a cell state will be 0,

- if $neigh1 > neigh2$, then a cell state will be 1,

- if $neigh1 < neigh2$, then a cell state will be 2,

- if $neigh1 = neigh2$, then a cell state will be rand{1,2},

where $neigh1$ and $neigh2$ denote the number of a cell's neighbors, respectively in state 1 and 2, and rand{1,2} selects randomly 1 or 2 with equal probability. After determining initial states of all cells, the CA runs for a maximum number of T time steps (if two subsequent CA's configurations are identical, the run is stopped). The intention is that cells will organize themselves into regions of similar class assignment (class 1 or 2). Fig. 5 presents the run of the n4V1nonstable rule on instance 6 of Problem 1 (see Tab. 5). One can see that after six time steps the CA converged to the final configuration without cells in state 0.
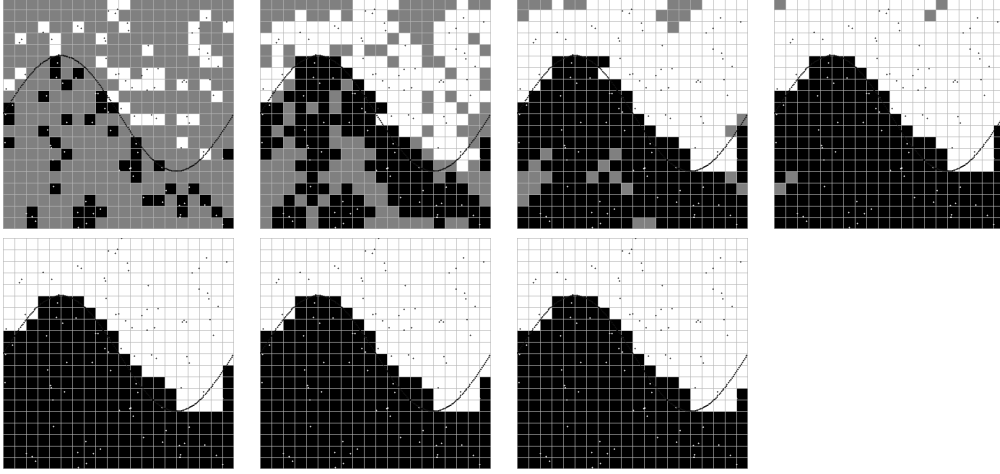
Fig. 5: Problem 1, instance 6, the n4V1nonstable rule: configurations of the CA at time steps: 0 (the initial configuration), 1, 2, 3, 4, 5, 6 (the final configuration).

In our approach there is the same goal but we want to discover such rules by the GA and compare them with the hand-designed n4V1nonstable rule and with the k-NN algorithm. The quality of a given CA rule (n4V1nonstable rule and rules discovered by the GA) is determined on the base of a final configuration of the CA. We generate l new observations of the classification problem and test if new points fall into cells with right states. If a cell is in state 1 then "the answer" of the CA is: "the class of all points falling into this cell is 1" (and similarly with state 2). In rare cases, when a final configuration contains cells in state 0, new points falling into these cells cannot be classified. The score of a CA rule is the sum of the correctly classified points.

3.3   A Genetic Algorithm for Searching Efficient Rules

We assume that the von Neumann neighborhood will be used. Five cells of the von Neumann neighborhood are usually described by directions on the compass: North (N), West (W), Central (C), East (E), South (S). Fig. 6 (left) presents the example of such a neighborhood: 02201. It also lists possible neighborhood states and presents the example of CA rule (on the right, in the rectangle). The value at position 0 in the rule (the value at the top in the rectangle) denotes a state of the central cell of the neighborhood
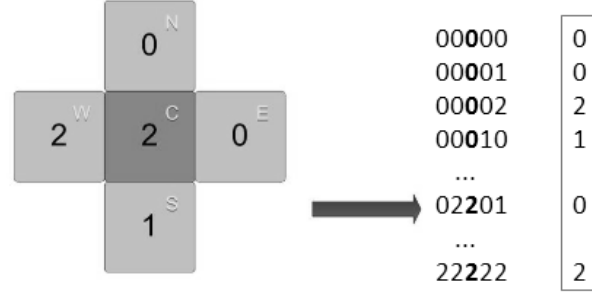
Fig. 6: The neighborhood coding (on the left) and the fragment of the rule - the chromosome of the GA (on the right, in the rectangle).

00000 at the next time step, the value at position 1 in the rule denotes a state of the central cell of the neighborhood 00001 at the next time step and so on, in lexicographic order of neighborhoods.

We can see that with three possible cell states and the neighborhood size equal to 5 we have $3^5 = 243$ possible neighborhood states. Thus, the length of a CA rule is equal to 243 and the number of possible rules is equal to 3^{243}. Since the search space is huge we use a GA to discover an appropriate CA rule. The initial population of P individuals (CA rules) is created randomly.

The important step of the GA is to evaluate rules in the population for the ability to perform the classification task. For this purpose each rule in the population is run on the initial configuration of a CA for T time steps. The initial configuration corresponds to the given problem instance and is determined as described in Sec. 3.2. The final configuration of a CA is used to compute the following fitness function components:

- the number of cells in state 0 ($n0$),

- the number of cells in correct state (1 or 2) (nc),

- the number of cells in incorrect state (1 or 2) (ni),

- the number of cells with a "suspicious neighbor" (nb).

Cells in correct states are these cells in state 1 or 2 whose states in the initial configuration remained unchanged in the final configuration. Cells in incorrect states are these cells in state 1 whose state in the initial configuration was 2 and vice versa: these cells in state 2 whose state in the initial configuration was 1. A cell with a "suspicious neighbor" is a cell which has at least one neighbor in different state than cell's own state. These values are used to compute the fitness f of a rule i, denoted as f_i:

$$f_i = nc - ni - n0 - w \cdot nb \,, \tag{2}$$

where $w \in \langle 0, 1 \rangle$ is a coefficient used to adjust the influence of the number of cells with a "suspicious neighbor" on the fitness. Omitting nb factor causes that the GA tends to evolve CA rules which change states of empty cells into the state 1 or 2 randomly: in the final configuration cells in states 1 and 2 do not form consistent regions, as one would expect.

Once we have the genetic representation and the fitness function defined, we can present the whole GA.

```
Algorithm 1: the GA for searching CA rules
 #01: Begin
 #02:   present an instance of the binary classification problem
 #03:   and create the corresponding CA;
 #04:   generate the initial population of CA rules of size P;
 #05:   for each rule in the population do
 #06:   begin
 #07:     run CA during T time steps;
 #08:     compute the fitness function value;
 #09:   end
 #10:   for i:=1 to G do
 #11:   begin
 #12:     copy E best rules (the elite) from the previous population;
 #13:     randomly choose P-E rules from the elite, with replacement;
 #14:     divide P-E chosen rules into disjoint pairs;
 #15:     cross each pair by means of one point crossover;
 #16:     mutate offsprings with the probability p_m;
 #17:     for each rule in the population do
 #18:     begin
 #20:       run CA during T time steps;
 #21:       compute the fitness function value;
 #22:     end
 #23:   end
 #24:   test the population on a set of l randomly generated
 #25    new points of a given instance;
 #26:   choose the best individual from the population as the result;
 #27: End
```

The GA starts to improve the initial population of rules through repetitive application of selection, crossover and mutation operators. In our experiments we used the selection scheme described by Mitchell Mitchell et al. (1993) in which E best individuals (the elite) are copied without modifications to the next generation (line 12 in Algorithm 1). The remaining $P - E$ rules (line 13) are formed by crossover and mutation from the elite rules. Crossover between two rules involves randomly selecting a single crossover point in the rules and exchanging parts of the rules before and after this point (line 15). Mutation is performed for each individual in the population (with the exception of the elite rules) with the probability p_m (line 16). When a given gene is to be mutated, we replace the current value of this gene by the value 1 or 2, with equal probability. Omitting the value 0 prevents from evolving rules with many 0s. Such rules are more likely to produce configurations containing cells with state 0. It would be unfavorable situation.

These steps are repeated through G generations (line 10). Then, the quality of the final population of rules is tested on $l = 1000$ randomly generated new points of the classification problem (lines 24-25). A new point is classified correctly if it falls into a cell whose state is the same as the class of a point. The quality of a rule is measured by the number of correct classifications. The higher score a rule obtains, the better classifier it represents. The result of the best rule is considered as the result of the proposed method.

4 Experimental Results

As test problems, we took three classification problems. In each problem the data space $[0, 1] \times [0, 1]$ is divided into two classes according to the value of the function $g(\mathbf{x})$, i.e. if $g(\mathbf{x}) \geq 0$ then $\mathbf{x}$ belongs to class 1, else $\mathbf{x}$ belongs to class 2. For each problem we randomly generated 10 problem instances, where each of them had 50 points in class 1 and 50 points in class 2 ($m = 100$). The functions used in experiments are:

- Problem 1: $g(\mathbf{x}) = -sin(2\pi x_1)/4 + x_2 - 0.5$ Ishibuchi et al. (1993)

- Problem 2: $g(\mathbf{x}) = -x_1^3 + x_2 - 0.3$

- Problem 3: $g(\mathbf{x}) = -2x_1 + x_2 + 0.5$

4.1 Experiments on a Grid 10×10

In this series of experiments each problem instance was used to determine the initial configuration of the CA (see Sec. 3.2), with the number of cells in each dimension equal to 10 (grid 10×10). Fig. 7 presents initial configurations of the CA in the case of the exemplary instance of Problem 1, 2 and 3. The dashed line separates points from both classes.

| (a) | (b) | (c) |
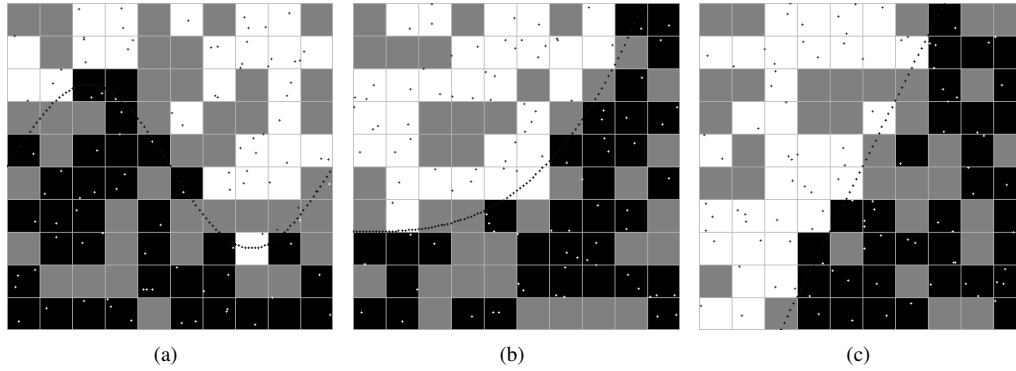
Fig. 7: Initial configurations of the CA: instance 1 of Problem 1 (a), instance 1 of Problem 2 (b) and instance 1 of Problem 3 (c).

The parameters of the CA and the GA were the following: $T = 50$, $P = 200$, $E = 50$, $p_m = 0.05$, $w = 0.1$, $G = 500$. The parameters were tuned during many experiments and these values were chosen to final runs.

An exemplary run of the GA related to the instance 3 of Problem 2 is presented in Fig. 8. One can see that in the early generations the GA quickly improves the best individual (CA rule): its fitness value increases rapidly. This situation is typical for all instances of Problems 1, 2 and 3. Usually, after approximately 100 generations the fitness of the best individual increases narrowly (or in some instances, does not change). However, in the run presented in Fig. 8 there is another rapid increase of the fitness of
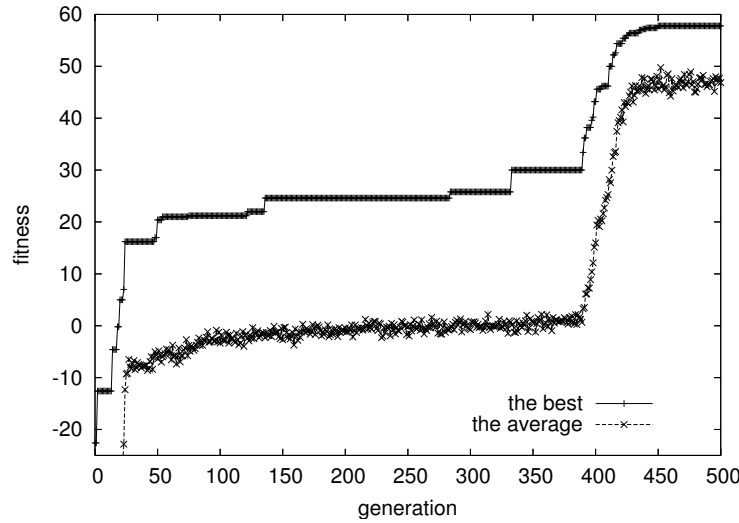
Fig. 8: The run of the GA: instance 3, Problem 2.

the best rule, starting at generation 390. The best rule from the final population obtained the fitness value equal to 57.8.

Since the GA and the n4V1nonstable rule are probabilistic, five runs of each of them were performed. Results obtained by the best found rules of five runs of the GA were compared with the best results obtained by five runs of the n4V1nonstable rule and with the best results obtained by the distance weighted k-NN method Bailey and Jain (1978). The k-NN was allowed to use up to five neighbors and the best k was determined experimentally. Results of these experiments are presented in Tab. 1, 2 and 3. The structure of these tables is as follows. The first column lists the number of a problem instance, the second presents the classification accuracy, measured by the number of correctly classified new points (the maximal value is equal to 1000), obtained by the GA (CA-GA), the third lists the classification accuracy obtained by the k-NN and the last presents the classification accuracy obtained by the n4V1nonstable rule (n4V1).

In the case of Problem 1, the CA-GA approach received better results than the k-NN and n4V1nonstable rule in three instances (instances: 1, 3, 5). In the case of instance 9, the CA-GA obtained the same result as the k-NN. The k-NN received better results than both CA-based methods in six instances (instances: 2, 4, 6, 7, 8, 10). The n4V1nonstable was never the best. Looking at the average results one can see that the best values was obtained by the k-NN. However, the CA-GA algorithm performed significantly better than the n4V1nonstable rule.

In the case of Problem 2, the CA-GA method gained higher score than the k-NN and the n4V1nonstable rule in seven cases (instances: 1, 3, 4, 6, 7, 9, 10). In two cases (instances 2 and 5) the best result was obtained by the k-NN. The n4V1nonstable was the best in only one case: instance 8. The best average result was obtained by the CA-GA method.

In the case of Problem 3, the CA-GA method gained higher score than the k-NN in six instances (instances: 4, 5, 6, 7, 9, 10). In the case of instance 8, the CA-GA obtained the same result as the k-NN.

Tab. 1: Problem 1, grid 10×10: the number of correct classifications, $l = 1000$.

instance	CA-GA	k-NN	n4V1
1	**938**	926	908
2	965	**981**	951
3	**956**	951	914
4	936	**944**	917
5	**966**	962	942
6	943	**952**	950
7	951	**961**	955
8	951	**956**	937
9	**967**	**967**	962
10	957	**970**	944
average	953.00	**957.00**	938.00

Tab. 2: Problem 2, grid 10×10: the number of correct classifications, $l = 1000$.

instance	CA-GA	k-NN	n4V1
1	**964**	959	952
2	954	**973**	966
3	**969**	962	964
4	**967**	962	943
5	970	**972**	957
6	**975**	966	971
7	**961**	945	956
8	941	939	**970**
9	**965**	947	937
10	**972**	961	957
average	**963.80**	958.60	957.30

The k-NN received better results than the CA-GA method in three instances (instances: 1, 2, 3). The n4V1nonstable never gained the best result. The best average result was again obtained by the CA-GA.

On the basis of the classification accuracy, we can conclude that for all methods the hardest cases are instances of Problem 1, then instances of Problem 2 and the easiest classification problem is represented by instances of Problem 3. We can also see that there is no absolutely the best method for all instances of the examined problems.

The run (space-time diagram) of the best rule found by the GA in the case of Problem 1 is presented in Fig. 9. For the comparison, the run of the n4V1nonstable rule is presented in Fig. 10. Final configurations presented in Fig. 9 and Fig. 10 are very similar. However, in the case of the rule discovered by the GA the border between cells in states 1 and 2 resembles more the shape of the dashed line separating points from both classes. This rule gained higher score than n4V1nonstable rule (Tab. 1).

Fig. 11 presents the run of the best rule found by the GA in the case of Problem 2 (instance 6) and Fig. 12 presents the run of the n4V1nonstable rule for the same problem instance. Fig. 13 and Fig. 14 present

Tab. 3: Problem 3, grid 10×10: the number of correct classifications, $l = 1000$.

instance	CA-GA	k-NN	n4V1
1	977	**983**	951
2	970	**971**	968
3	969	**975**	944
4	**960**	948	924
5	**972**	970	941
6	**976**	975	947
7	**973**	956	963
8	**971**	**971**	957
9	**972**	967	961
10	**980**	972	954
average	**972.00**	968.80	951.00
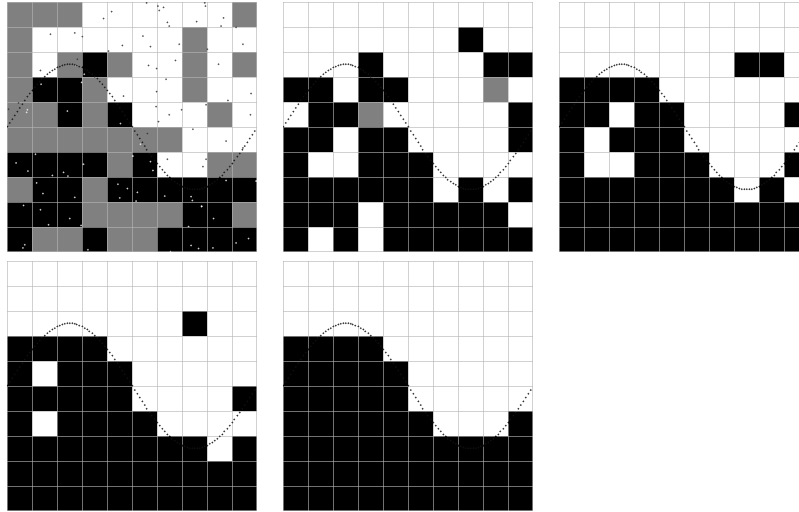
Fig. 9: Problem 1, instance 9, the best rule found by the GA: configurations of the CA at time steps: 0 (the initial configuration), 1, 2, 3, 4 (the final configuration).

the runs of the CA-GA rule and the n4V1nonstable rule in the case of instance 10 of Problem 3.

For all problem instances, final configurations of CAs are very similar in the case of CA rule discovered by the GA and in the case of the n4V1nonstable rule. However, rules discovered by the GA generate final configurations which can more precisely divide points from both classes. They are better tuned to solve the classification problem than the n4V1nonstable rule. The best rules (chromosomes of the GA) found for Problem 1, 2 and 3 are presented in Tab. 4.
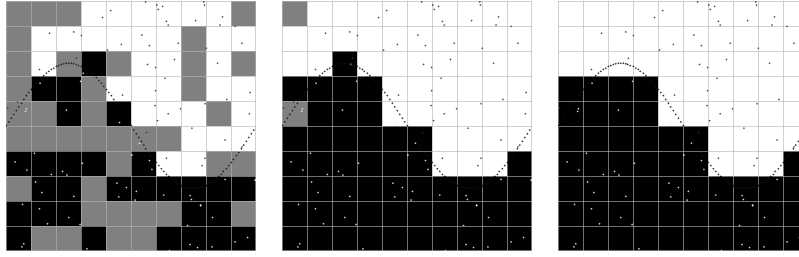
Fig. 10: Problem 1, instance 9, the n4V1nonstable rule: configurations of the CA at time steps: 0 (the initial configuration), 1, 2 (the final configuration).

Fig. 11: Problem 2, instance 6, the best rule found by the GA: configurations of the CA at time steps: 0 (the initial configuration), 1, 2, 3 (the final configuration).
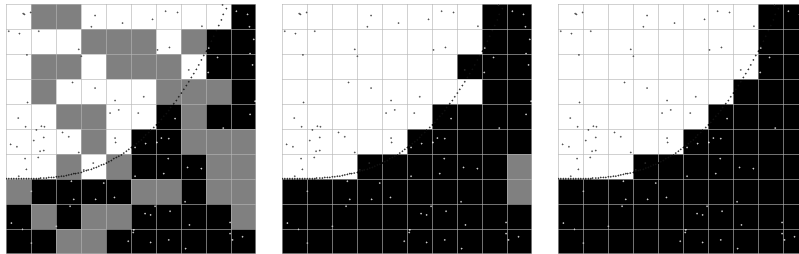
Fig. 12: Problem 2, instance 6, the n4V1nonstable rule: configurations of the CA at time steps: 0 (the initial configuration), 1, 2 (the final configuration).

4.2 Experiments on a Grid 20×20

In order to study the influence of the size of the partition of the data space on the performance of both CA-based methods, a set of new experiments was conducted on the grid 20×20 with the same instances of Problem 1, 2 and 3. It is worth to notice that the k-NN method does not depend on the grid size so results are the same as in the case of the grid 10×10. In the case of the n4V1nonstable rule the methodology of conducted experiments was the same as for the grid 10×10. However, the methodology of using the GA was modified. Instead of generating the initial population of rules totally randomly, we took five the
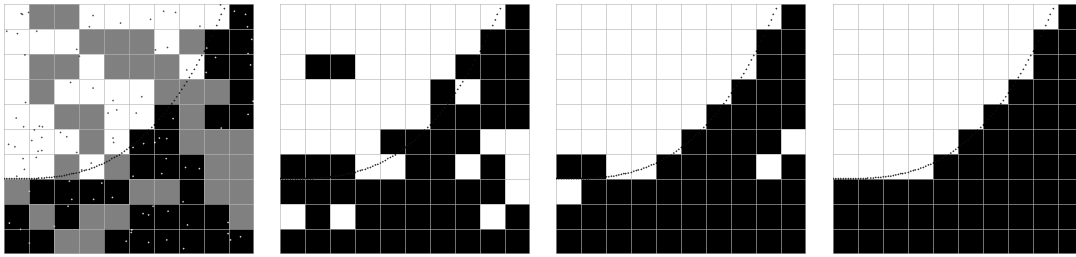
Fig. 13: Problem 3, instance 10, the best rule found by the GA: configurations of the CA at time steps: 0 (the initial configuration), 1, 2, 3, 4, 5, 6, 7 (the final configuration).

Fig. 14: Problem 3, instance 10, the n4V1nonstable rule: configurations of the CA at time steps: 0 (the initial configuration), 1 (the final configuration).

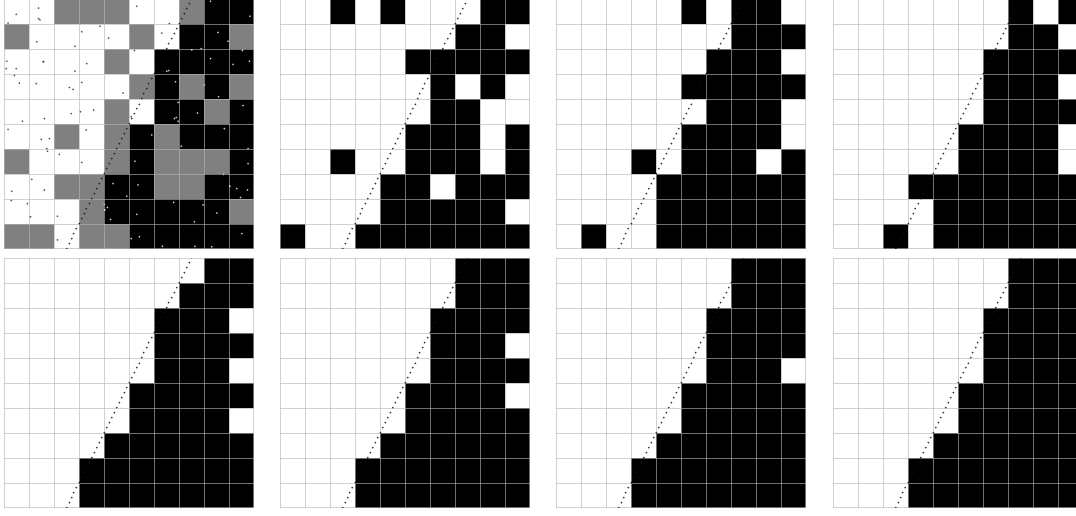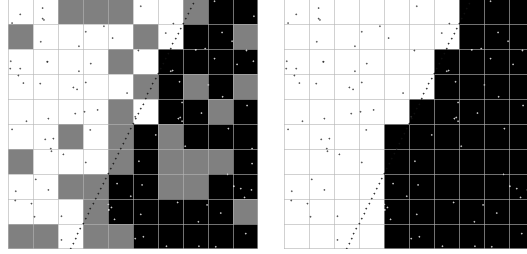best rules from the final population obtained for the grid size 10×10 and then inserted them into the initial population of the GA (with the grid size 20×20). Experiments showed that the GA (on the grid size 20×20) with the initial population containing previously discovered rules evolved better solutions than the GA with randomly generated initial population. Moreover, inserting the best rules into the initial population caused that the efficient rules were discovered very quickly.

As the example, let us look at the instance 5 of Problem 1. Fig. 15 presents two runs of the GA in the case of randomly generated initial population and in the case of inserting previously discovered rules into it. In the first case the GA needs 500 generations to evolve the best rule, which obtained the fitness value equal to 75.30. In the second case, one of the best rules found for the grid size 10×10 receives very high fitness value on the double grid size: 58.20 in the generation 0. The GA improves the best individual and finally finds a rule which obtains the fitness value equal to 83.60.

Tab. 4: Chromosomes of the best rules found by the GA (grid 10×10).

Problem 1 instance 9	1121112121112121221211222122111112211121111122112 22122221221012211111212211211221211211121212101112 2211122222221212111211111211212111221222211222212 11121221211212221122221211122221222102222222112121 11222211222111211222111212211122111222222212222
Problem 2 instance 6	1111122211111111222121222212221221222111211122212222 22121212212211112221121211211212122122212112112211211 22122222122111222211222111111122211111121221222221122 1111112122211212121222222211111122221222112121211112122 211122122121112221222221211111221122222212222222222
Problem 3 instance 10	1111121122110112201121222212222212121112212212121220 11222211121211112222021221212112221111211111211111 1212112112221211111211111111122111211211212121221221 221221212122212222211221221222222212112221122211211 211111111121212122221211111122211221221222222122222
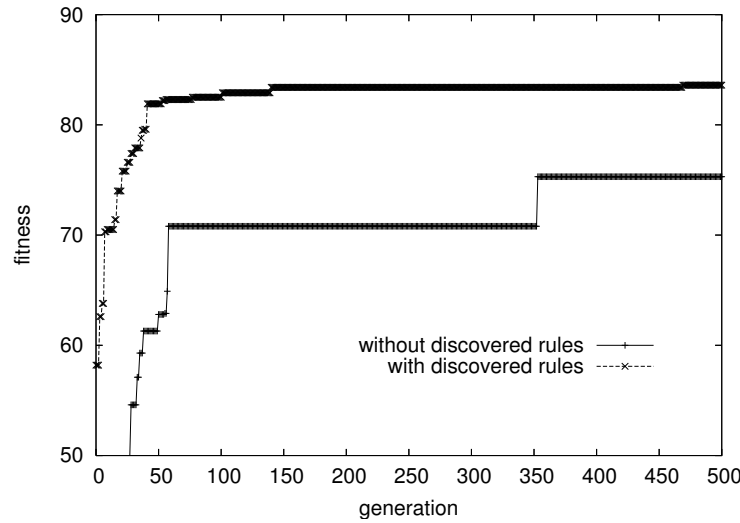
Fig. 15: Problem 1, instance 5: fitness of the best individual.

Fig. 16 and 17 present the final configurations of the best rules in the first and the last generation of the GA in the case of randomly generated initial population (Fig. 16) and in the case of inserting previously discovered rules into the initial population (Fig. 17). One can see the best rule in the random initial population performs chaotically (Fig. 16, left). On the contrary, one of the best rules found for the grid size 10×10 and inserted into random initial population performs quite good on the double grid size (Fig. 17, left). This is the evidence of the scalability of CAs rules (reported in Swiecicka et al. (2006) in the

context of multiprocessor scheduling): rules discovered for a given problem instance can be used to solve the same problems on denser grids. Looking at the configurations from the final generation (Fig. 16 and 17, right) one can see that the rule presented in Fig. 17 can solve the instance of the classification problem more precisely than the rule presented in Fig. 16.
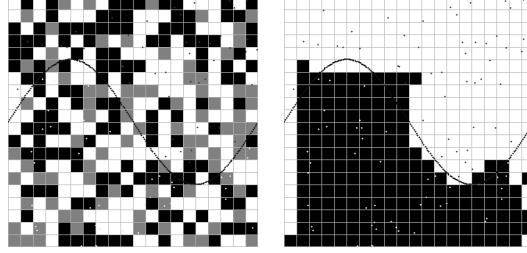
Fig. 16: Problem 1, instance 5, random initial population: the final configurations of the best rule in the generation 0 and 500.
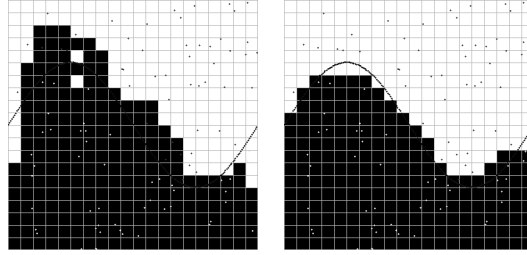
Fig. 17: Problem 1, instance 5, initial population with inserted previously discovered rules: the final configurations of the best rule in the generation 0 and 500.

Results of the experiment in which previously discovered rules are inserted into the initial population of the GA are presented in the second column of Tab. 5, 6, 7. In the case of Problem 1, the CA-GA approach received better results than the k-NN and the n4V1nonstable rule in seven instances. In the case of instance 7, the CA-GA obtained the same result as the k-NN. The k-NN received better results than both CA-based methods in one instance. In one case both the n4V1nonstable and the k-NN received the highest score.

In the case of Problem 2, the CA-GA method gained higher score than the k-NN and n4V1nonstable rule in seven cases. In one case, the CA-GA obtained the same result as the k-NN. The n4V1nonstable rule was the best in two cases.

In the case of Problem 3, the CA-GA method gained the best result in nine instances. In one case the n4V1nonstable was the best. The k-NN method was never the best.

One can see that results obtained for a grid 20×20 are usually better than results obtained for a grid 10×10. This conclusion is true in the case of rules discovered by the GA and in the case of the n4V1nonstable rule. Only in individual cases (e.g. Problem 2, instance 4, CA-GA approach, Problem

Tab. 5: Problem 1, grid 20×20: the number of correct classifications, $l = 1000$.

instance	CA-GA	k-NN	n4V1
1	**964**	926	918
2	974	**981**	977
3	**967**	951	951
4	943	**944**	**944**
5	**975**	962	953
6	**965**	952	964
7	**961**	**961**	947
8	**964**	956	953
9	**971**	967	936
10	**972**	970	953
average	**965.60**	957.00	949.60

Tab. 6: Problem 2, grid 20×20: the number of correct classifications, $l = 1000$.

instance	CA-GA	k-NN	n4V1
1	**975**	959	956
2	976	973	**977**
3	**970**	962	968
4	964	962	**965**
5	**972**	**972**	966
6	**977**	966	959
7	**965**	945	939
8	**978**	939	946
9	**972**	947	941
10	**973**	961	947
average	**972.20**	958.60	956.40

1, instance 7, the n4V1nonstable rule) obtained results were worse. In the case of all problems, the best average result was obtained by the CA-GA approach.

Fig. 18 presents final configurations of the best rules discovered by the GA for Problem 1, 2, and 3. The rules are presented in Tab. 8.

5 Conclusions and Future Work

In this paper we have presented the new approach concerning binary classification in the context of CAs. The main purpose of the paper was to study possibilities of the GA in discovering CA rules which are able to perform binary classification task. Results of presented experiments show that the GA is able to discover rules appropriate to solve this task for a given instance of a problem. The best found rules perform better than the heuristic rule designed by human and better than the k-NN algorithm.

Results of experiments showed very interesting ability of discovered rules. CA rules discovered for grid partition 10×10 have the ability of solving problems for grid partition 20×20. Rules discovered

Tab. 7: Problem 3, grid 20×20: the number of correct classifications, $l = 1000$.

instance	CA-GA	k-NN	n4V1
1	979	983	**985**
2	**984**	971	972
3	**980**	975	978
4	**965**	948	949
5	**974**	970	955
6	**981**	975	964
7	**985**	956	963
8	**979**	971	964
9	**973**	967	964
10	**984**	972	975
average	**978.40**	968.80	966.90

(a)   (b)   (c)
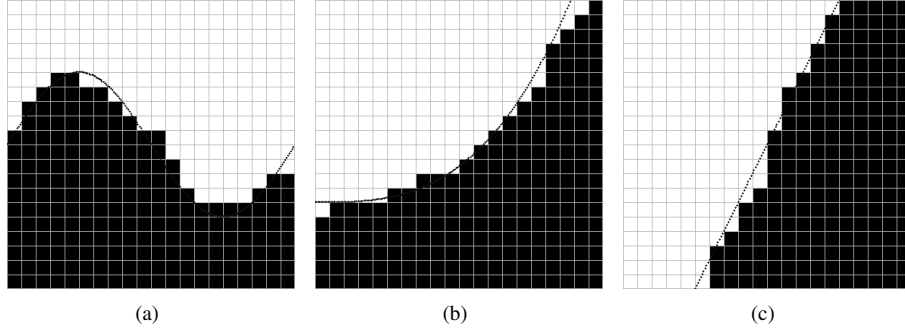
Fig. 18: Final configurations of the best rules discovered by the GA: Problem 1, instance 5 (a), Problem 2, instance 8 (b) and Problem 3, instance 7 (c).

during artificial evolution store some kind of knowledge about instance which is solved. This knowledge can be successfully reused in the process of discovering rules defined on larger grid size. From the point of view of searching the rules on the grid size 20×20, we can interpret the run of the GA on the grid size 10×10 as the preprocessing phase. Then, when more precise results are needed, the best rules can be again used by the GA searching on more fine grid partition.

Experiments performed for both grid sizes show that there is no absolutely the best method for all instances of all problems. The aim of the future research is to discover more universal (no special-purpose) CA rules which could be used to solve the large class of classification problems. The important issue is also time-consuming process of learning CA rules. Instead of population-based method of learning like the GA, we plan to examine other methods requiring only single solution, e.g. recently proposed technique called generalized extremal optimization de Sousa et al. (2004).

Tab. 8: Chromosomes of the best rules found by the GA (grid 20×20).

Problem 1 instance 5	21212122211111111122221122121122111211112111111121 12121211221222221112112222121221112221221111121111 11221122221222121111111121111111122112122121111221112 21111222221222221122221111111111122111221211222211 2112111211212121121222221221121212122122222212212
Problem 2 instance 8	22201101112211101112122102112111112121121112211222 11121220112121210112121212121211222222222222101122 11112112122120211101212112112122112122111122211112 12221221211122221210222221112121221221121120111102 12110111112112112212112211012222222212222011222
Problem 3 instance 7	21211220211211121121202221211211122222222121111122 12212121112211221111212222122222122122222222111111 11211112012122222222211111111122111212221222111212 11122121212211221211221121111212222221211212112111221 20221212121122111212222221212221221122221201222

Acknowledgements

References

D. Andre, F. Bennett III, and J. Koza. Discovery by genetic programming of a cellular automata rule that is better than any known rule for the majority classification problem. In *Proceedings of the First Annual Conference on Genetic Programming GECCO '96*, pages 3–11, 1996.

T. Bailey and A. Jain. A note on distance-weighted k-nearest neighbor rules. *IEEE Transactions on Systems, Man and Cybernetics*, 8(4):311–313, 1978.

M. Banham and A. Katsaggelos. Digital image restoration. *IEEE Signal Processing Magazine*, 14:24–41, 1997.

R. Breukelaar and T. Back. Evolving transition rules for multi dimensional cellular automata. In *Lecture Notes in Computer Science 3305*, pages 182–191. Springer Verlag, 2004.

R. Das, J. Crutchfield, and M. Mitchell. Evolving globally synchronized cellular automata. In *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 336–243, 1995.

F. de Sousa, V. Vlassov, and F. Ramos. Generalized extremal optimization: An application in heat pipe design. *Applied Mathematical Modelling*, 28(10):911–931, 2004.

T. Fawcett. Data mining with cellular automata. *ACM SIGKDD Explorations Newsletter*, 10(1):32–39, 2008.

C. Ferreira. *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence*. Springer, 2006.

P. Gacs, G. Kurdyumov, and L. Levin. One dimensional uniform arrays that wash out finite islands. *Problemy Peredachi Informatsii*, 12:92–98, 1978.

G. Hernandez and H. Herrmann. Cellular automata for elementary image enhancement. *Graphical Models And Image Processing*, 58(1):82–89, 1996.

H. Ishibuchi, K. Nozaki, and N. Yamamoto. Selecting fuzzy rules by genetic algorithm for classification problems. *Fuzzy Systems*, 2:1119–1124, 1993.

P. Maji, B. Sikdar, and P. Chaudhuri. Cellular automata evolution for pattern classification. In *Lecture Notes in Computer Science 3305*, pages 660–669. Springer Verlag, 2004.

M. Mitchell, P. Hraber, and J. Crutchfield. Revisiting the edge of chaos: Evolving cellular automata to perform computations. *Complex Systems*, 7:89–130, 1993.

C. Oliveira Jr. and P. de Oliveira. An approach to searching for two-dimensional cellular automata for recognition of handwritten digits. In *Lecture Notes in Artificial Intelligence 5317*, pages 462–471. Springer Verlag, 2008.

S. Omohundro. Modelling cellular automata with partial differential equations. *Physica 10D*, 10(1-2): 128–134, 1984.

N. Packard and S. Wolfram. Two-dimensional cellular automata. *Journal of Statistical Physics*, 38: 901–946, 1985.

A. Piwonska and F. Seredynski. Learning cellular automata rules for pattern reconstruction task. In *Lecture Notes in Computer Science 6457*, pages 240–249. Springer Verlag, 2010.

P. Povalej, M. Lenic, and P. Kokol. Improving ensembles with classificational cellular automata. In *Lecture Notes in Computer Science 3305*, pages 242–249. Springer Verlag, 2004.

M. Sipper. The evolution of parallel cellular machines toward evolware. *Biosystems*, 42(1):29–43, 1997.

S. Slatnia, M. Batouche, and K. Melkemi. Evolutionary cellular automata based-approach for edge detection. In *Lecture Notes in Computer Science 4578*, pages 404–411. Springer Verlag, 2007.

A. Swiecicka, F. Seredynski, and A. Zomaya. Multiprocessor scheduling and rescheduling with use of cellular automata and artificial immune system support. *IEEE Transactions on Parallel and Distributed Systems*, 17(3):253–262, 2006.

S. Wolfram. *A New Kind of Science*. Wolfram Media, 2002.

The structure of communication problems in cellular automata

Raimundo Briceño[1] and Pierre-Etienne Meunier[2]

[1]*DIM, Universidad de Chile* [2]*LAMA, Université de Savoie et DIM, Universidad de Chile*

Studying cellular automata with methods from communication complexity appears to be a promising approach. In the past, interesting connections between communication complexity and intrinsic universality in cellular automata were shown. One of the last extensions of this theory was its generalization to various "communication problems", or "questions" one might ask about the dynamics of cellular automata. In this article, we aim at structuring these problems, and find what makes them interesting for the study of intrinsic universality and quasi-orders induced by simulation relations.

Keywords: cellular automata, communication complexity, intrinsic universality, ideals

Outline

In Section 1, we recall the basic notions of communication complexity and its application to cellular automata. In Section 2, we show how communication complexity incorporates in the model of cellular automata, generalizing the previous works to other simulation relations, and developing new *communication problems*. Then, in Section 3, we study sets of cellular automata closed under simulation (*ideals*), and how our communication approach relates with them.

1   Introduction and definitions

1.1   Cellular automata and shift spaces

In this paper we are always going to consider one-dimensional cellular automata (CA). A CA is defined by a *local rule* $\phi : Q^{2r+1} \to Q$, where r denotes the *radius* and Q, the *set of states* (or *alphabet*).

We denote by $\Phi : Q^{\mathbb{Z}} \to Q^{\mathbb{Z}}$ the *global function* induced by ϕ following the classical definition:

$$\Phi(x)_i = \phi(x_{i-r}, \dots, x_{i+r}),$$

where x is some element from $Q^{\mathbb{Z}}$ (or $Q_\Phi^{\mathbb{Z}}$, if we want to avoid ambiguities) called *configuration*. Finally, we denote by Φ^t the *t-step iteration* of the global function Φ, such that $\Phi^{t+1} = \Phi^t \circ \Phi$ and $\Phi^1 = \Phi$.

A global function Φ can be represented by different local rules. All properties considered in this paper depend only on Φ and are not sensitive to the choice of a particular local function. However, to avoid useless formalism, we will use the following notion of *canonical local representation*: (ϕ, r) is the canonical

local representation of Φ if ϕ has radius r and it is the local function of smallest radius having Φ as its associated global function. Throughout this work we are going to refer to a CA Φ with (ϕ, r).

The *limit set* of a given CA Φ, denoted $\omega(\Phi)$, is defined as follows:

$$\omega(\Phi) = \bigcap_{t \in \mathbb{N}} \Phi^t(Q^{\mathbb{Z}}).$$

A limit set is always a non-empty *shift space*. A shift space X over an alphabet Q is any subset $X \subseteq Q^{\mathbb{Z}}$ that can be defined by a family of forbidden words $F \subseteq Q^+$, such that X is the set of all configurations where no word of F occurs. A shift space is said to be a *shift of finite type* (SFT) if it can be defined by a finite family F. We denote by $\mathcal{L}(X)$ the set of words occurring in configurations that belongs to X and, by $\mathcal{L}_n(X) := \mathcal{L}(X) \cap Q^n$, its restriction to words of length $n \in \mathbb{N}$. A shift space is said to be a *sofic shift* if $\mathcal{L}(X)$ is a regular language. Clearly, every SFT is a sofic shift.

If there exists a time $t^* \in \mathbb{N}$ such that $\omega(\Phi) = \Phi^{t^*}(Q^{\mathbb{Z}})$, Φ is said to be *stable*, and *unstable*, otherwise. A stable limit set is always a sofic shift.

Finally, we denote $\mathcal{AC}$ the set of one-dimensional CAs.

1.2 Simulations and universality

We define two parallel notions of simulation between CAs developed in [DMOT11], based on geometrical transformations of diagram spaces and injections or projections between them.

Definition 1 (Rescaling) *The ingredients of a rescaling are simple: packing cells into blocks, iterating the rule and composing with a traslation. Formally, given any state set Q and any $m \geq 1$, we define the bijective packing map $b_m : Q^{\mathbb{Z}} \to \left(Q^m \right)^{\mathbb{Z}}$ by:*

$$\forall i \in \mathbb{Z} : \big(b_m(x) \big)(i) = \big(x(mi), \dots, x(mi + m - 1) \big),$$

for all $x \in Q^{\mathbb{Z}}$. The rescaling $\Phi^{\langle m, t, z \rangle}$ of Φ by parameters m (packing), $t \geq 1$ (iterating) and $z \in \mathbb{Z}$ (shifting, denoted σ) is the CA of state set Q^m and global rule:

$$b_m \circ \sigma^z \circ \Phi^t \circ b_m^{-1}.$$

The fact that the above function is the global rule of a cellular automaton follows from Curtis-Lyndon-Hedlund theorem [Hed69] because it is continuous and commutes with traslations.

In the rest of this section, we define various relations between cellular automata. They are all defined in [DMOT11], and we just recall them here.

Definition 2 (Sub-automaton) *A CA Φ_1 is a* sub-automaton *of a CA Φ_2, denoted by $\Phi_1 \sqsubseteq \Phi_2$, if there is an injective map ι from Q_1 to Q_2 such that $\bar{\iota} \circ \Phi_1 = \Phi_2 \circ \bar{\iota}$, where $\bar{\iota} : Q_1^{\mathbb{Z}} \to Q_2^{\mathbb{Z}}$ denotes the uniform extension of ι.*

Definition 3 (Quotient) *A CA Φ_1 is a* quotient *of a CA Φ_2, denoted by $\Phi_1 \trianglelefteq \Phi_2$, if there is a surjective map φ from Q_2 to Q_1 such that $\overline{\varphi} \circ \Phi_2 = \Phi_1 \circ \overline{\varphi}$, where $\overline{\varphi} : Q_2^{\mathbb{Z}} \to Q_1^{\mathbb{Z}}$ denotes the uniform extension of φ.*

Definition 4 (Injective simulation) *We say that Φ_2 injectively simulates Φ_1, denoted $\Phi_1 \preccurlyeq_i \Phi_2$, if there exist rescaling parameters m_1, m_2, t_1, t_2, z_1 and z_2 such that $\Phi_1^{\langle m_1, t_1, z_1 \rangle} \sqsubseteq \Phi_2^{\langle m_2, t_2, z_2 \rangle}$.*

Definition 5 (Surjective simulation) *We say that Φ_2 surjectively simulates Φ_1, denoted $\Phi_1 \preccurlyeq_s \Phi_2$, if there exist rescaling parameters m_1, m_2, t_1, t_2, z_1 and z_2 such that $\Phi_1^{\langle m_1, t_1, z_1 \rangle} \trianglelefteq \Phi_2^{\langle m_2, t_2, z_2 \rangle}$.*

Definition 6 (Intrinsic universality) *Let $\preccurlyeq \in \{\preccurlyeq_i, \preccurlyeq_s\}$. Ψ is* intrinsically $\preccurlyeq$-universal *if for all Φ it holds that $\Phi \preccurlyeq \Psi$.*

It is well known that there exist intrinsically universal cellular automata for the $\preccurlyeq_i$ relation, and this property has been shown undecidable (see for instance [Oll03] and [DMOT11]). An open problem, appearing in various contexts (see [The05] or [BT10]), is the existence of a cellular automaton universal for the $\preccurlyeq_s$ relation:

Open Problem 1 *Is there some Ψ such that for all Φ it holds that $\Phi \preccurlyeq_s \Psi$?*

1.3 Ideals

Informally speaking, ideals are strict subsets of $\mathcal{AC}$ closed under simulation. In the general order theory, the precise definition is the following.

Definition 7 (Ideal) *Let $\preccurlyeq$ be a quasiorder in $\mathcal{AC}$. An ideal $\mathcal{I}$ is a subset of $\mathcal{AC}$ such that:*

1. If $\Phi_2 \in \mathcal{I}$ and $\Phi_1 \preccurlyeq \Phi_2$, then $\Phi_1 \in \mathcal{I}$.

2. For any $\Phi_1, \Phi_2 \in \mathcal{I}$ there is some $\Phi_3 \in \mathcal{I}$ such that $\Phi_1 \preccurlyeq \Phi_3$ and $\Phi_2 \preccurlyeq \Phi_3$.

Moreover, $\mathcal{I}$ is said principal *if there is some Φ_I such that:*

$$\Phi \in \mathcal{I} \iff \Phi \preccurlyeq \Phi_I.$$

Adapted to our context, we have the following sufficient conditions to be an ideal.

Proposition 1 ([DMOT11]) *$\mathcal{I} \subseteq \mathcal{AC}$ is an ideal for $\preccurlyeq_i$ (resp. $\preccurlyeq_s$) if:*

1. $\forall m, t \in \mathbb{N}, z \in \mathbb{Z} : \Phi \in \mathcal{I} \iff \Phi^{\langle m, t, z \rangle} \in \mathcal{I}$;

2. $\Phi_2 \in \mathcal{I} \wedge \Phi_1 \sqsubseteq \Phi_2$ (resp. $\Phi_1 \trianglelefteq \Phi_2$) $\implies \Phi_1 \in \mathcal{I}$;

3. $\Phi_1 \in \mathcal{I} \wedge \Phi_2 \in \mathcal{I} \implies \Phi_1 \times \Phi_2 \in \mathcal{I}$.

Finally, let us notice that the ideal of reversible CAs is principal, as shown in [DMOT11].

1.4 Communication complexity

Communication complexity is a computational model designed by A. C.-C. Yao in [Yao79] to study parallel programs. In this framework, we consider two players, Alice and Bob, each with an arbitrarily high computational power, communicating to compute the value of some function $f : X \times Y \to Z$. We say that f has communication complexity c if, in the best protocol we can design to compute f on all possible inputs $(x, y) \in X \times Y$, where Alice only knows x, and Bob only knows y, they communicate at most c bits to decide the value of $f(x, y)$.

A more detailed introduction to this framework may be found in [KN97]. Here we just sum up the results and definitions important for our study. First we define what a *protocol* is.

Definition 8 *A* protocol $\mathcal{P}$ *over a domain* $X \times Y$ *with range* Z *is a binary tree where each internal node* v *is labeled either by a map* $a_v : X \to \{0,1\}$ *or by a map* $b_v : Y \to \{0,1\}$*, and each leaf* v *is labeled either by a map* $A_v : X \to Z$ *or by a map* $B_v : Y \to Z$.

The internal nodes of the protocol tree model communications. If a node v is labeled with an a_v, Alice says one bit according to her input. If a node v is labeled with an b_v, Bob says one bit according to his input. If this bit is 0, they go on to the left child of node v. If it is 1, they go on to its right child. Not surprisingly, the value of protocols, or the functions they compute, is the label of the leaf Alice and Bob arrive to if they follow all the internal nodes of the protocol tree. Hence the following definition.

Definition 9 *The* value *of protocol* $\mathcal{P}$ *on input* $(x,y) \in X \times Y$ *is given by* $A_v(x)$ *(or* $B_v(y)$*) where* A_v *(or* B_v*) is the label of the leaf reached by the path over the tree which starts at the root, turns left if* $a_v(x) = 0$ *(or* $b_v(y) = 0$*), and turns right otherwise. We say that a protocol computes a function* $f : X \times Y \to Z$ *if for any* $(x,y) \in X \times Y$*, its value on input* (x,y) *is* $f(x,y)$.

We denote by $D(f)$ the (deterministic) communication complexity of a function $f : X \times Y \to Z$. It is the minimal cost of a protocol, over all protocols computing f, where the cost of a protocol is the depth of its corresponding tree.

In order to prove lower bounds on our constructions, we are going to use the following classical bounds on communication complexity (the proofs appear in [KN97]).

Proposition 2 *Let* $n \geq 1$ *be fixed. Let* EQ *and* DISJ *be the functions "equality" and "disjointness" defined from* $\{0,1\}^n \times \{0,1\}^n$ *to* $\{0,1\}$ *by:*

$$\mathrm{EQ}(x,y) = \begin{cases} 1 & \text{if } (\forall i)(x_i = y_i), \\ 0 & \text{otherwise.} \end{cases}$$

$$\mathrm{DISJ}(x,y) = \begin{cases} 1 & \text{if } (\forall i)(x_i y_i \neq 1), \\ 0 & \text{otherwise.} \end{cases}$$

Both problems have maximal communication complexity, i.e. $D(\mathrm{EQ}) \geq n$ *and* $D(\mathrm{DISJ}) \geq n$.

In [GMRT09], there is an explanation on how to turn computational problems into communicational ones. Here we just recall the corresponding definition.

Definition 10 *Let* $P : Q^+ \to Z$ *be a computational problem. The* communication complexity *of* P*, denoted* CC (P)*, is the function:*

$$n \mapsto \max_{1 \leq i \leq n-1} D\big(P|_n^i\big).$$

2   Communication complexity and simulations

In this section, we continue the work begun in [GMRT09], incorporating two new communication problems to the three "canonical problems" developed there, and we try to extend the compatibility of these problems to the $\preccurlyeq_s$ relation.

In order to do this, we consider the following relation between functions from $\mathbb{R}_+$ to $\mathbb{R}_+$:

$$f_1 \prec f_2 \iff \exists \alpha, \beta, \gamma \text{ increasing affine functions}, f_1 \circ \alpha \leq \beta \circ f_2 \circ \gamma.$$

Also, we use the same notation than in [GMRT09] to represent periodic configurations: if $u = u_1 \ldots u_l$ is a finite word we call p_u the infinite configuration where for all $i \in \mathbb{Z}$, $(p_u)_i = u_{i \bmod l}$. Overmore, we denote $p_u(x_1, \ldots, x_n)$ the configuration obtained by modifying p_u as follows:

$$(p_u(x_1, \ldots, x_n))_i = \begin{cases} (p_u)_i & \text{for } i \leq 0 \text{ or } i \geq n+1, \\ x_i & \text{otherwise.} \end{cases}$$

The problem called INVASION in [GMRT09], has a good behavior with respect to $\preccurlyeq_i$. Here, we choose to rename it in order to avoid confusions.

Definition 11 (Spatial invasion (SINV) [GMRT09]) *Let Φ be a cellular automaton, and u a finite configuration for Φ. The problem SINV^u_Φ consists in determining whether the differences between p_u and $p_u(x)$ will expand to an infinite width as times tends to infinity when applying Φ (the answer 1 means yes and the answer 0 means no).*

Proposition 3 *Let Φ and Ψ be two cellular automata. If $\Phi \preccurlyeq_i \Psi$, then for all $u \in Q^+_\Phi$ there exists $v \in Q^+_\Psi$ (the corresponding word by $\preccurlyeq_i$), such that:*

$$\text{CC}\left(\text{SINV}^u_\Phi\right) \prec \text{CC}\left(\text{SINV}^v_\Psi\right).$$

Corollary 1 *If Ψ is intrinsically $\preccurlyeq_i$-universal, then there exists a word $u \in Q^+$ such that:*

$$\text{CC}\left(\text{SINV}^u_\Psi\right) \in \Omega(n).$$

2.1 Temporal invasion

Definition 12 (Temporal invasion (TINV) [GMRT09]) *Let Φ be a cellular automaton, and u a finite configuration for Φ. The TINV problem is the following:*

$$\text{TINV}^u_\Phi(x) = \forall t, [t \in \mathbb{N} \Rightarrow \Phi^t(p_u(x)) \neq \Phi^t(p_u)].$$

Proposition 4 *Let Φ and Ψ be two cellular automata. If $\Phi \preccurlyeq_i \Psi$, then for all $u \in Q^+_\Phi$ there exists $v \in Q^+_\Psi$ (the corresponding word by $\preccurlyeq_i$), such that:*

$$\text{CC}\left(\text{TINV}^u_\Phi\right) \prec \text{CC}\left(\text{TINV}^v_\Psi\right).$$

Proof: *As in the other cases, we need to decompose the simulation relation:*

- $\Phi^{\langle m,1,0 \rangle}$*: to simulate a protocol for Φ with a protocol for $\Phi^{\langle m,1,0 \rangle}$,* Alice *and* Bob *need to communicate $O(m)$ bits to describe the cell shared between them. The other direction is easy.*

- $\Phi^{\langle 1,t,0 \rangle}$*: the protocol is exactly the same, because of the "$\forall t$" in the definition of* TINV*.*

- $\Phi^{\langle 1,1,z \rangle}$*: this is still the same protocol than for Φ, since the worst case in the partition of the input will be the same.*

- $\Phi \sqsubseteq \Psi$*: here* Alice *and* Bob *both know the injection given in the simulation. Then, they can apply it and use a protocol for Ψ to solve* TINV *on a configuration of Φ, with no overhead.*

□

Proposition 5 *There is a cellular automaton Φ and a word $u \in Q^+$ such that $\mathrm{TINV}_\Phi^u \in \Omega(n)$.*

Proof: *We reduce* DISJ*, a classical problem in communication complexity. We build an automaton over alphabet $Q = \{\overrightarrow{0}, \overrightarrow{1}, \overleftarrow{0}, \overleftarrow{1}, \square, \boxtimes, u\}$ with the following transition table (read it from left to right, using the first rule that applies):*

Now let (x, y) an instance of DISJ*, i.e. two sets of $\{1, \ldots, n\}$. An easy recursion on n shows that* $\mathrm{DISJ}(x, y) \Leftrightarrow \neg (\mathrm{TINV}_\Phi^u(\rho(x, y)))$*, where $\rho(x, y)$ is the following configuration:*

$$\rho(x, y) = \overrightarrow{x_n} \ldots \overrightarrow{x_0} \square \overleftarrow{y_0} \ldots \overleftarrow{y_n}.$$

The recurrence hypothesis is: $\boxtimes$ appears in the orbit of $p_u(\rho(x, y))$ if and only if $x \cap y \neq \emptyset$ – remark that if $\boxtimes$ does not appear in any configuration of the orbit of $p_u(\rho(x, y))$, then all cells are in state u after a finite number of steps. □

Corollary 2 *If Ψ is intrinsically $\preccurlyeq_i$-universal, then there exists a word $u \in Q^+$ such that:*

$$\mathrm{CC}\left(\mathrm{TINV}_\Psi^u\right) \in \Omega(n).$$

2.2 Controlled invasion and incomparability

We shall see now a surprising connection between a well known open problem in communication complexity (the *direct sum conjecture*, see [KN97]), and the idea of "orthogonality" between the communication problems on cellular automata, introduced in [GMRT09].

Definition 13 (Controlled invasion (CINV)) *Let Φ be a cellular automaton, and u a finite configuration for Φ. The problem CINV_Φ^u is defined as follows:*

$$\mathrm{CINV}_\Phi^u(x) = \mathrm{TINV}_\Phi^u(x) \wedge \neg \mathrm{SINV}_\Phi^u(x)$$

Therefore, the output of $\mathrm{CINV}_\Phi^u(x)$ consists in determining whether the differences between p_u and $p_u(x)$ persists forever but remain bounded to a finite width $1 \leq w < \infty$ when applying Φ (the answer 1 means yes and the answer 0 means no).

We shall now prove a partial result of "orthogonality" (incomparability), in the sense used in [GMRT09]: for each of the three problems SINV, TINV and CINV, we may find an automaton where it is easy, but the other ones are hard.

Proposition 6 *None of the three problems SINV, TINV and CINV is stronger than the other ones.*

Proof:

- *Let Φ be an automaton and $u \in Q^+$ such that $\mathrm{TINV}_\Phi^u \in \Omega(n)$, and $\mathrm{SINV}_\Phi^u \in o(n)$. Then Φ must satisfy that $\mathrm{CINV}_\Phi^u \in \Omega(n)$. If not, knowing SINV_Φ^u and CINV_Φ^u, we could deduce TINV_Φ^u with less than $\Omega(n)$ bits (indeed, $\mathrm{TINV}_\Phi^u = \mathrm{CINV}_\Phi^u \vee \mathrm{SINV}_\Phi^u$).*

- *The same proof can be used to find an automaton Φ such that $\text{TINV}_\Phi^u \in o(n)$ and $\text{CINV}_\Phi^u \in \Omega(n)$, for all $u \in Q^+$.*

- *We describe here an automaton hard (i.e. with communication complexity in $\Omega(n)$) for TINV, SINV, but easy (in $O(1)$) for CINV. The idea is simple: we transform the construction of Proposition 5 by converting the $\boxtimes$ state into a spreading state. Also, we ensure the simplicity of the problem by making a $\boxtimes$ state appear each time the configuration is incorrect (i.e. not of the form $\rho(x, y)$ for some x and y). This way, we get:*

$$
\begin{aligned}
\text{DISJ}(x, y) &\Leftrightarrow \neg \text{TINV}_\Phi^u(\rho(x, y)) \\
\text{DISJ}(x, y) &\Leftrightarrow \neg \text{SINV}_\Phi^u(\rho(x, y)) \\
\text{CINV}_\Phi^u(\rho(x, y)) &= \text{TINV}_\Phi^u(\rho(x, y)) \wedge \neg \text{SINV}_\Phi^u(\rho(x, y)) \\
&= \bot
\end{aligned}
$$

Therefore, it follows that $\text{CC}\,(\text{CINV}_\Phi^u) \in O(1)$ in configurations of the form $p_u(\rho(x, y))$. On the other hand, in all the other configurations, a spreading state is generated and the configuration is always spatially invaded, thus $\text{CC}\,(\text{CINV}_\Phi^u) \in O(1)$.

$\square$

2.3   Non determinism and limit sets

In order to prove non-universalities for $\preccurlyeq_s$ simulation, we can use the same techniques that we used previously. For instance, it is relatively simple to see why the problem PRED (see [GMRT09]) will still work in this relation. However, problems that had slightly more subtle formulations, such as TINV, formulated as "does something change?", behave in an interesting way. First, it is necessary to design a cellular automaton that we'll use in several proofs below.

2.3.1   A convenient CA

Let $\Phi_{2.3.1}$ be a cellular automaton, product of three layers:

- The first layer operates on alphabet $Q_1 = \{\overrightarrow{0}, \overrightarrow{1}, \overleftarrow{0}, \overleftarrow{1}, \top, \bot, S\}$. The $\overrightarrow{x}$ signals move to the right, the $\overleftarrow{x}$ signals to the left. The $\top$ states change to $\bot$ whenever the symbols on its left and right are two 1s. The $\bot$ state never changes.

  In any other case, transitions result in the spreading state S.

- The second layer operates on alphabet $Q_2 = \{\leftharpoondown, \leftharpoonup, \blacklozenge\}$.

  Whenever —on the first layer— the $\top$ state has two 1s signals on its sides, a $\leftharpoondown$ is generated, moving to the left.

  When the signals' contents are other than two 1s, a $\leftharpoonup$ appears, also moving to the left.

  Also, $\blacklozenge$ is a quiescent state.

- The third layer is like the second one, but on alphabet $Q_3 = \{\rightharpoondown, \rightharpoonup, \blacklozenge\}$, moving to the right.

We shall argue now that this automaton has a trivial SINV problem, as well as a trivial TINV problem. Indeed, there are three cases for the background u:

1. If it has a $\overrightarrow{x}$ or $\overleftarrow{x}$ signal on the first layer, then all the other states must be signals in the same direction. In this case, the configuration is invaded if and only if the input is anything else than signals in this direction: a spreading state is generated on this component.

2. If it has a spreading state, no invasion can occur.

3. Otherwise, the background can be only $\diamond$, in which case invasion occurs, possibly on the second layer, if and only if the input is not only signals in the same direction.

In all three cases, the property can be checked by Alice and Bob with very few communicated bits, thus answering to the SINV problem. Since the configuration is changed, the TINV problem is also easy.

Proposition 7 *There is a cellular automaton Ψ such that $\Psi \trianglelefteq \Phi_{2.3.1}$ and:*

$$\mathrm{CC}\left(\mathrm{SINV}_\Psi\right) \in \Omega(n).$$

Proof: *If the quotient relation identifies states $\leftharpoondown$ and $\blacklozenge$ on the second component, $\rightharpoonup$ and $\blacklozenge$ on the third component, there is a set of configurations for which the problem becomes as difficult as the* DISJ *problem, defined in [KN97], i.e.* $\mathrm{CC}\left(\mathrm{SINV}_\Psi\right) \in \Omega(n)$. $\square$

Proposition 8 *There is a cellular automaton Ψ such that $\Psi \trianglelefteq \Phi_{2.3.1}$ and:*

$$\mathrm{CC}\left(\mathrm{TINV}_\Psi\right) \in \Omega(n).$$

Proof: *If the quotient relation identifies all states with $\blacklozenge$ on the second component and all states with $\blacklozenge$ on the third component, on configurations of the form $\overrightarrow{\divideontimes}{}^n \top \overleftarrow{\divideontimes}{}^n$, solving the problem* TINV *requires deciding if the middle $\top$ symbol turns $\bot$ somewhere in the space-time diagram, which is as difficult as the* DISJ *problem, defined in [KN97], i.e.* $\mathrm{CC}\left(\mathrm{TINV}_\Psi\right) \in \Omega(n)$. $\square$

This raises new questions: is this simulation stronger or weaker than the previous one? As studied in [DMOT11], we know that they are incomparable. In this section, we introduce a problem whose communication complexity grows with respect to $\preccurlyeq_s$, but for which it is not the case in relation $\preccurlyeq_i$. At the same time, it may be a clue that our approach with communication complexity will not be able to tell much about Open Problem 1.

Definition 14 (Limit set word (LIMIT)) LIMIT$_\Phi$ *is the problem of deciding if the input word belongs to the language of the limit set of Φ:*

$$\mathrm{LIMIT}_\Phi(x) = \begin{cases} 1 & \text{if } \forall t, \exists y, x = \Phi^t(y), \\ 0 & \text{otherwise.} \end{cases}$$

Let

$$\mathrm{CC}\left(\mathrm{LIMIT}_\Phi\right) = \max_i D\big(\mathrm{LIMIT}_\Phi(x_{[0,i]}, x_{[i+1,n-1]})\big)$$

be the deterministic communication complexity of this problem, and

$$\mathrm{NCC}\left(\mathrm{LIMIT}_\Phi\right) = \max_i N^1(\mathrm{LIMIT}_\Phi(x_{[0,i]}, x_{[i+1,n-1]}))$$

the non-deterministic version.

Now we need to show that the non-deterministic communication complexity of this problem grows with respect to the $\preccurlyeq_s$ relation. As in the other definitions (see [GMRT09], and Subsection 2.1), we only need to show that the complexity is preserved with each ingredient of the simulation:

Proposition 9 *If $\Phi \preccurlyeq_s \Psi$, then $\mathrm{NCC}\left(\mathrm{LIMIT}_\Phi\right) \prec \mathrm{NCC}\left(\mathrm{LIMIT}_\Psi\right)$.* **Proof:** *We showed that rescaling did not change the communication complexity of similar problems in the deterministic case; we can use exactly the same proof here, in the non-deterministic case.*

For the quotient relation, if $\varphi : Q_\Psi^{m_\Psi} \to Q_\Phi^{m_\Phi}$ is the quotient map, and $\overline{\varphi}$ is its uniform extension to infinite configurations, if at least one element x of $(\overline{\varphi})^{-1}(x)$ is in $\omega(\Psi)$, then $\overline{\varphi}(x) \in \omega(\Phi)$. Then, we use the non-determinism of the protocol to choose the correct traslation of each of the cells: at the first step of the protocol where Alice (resp. Bob) speaks, they take a non-deterministic step to choose a traslation of their configuration that leads to a positive answer if there is any. We insist on the fact that this step needs to non-deterministically choose a traslation and keep it for the rest of the protocol. □

Proposition 10 *There exists a cellular automaton Φ such that:*

$$\mathrm{NCC}\left(\mathrm{LIMIT}_\Phi\right) \in \Omega(n).$$

Proof: *We consider a cartesian product of three layers:*

1. *A shift to the left, i.e. σ over alphabet $\{0,1\}$.*

2. *A shift to the right, i.e. σ^{-1} over alphabet $\{0,1\}$.*

3. *A test layer with three states: a blank state, a "test" state, and a "corrupt test" state. This is a cellular automaton of radius 0, with the following rule:*

   - *The blank state remains blank.*

   - *The corrupt test remains corrupt.*

   - *Whenever a test sees a 1 on both of the two other layers, it gets corrupt. Otherwise, it remains a normal test.*
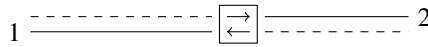
Fig. 1: An automaton hard for LIMIT.

We only need to find a set of configurations big enough, and hard for the LIMIT *problem: the configurations of odd size, with one test cell in the middle of the third layer, are hard. Indeed, for this configuration*

to be in the limit set, an infinite DISJ *problem needs to be solved between the word on layer 1, and the mirror of the word on layer 2 (i.e. on the whole lines on Figure 1), so* Alice *and* Bob *need to solve an instance of* DISJ *in order to find out whether the input belongs to the limit set, which, according to [KN97], requires* $\Omega(n)$ *bits to be solved by a non-deterministic protocol.* □

This shows that this problem is compatible with the $\preccurlyeq_s$ relation, i.e. that the following holds.

Corollary 3 *If* Ψ *is intrinsically* $\preccurlyeq_s$*-universal, then:*

$$\text{NCC}\left(\text{LIMIT}_\Phi\right) \in \Omega(n).$$

Now, according to the analysis of [GMT10], there exists a cellular automaton universal for $\preccurlyeq_i$, with complexity not greater than $O(\log n)$ for the LIMIT problem.

This problem may seem an odd counter-example. However, we will see in Proposition 14 how to use it to show that no stable automaton can be intrinsically $\preccurlyeq_s$-universal.

3   Structuring communication problems

In this section, we explore the links between our approach using communication complexity to study cellular automata and the general theory of bulking, developed in [DMOT10, DMOT11]. In order to do this, we prove the existence of new ideals and inclusions, giving a more detailed vision of quasi-orders induced by simulations. As showing that a given CA belongs to an ideal is a way to prove that it cannot be intrinsically universal, we illustrate here how our tools adapt well to this framework and extend it, being the communication approach, to our knowledge, the best way of proving non-universality in cellular automata.

3.1   Closing CAs

In the following, we ennunciate some results concerning the sets of closing and open CAs. As we suspect, this is closely related with the following open problem.

Open Problem 2 ([DMOT11]) *Is the ideal of surjective CA principal, and for which simulation quasi-order?*

Definition 15 (Asymptotic configurations [Ků09]) $x, y \in Q^{\mathbb{Z}}$ *are left (right) asymptotic, if there exists* $m \in \mathbb{Z}$ *such that* $x_i = y_i$ *for all* $i \leq m$ *(for all* $i \geq m$*).*

Definition 16 (Closingness [Ků09]) *A CA* Φ *is right closing (resp. left closing), if for every distinct left asymptotic (right asymptotic)* $x, y \in Q^{\mathbb{Z}}$, $\Phi(x) \neq \Phi(y)$. *A CA is closing if it is either left or right closing. Clearly,*

$$\Phi \text{ injective} \implies \Phi \text{ closing} \implies \Phi \text{ preinjective} \iff \Phi \text{ surjective}.$$

Theorem 1 *Let* $\preccurlyeq \in \{\preccurlyeq_i, \preccurlyeq_s\}$. *Then, the set of right closing (left closing) CAs is an ideal for* $\preccurlyeq$.

Proof: See the Appendix. □

Proposition 11 ([Hed69]) *A CA is open if and only if is right-closing and left-closing.*

Corollary 4 *Let* $\preccurlyeq \in \{\preccurlyeq_i, \preccurlyeq_s\}$. *Then, the set of open CAs is an ideal for* $\preccurlyeq$.

3.2 Stable CAs

As proved in [GMT10], the set of stable cellular automata is an ideal for $\preccurlyeq_s$. Nevertheless, its behavior with respect to relation $\preccurlyeq_i$ is unclear. We do not even know whether there is any stable universal automaton. However, the following results might help.

Lemma 1 *Let $\Phi \in \mathcal{AC}$. Then, if $\omega(\Phi)$ is an SFT, Φ is stable.*

Proof: *A SFT subshift is characterized by a finite set of forbidden words F. For each $w \in F$, by compactness, there is a first time step t_w in which word w does not appear anymore in $\Phi^{t_w}(Q^{\mathbb{Z}})$. Taking $t^* = \max_{w \in F} t_w$, it follows that $\omega(\Phi) = \Phi^{t^*}(Q^{\mathbb{Z}})$.* □

Lemma 2 *Let $\Phi \in \mathcal{AC}$. Then, if $\omega(\Phi)$ is an SFT, Φ is preinjective restricted to its limit set.*

Proof: *The result follows directly by considering the more general case of an onto sliding block code from an irreducible SFT (for a further explanation, see [LM95]).* □

Proposition 12 *Let Φ be a CA with SFT limit set. Then, for all $u \in Q^+$:*

$$\mathrm{CC}\left(\mathrm{TINV}_{\Phi}^{u}\right) \in O(1).$$

Proof: *By Lemma 1, all CA with SFT limit set are stable. Therefore, there exists a time t^* such that $\omega(\Phi) = \Phi^{t^*}(Q^{\mathbb{Z}})$. Then, the protocol just consists of iterating t^* times the periodic configuration and the perturbation (this has constant cost) and reach the limit set. Later,* Alice *and* Bob *only have to check if they have some difference with respect to the non perturbated pattern (which also has constant cost). As Φ is preinjective on $\omega(\Phi)$ (by Lemma 2), finite differences will remain forever. Therefore, the whole protocol has constant cost and stable intrinsically universal CAs cannot exist.* □

Corollary 5 *No cellular automaton with a SFT limit set can be intrinsically $\preccurlyeq_i$-universal.*

Proposition 13 *Let Φ be a CA with sofic limit set. Then:*

$$\mathrm{CC}\left(\mathrm{LIMIT}_{\Phi}\right) \in O(1).$$

Proof: *The language of the limit set $\omega(\Phi)$ of such a CA is regular, recognized by a finite automaton $\mathcal{A}$.*

Knowing both Alice *and* Bob *$\mathcal{A}$, the only thing* Alice *needs to say to solve LIMIT_{Φ} is the state she gets on it after having read her half of the configuration, if possible, and answer 0 elsewhere.* □

A result that follows directly from Proposition 13 is that no stable CA can be $\preccurlyeq_s$-universal, because every stable CA has a sofic limit set. However, they accept a different protocol with constant cost, besides the previous one for the more general case.

Proposition 14 *Let Φ be a stable CA. Then:*

$$\mathrm{CC}\left(\mathrm{LIMIT}_{\Phi}\right) \in O(1).$$

Proof: *If $\omega(\Phi) = \Phi^{t^*}(Q^{\mathbb{Z}})$ and (ϕ, r) is the canonical local representation of Φ,* Alice *only has to send to* Bob *the $2(t^* + 1)r$ rightmost bits of each of the possible antecedents of her input. There may be many of them, but there are just $|Q|^{2(t^*+1)r}$ combinations of the relevant parts of the configuration (which has constant cost for our purposes). Later,* Bob *can verify by his own if the input belongs to $\omega(\Phi)$ or don't.* □

Corollary 6 *No cellular automaton with a sofic limit set can be intrinsically $\preccurlyeq_s$-universal. In particular, no stable CA can be intrinsically $\preccurlyeq_s$-universal.*

Ideals	$\preccurlyeq_i$	$\preccurlyeq_s$
Sofic limit set	?	✓
Stable limit set	?	✓
SFT limit set	?	✓
Surjective	✓	✓
Closing	✓	✓
Open	✓	✓
Injective	✓	✓
Positive expansive	✓	?
Nilpotent over periodic configurations	✓	✓

Tab. 1: Relevant known ideals.

3.3   Communication ideals

The last proposition showed another example of a suprising correlation between ideals and simple protocols, that seems to generalize in a way that we were not completely able to formalize until now. Although the notion of uniformity among protocols, which we need here, seems difficult to formalize, probably due to the generality of the communication approach, the following proposition may be a first step in this direction:

Proposition 15 *Let $(X_\Phi)_{\Phi \in \mathcal{AC}}$ a family of communication problems, defined for each cellular automaton, of complexity increasing with respect to simulation $\preccurlyeq$ (i.e. if $\Phi \preccurlyeq \Psi$, then $\mathrm{CC}\,(X_\Phi) \prec \mathrm{CC}\,(X_\Psi)$). Let f be a non-decreasing function from $\mathbb{N} \to \mathbb{N}$. Then the following set is an ideal for $\preccurlyeq$:*

$$\mathcal{I} = \{\Phi \in \mathcal{AC} | \mathrm{CC}\,(X_\Phi) \prec f\}.$$

Now, what can be these new ideals? As the following example shows, their "shape" is quite undefined and might be complicated. Indeed, we showed in proposition 14 that stable CA had a simple protocol for LIMIT. Now we show an example of the same class of communication complexity, this time unstable:

Example 1 *Is easy to show that there exists an unstable CA Φ such that, for all $u \in Q^+$, $\mathrm{CC}\,(\mathrm{LIMIT}_\Phi) \in O(1)$. In fact, to see this, consider the "multiplication" automaton, on alphabet $\{0, 1\}$, given by the local rule:*

$$\phi(x_{i-1}, x_i, x_{i+1}) = x_{i-1} \cdot x_i \cdot x_{i+1}$$

This CA is unstable, since for all integer t, configurations of the form $^\infty 010^{2t}010^\infty$ have an antecedent by Φ^t but not by Φ^{t+1}. Now, it can be checked that for all $u \in \{0, 1\}^+$: $\mathrm{CC}\,(\mathrm{TINV}_\Phi^u) \in O(1)$. To see this, notice that a configuration $x \in \{0, 1\}^\mathbb{Z}$ converges to $^\infty 0^\infty$ if and only if $x \neq ^\infty 1^\infty$. Then, a protocol for LIMIT_Φ^u only has to check that Alice *and* Bob *have only 0s ore only 1s.*

This shows that communication complexity may allow us to describe a large number of complicated ideals in a really simple way. We are now just missing a finer definition of "class of protocols"...

4 Conclusion and perspectives

The theory of bulking and intrinsic universality in cellular automata is a fascinating topic, and communication complexity seems well suited to study this complexity.

Among the many open problems and perspectives, we would like to emphasize the following ones:

- How can we characterize a *stable family* of protocols? Each communication problem we studied until now was proved *increasing by simulation* in a way pretty similar to caracterizations of ideals. What are the exact relations, and how can simple protocols give us ideals "for free"?

- Although [GMT10] proved that there are few relations between the complexity of the limit set and intrinsic universality, it seems impossible that the limit sets of $\preccurlyeq_i$-instrinsic universal CAs be as simple as envisioned in that paper. An automaton with a sofic limit set can have sub-automata with more complex limit sets? What about SFT limit sets?

- Here, and for the first time, we were forced to introduce non-determinism in our proofs of compatibility between simulation and communication complexity. What does this tell us on the relation between the two simulations?

- Considering Example 1, how complex can be the limit set of an unstable CA Φ such that, for all $u \in Q^+$, $\mathrm{CC}\left(\mathrm{TINV}_\Phi^u\right) \in O(1)$?

- Until now, we only have used deterministic protocols to prove non-universality. Simulation $\preccurlyeq_s$ did not give us the choice: it seems that we really *need* non-determinism in the proof of proposition 9. Why does this happen? With what consequences?

- In [BR] is showed a way to generalize the framework of communication complexity, using relations instead of functions and giving an unification of some of the problems developed here and in [GMRT09]. What happen when non-determinism is used as in LIMIT? Is there any way to incorporate it to that technique?

Acknowledgement

References

[BR]   R. Briceño and I. Rapaport. Letting Alice and Bob choose which problem to solve: implications to the study of cellular automata (prepublication).

[BT10]   L. Boyer and G. Theyssier. On factor universality in symbolic spaces. In P. Hlinený and A. Kucera, editors, *MFCS*, volume 6281 of *Lecture Notes in Computer Science*, pages 209–220. Springer, 2010.

[DMOT10] M. Delorme, J. Mazoyer, N. Ollinger, and G. Theyssier. Bulking I: an abstract theory of bulking. HAL:hal-00451732, 2010.

[DMOT11]   M. Delorme, J. Mazoyer, N. Ollinger, and G. Theyssier. Bulking II: Classifications of cellular automata. *Theor. Comput. Sci.*, 412(30):3881–3905, 2011.

[GMRT09]   E. Goles, P.-E. Meunier, I. Rapaport, and G. Theyssier. Communication complexity and intrinsic universality in cellular automata. *CoRR*, abs/0912.1777, 2009.

[GMT10]   P. Guillon, P.-E. Meunier, and G. Theyssier. Clandestine simulations in cellular automata. *CoRR*, abs/1009.5621, 2010.

[Hed69]   G. A. Hedlund. Endomorphisms and automorphisms of the shift dynamical systems. *Mathematical Systems Theory*, 3(4):320–375, 1969.

[Jun09]   U. Jung. On the existence of open and bi-continuing codes. *arXiv:0810.4632v2*, 2009.

[KN97]   E. Kushilevitz and N. Nisan. *Communication complexity*. Cambridge university press, 1997.

[Ků09]   P. Kůrka. Topological dynamics of cellular automata. In *Encyclopedia of Complexity and Systems Science*, pages 9246–9268. Springer, 2009.

[LM95]   D. Lind and B. Marcus. *An introduction to symbolic dynamics and coding*. Cambridge University Press, 1995.

[Oll03]   N. Ollinger. The intrinsic universality problem of one-dimensional cellular automata. In *STACS*, pages 632–641, 2003.

[The05]   G. Theyssier. *Cellular automata : a model of complexities*. PhD thesis, ENS Lyon, 2005.

[Yao79]   A. C.-C. Yao. Some complexity questions related to distributive computing (preliminary report). In *STOC*, pages 209–213. ACM, 1979.

Appendix

Proof: ideal of closing CAs

Proof: In order to prove this, we adopt an enumeration like in Proposition 1.

1. Let $m, t \in \mathbb{N}, z \in \mathbb{Z}$. Then:

   - Φ *es right closing* $\iff b_m \circ \Phi \circ b_m^{-1}$ *is right closing*.
     In fact, if Φ is right closing, suppose there exist different left asymptotic configurations $x, y \in (Q^m)^{\mathbb{Z}}$ such that: $\Phi^{\langle m,1,0 \rangle}(x) = \Phi^{\langle m,1,0 \rangle}(y)$. Then,

     $$b_m \circ \Phi \circ b_m^{-1}(x) = b_m \circ \Phi \circ b_m^{-1}(y).$$

     As γ_m is bijective, it follows that:

     $$\Phi \circ b_m^{-1}(x) = \Phi \circ b_m^{-1}(y).$$

     Therefore, $b_m^{-1}(x), b_m^{-1}(y) \in Q^{\mathbb{Z}}$ are different left asymptotic configurations and their images via Φ are equal, which is a contradiction.
     On the other hand, suppose that $\Phi^{\langle m,1,0 \rangle}$ is right closing and there exist different left asymptotic configurations $x, y \in Q^{\mathbb{Z}}$, $x \neq y$ such that:

     $$\Phi(x) = \Phi(y)$$
     $$\implies \quad \Phi \circ b_m^{-1}(b_m(x)) = \Phi \circ b_m^{-1}(b_m(y))$$
     $$\implies \quad b_m \circ \Phi \circ b_m^{-1}(b_m(x)) = b_m \circ \Phi \circ b_m^{-1}(b_m(y))$$
     $$\implies \quad \Phi^{\langle m,1,0 \rangle}(b_m(x)) = \Phi^{\langle m,1,0 \rangle}(b_m(y)),$$

     but this is a contradiction, because $b_m(x), b_m(y) \in (Q^m)^{\mathbb{Z}}$ are different left asymptotic configurations.

   - Φ *is right closing* $\iff \Phi^t$ *is right closing*.
     Suppose that Φ is right closing. Then, if $x, y \in Q^{\mathbb{Z}}$ are different left asymptotic configurations, $\Phi(x), \Phi(y) \in Q^{\mathbb{Z}}$ too. Iterating the argument, it can be concluded that $\Phi^t(x) \neq \Phi^t(y)$.
     By other side, if Φ^t is right closing and $x, y \in Q^{\mathbb{Z}}$ are different left asymptotic configurations, it follows that $\Phi^t(x) \neq \Phi^t(y)$ and this implies that necessarily $\Phi(x) \neq \Phi(y)$.

   - Φ *is right closing* $\iff \sigma^z \circ \Phi$ *is right closing*.
     Notice that $x, y \in Q^{\mathbb{Z}}$ are different left asymptotic configurations if and only if $\sigma^z(x)$ and $\sigma^z(y)$ satisfy that, too. Therefore,

     $$\Phi(\sigma^z(x)) \neq \Phi(\sigma^z(x))$$
     $$\implies \quad \sigma^z \circ \Phi(x) \neq \sigma^z \circ \Phi(y)$$
     $$\implies \quad \Phi(x) \neq \Phi(y).$$

   All this by the commutativity of the shift and its bijectivity.

Then, composing all the partial results, we conclude.

2. For the two simulation relations:

- $\sqsubseteq$: let $\Phi_2 \in \mathcal{AC}$ be right closing and $\Phi_1 \in \mathcal{AC}$ such that $\Phi_1 \sqsubseteq_{\bar{\iota}} \Phi_2$. Then, if $x, y \in Q_1^{\mathbb{Z}}$ are distinct left asymptotic configurations, then $\bar{\iota}(x), \bar{\iota}(y) \in A_2^{\mathbb{Z}}$ satisfy that, too. Therefore,

$$\bar{\iota} \circ \Phi_1(x) = \Phi_2 \circ \bar{\iota}(x) \neq \Phi_2 \circ \bar{\iota}(y) = \bar{\iota} \circ \Phi_1(y).$$

Then, $\bar{\iota} \circ \Phi_1(x) \neq \bar{\iota} \circ \Phi_1(y)$, which implies that $\Phi_1(x) \neq \Phi_1(y)$.

- $\trianglelefteq$: the proof is in Subsection *Proof: quotient of closing CAs*.

3. Let $\Phi_1, \Phi_2 \in \mathcal{AC}$ be right closing. Then, their cartesian product $\Phi_1 \times \Phi_2$ is also right closing. In fact, let $(x_1, x_2), (y_1, y_2) \in Q_{\Phi_1 \times \Phi_2}^{\mathbb{Z}}$ be distinct left asymptotic configurations. Then, so does one of the pairs x_1, y_1 or x_2, y_2. Then, by the closingness of Φ_1 o Φ_2, respectively, it follows that:

$$\Phi_1 \times \Phi_2(x_1, x_2) = (\Phi_1(x_1), \Phi_2(x_2)) \neq (\Phi_1(y_1), \Phi_2(y_2)) = \Phi_1 \times \Phi_2(x_1, x_2).$$

The proof for the left closing case is analogous. □

Proof: quotient of closing CAs

Definition 17 (Entropy [LM95]) *Let X be a space shiftt. We define the* entropy *of X as follows:*

$$h(X) = \lim_{n \to \infty} \frac{1}{n} \log |\mathcal{L}_n(X)|.$$

Lemma 3 ([LM95]) *Let $\Phi : X \to Y$ be a sliding block code. Then, Φ is right closing if and only if:*

$$\exists N \in \mathbb{N} : x_{[-N,0]} = y_{[-N,0]} \wedge \Phi(x)_{[-N,N]} = \Phi(y)_{[-N,N]} \implies x_1 = y_1. \tag{1}$$

The case of a left closing CA is analogous.

Definition 18 *A sliding block code $\Phi : X \to Y$ is a 1-block code if it can be induced by a 1-block map with memory $m = 0$ and anticipation $a = 0$, namely, it exists $\phi : \mathsf{A}(X) \to \mathsf{A}(Y)$ such that:*

$$\Phi(x)_i = \phi(x_i),$$

for all $i \in \mathbb{Z}$, for all $x \in X$.

Definition 19 *A SFT X is said to be M-step if it can be defined by a family of forbidden words $F \subseteq A(X)^{M+1}$.*

Definition 20 *A sliding block code $\Phi : X \to Y$ is a conjugacy between X and Y if it is invertible. Two shift spaces are conjugated if there exists a conjugacy between them.*

Proposition 16 *Let $\Phi : X \to Y$ be a sliding block code. Then, exists a shift space $\tilde{X}$, a conjugacy $\pi : X \to \tilde{X}$ and a 1-block code $\tilde{\Phi} : \tilde{X} \to Y$ such that $\tilde{\Phi} \circ \pi = \Phi$, that is to say, the following diagram commutes:*

$$X \xrightarrow[\cong]{\pi} \tilde{X} \ . \tag{2}$$

$$\Phi \downarrow \quad \swarrow_{\tilde{\Phi}}$$

$$Y$$

Proof: Suppose that Φ have memory m, anticipation a and it is induced by a block map ϕ. Let $\pi : X \to \mathcal{L}_{m+a+1}(X)^{\mathbb{Z}}$ such that $\pi(x)_{[i]} = x_{[i-m,i+a]}$. Then, $\pi = \sigma^{-m} \circ \beta_{m+n+1}$. Therefore, $\tilde{X} = \pi(X) = X^{[m+n+1]}$ is a shift space and, because σ and β_{m+n+1} are conjugacies, π is a conjugacy, too. Considering $\tilde{\Phi} = \Phi \circ \pi^{-1}$, the result follows. Finally, note that $\tilde{\Phi}$ is a 1-block code. $\quad\square$

Definition 21 *Let S and T be two SFT. A 1-block code $\Phi : S \to T$ is said to be* e-right-resolving *if, given $b_1 b_2 \in \mathcal{L}(T)$ and $a_1 \in \mathsf{A}(S)$ such that $\phi(a_1) = b_1$, there exists $a_2 \in \mathsf{A}(S)$ such that $\phi(a_2) = b_2$ and $a_1 a_2 \in \mathcal{L}(S)$.*

Definition 22 *Let S and T be two SFT. A 1-block code $\Phi : S \to T$ is said to be* u-right-resolving *if, given $b_1 b_2 \in \mathcal{L}(T)$ and $a_1 \in \mathsf{A}(S)$ such that $\phi(a_1) = b_1$, there is one and only one $a_2 \in \mathsf{A}(S)$ such that $\phi(a_2) = b_2$ and $a_1 a_2 \in \mathcal{L}(S)$.*

Remark 1 *Clearly, every u-right-resolving 1-block code is e-right-resolving, too. Nevertheless, there exist examples where the other implication is not true.*

Sliding block codes which are right closing can be characterized as those which can be conjugated to an u-right-resolving 1-block code. There exists an analogous characterization between e-right-resolving 1-block codes and a family of sliding block codes called *right continuing*. Here we state the non trivial implication.

Proposition 17 *Let $\Phi : X \to Y$ be a right closing sliding block code. Then, there exists a space shift X', a conjugacy $\Theta : X' \to X$ and an u-right-resolving 1-block code Φ' such that $\Phi' \circ \Theta = \Phi$, that is to say, the following diagram commutes:*

$$X \xrightarrow{\Theta} X' \tag{3}$$

$$\Phi \downarrow \quad \swarrow_{\Phi'}$$

$$Y$$

Proof: *Without loss of generality, by Proposition 16, it can be considered Φ as a right closing 1-block code, because right closing property is an invariant under conjugacies. By Lemma 3, there exists $N \in \mathbb{N}$ such that:*

$$x_{[-N,0]} = x'_{[-N,0]} \wedge \Phi(x)_{[-N,N]} = \Phi(x')_{[-N,N]} \implies x_1 = x'_1.$$

Given N, consider the equivalence relation $\simeq$ defined over $\mathcal{L}_{2N+1}(X)$ as:

$$x_{-N} \cdots x_N \quad \simeq \quad x'_{-N} \cdots x'_N \quad \Longleftrightarrow \quad \begin{array}{l} x_{-N} \cdots x_0 = x'_{-N} \cdots x'_0 \\ \phi(x_i) = \phi(x'_i), \forall |i| \le N. \end{array}$$

Later, define $\mathsf{A}(X')$ *as* $\mathcal{L}_{2N+1}(X)/\simeq$*, this is to say:*

$$\mathsf{A}(X') = \left\{ W(x_{-N}\cdots x_0; y_{-N}\cdots y_N) : \begin{array}{ccc} x_{-N}\cdots x_0 & \in & \mathcal{L}(X) \\ y_{-N}\cdots y_N & \in & \mathcal{L}(Y) \end{array} \right\},$$

where:

$$W(x_{-N}\cdots x_0; y_{-N}\cdots y_N) = \left\{ x'_{-N}\cdots x'_N \in \mathcal{L}_{2N+1}(X) : \begin{array}{l} x_{-N}\cdots x_0 = x'_{-N}\cdots x'_0 \\ \phi(x_i) = \phi(x'_i), \forall |i| \leq N. \end{array} \right\}.$$

Then, define X' *as a 1-step subshift such that:*

$$W(x_{-N}\cdots x_0; y_{-N}\cdots y_N)W(x'_{-N}\cdots x'_0; y'_{-N}\cdots y'_N) \in \mathcal{L}_2(X')$$

if and only if:

$$x_{-N+1}\cdots x_0 = x'_N \ldots x'_1, y_{-N+1}\cdots y_N = y'_{-N}\cdots y'_{N-1},$$

being x'_0 *the state determined by* $x_{-N}\cdots x_0$ *and* $y_{-N}\cdots y_N$*, because the right closingness of* Φ*.*

Considering that, it is defined the 1-block code $\Theta : X' \to X$ *by the block map:*

$$\theta\left(W(x_{-N}\cdots x_0; y_{-N}\cdots y_N)\right) = x_0,$$

with local inverse given by:

$$\theta^{-1}\left(x_{-N},\ldots,x_N\right) = W(x_{-N}\cdots x_0; \phi(x_{-N})\cdots\phi(x_N)).$$

Finally, is defined the 1-block code $\Phi' : X' \to Y$ *by the block map:*

$$\phi'\left(W(x_{-N}\cdots x_0, y_{-N}\cdots y_N)\right) = y_N,$$

this is to say, $\Phi' = \Phi \circ \sigma^N \circ \Theta$*. Therefore,* Φ' *is u-right-resolving. In fact, suppose there exist* $y_N y_{N+1} \in \mathcal{L}(Y)$ *and* $W \in \mathsf{A}(X')$ *such that* $\phi'(W) = y_N$*. Then,* W *should have the following structure:*

$$W = W(x_{-N}\cdots x_0; y_{-N}\cdots y_N).$$

By definition of X'*, any* $W' \in \mathsf{A}(X')$ *such that* $WW' \in \mathcal{L}(X')$ *must satisfy that:*

$$W' = W(x_{-N+1}\cdots x_0 a; y_{-N+1}\cdots y_N b),$$

where a *and* b *must be determined. The value of* b *is determined by* y_{N+1} *and because* $\phi'(W') = y_{N+1}$*. Finally,* a *is determined (both existence and uniqueness) by the right closingness of* Φ*. Then, the result follows.* $\qquad\square$

Lemma 4 ([Jun09]) *Let* $\Phi : S \to T$ *be an e-right-resolving 1-block code, with* S *and* T *two irreducible SFT such that* $h(S) = h(T)$*. Therefore,* Φ *is u-right-resolving.*

Proposition 18 *Let* $\Phi : X \to X$ *and* $\Psi : Y \to Y$ *be two CAs, such that* Ψ *is right closing and* $\Phi \trianglelefteq \Psi$*. Therefore,* Φ *is right closing.*

Proof: Let $\varphi : A(X) \to A(Y)$ such that $\Phi \trianglelefteq_\varphi \Psi$. The hypothesis of the proposition can be represente by the following diagram:

$$
\begin{array}{ccc}
X & \xrightarrow{\varphi} & Y \\
\Psi \downarrow & & \downarrow \Phi \\
X & \xrightarrow{\varphi} & Y
\end{array}
\quad .
$$

By Proposition 16, there exist shift spaces $\tilde{X}$ and $\tilde{Y}$, conjugacies π_X and π_Y, and 1-block codes $\tilde{\Psi} : \tilde{X} \to X$ and $\tilde{\Phi} : \tilde{X} \to X$ such that $\tilde{\Psi} \circ \pi_X = \Psi$ and $\tilde{\Phi} \circ \pi_X = \Phi$, respectively. Then, we have the following completion of the previous diagram:

$$
\tilde{X} \xrightarrow{\pi_X^{-1}} X \xrightarrow{\varphi} Y \xrightarrow{\pi_Y} \tilde{Y} \quad .
$$

Applying the previous argument to $\pi_X^{-1} \circ \varphi \circ \pi_Y$, there exists a shift space $\overline{X}$, a conjugacy $\pi_{\tilde{X}}$ and a 1-block code $\overline{(\pi_X^{-1} \circ \varphi \circ \pi_Y)} : \overline{X} \to \tilde{Y}$ such that $\overline{(\pi_X^{-1} \circ \varphi \circ \pi_Y)} \circ \pi_{\tilde{X}} = \pi_X^{-1} \circ \varphi \circ \pi_Y$.

$$
\tag{4}
$$

Then, renaming variables ($X_1 = \overline{X}$, $X_2 = X$, $Y_1 = \tilde{Y}$, $Y_2 = Y$, $\varphi_1 = \overline{(\pi_X^{-1} \circ \varphi \circ \pi_Y)}$ y $\varphi_2 = \varphi$, $\Psi_X = \tilde{\Psi} \circ \pi_{\tilde{X}}^{-1}$) y $\Phi_Y = \tilde{\Phi}$), we can summarize with the following diagram:

$$
\begin{array}{ccc}
X_1 & \xrightarrow{\varphi_1} & Y_1 \\
\Psi_X \downarrow & & \downarrow \Phi_Y \\
X_2 & \xrightarrow{\varphi_2} & Y_2
\end{array}
$$

where $\Psi_X, \Phi_Y, \varphi_1$ y φ_2 are 1-block codes and Ψ_X is right closing. By Proposition 17, there exists a shift space X_1', a conjugacy $\Theta : X_1' \to X_1$ and an u-right-resolving 1-block code Ψ_X' such that $\Psi_X' \circ \Theta = \Psi_X$.

$$
\begin{array}{ccccc}
X_1' & \xrightarrow{\Theta} & X_1 & \xrightarrow{\varphi_1} & Y_1 \\
& \Psi_X' \searrow & \downarrow \Psi_X & & \downarrow \Phi_Y \\
& & X_2 & \xrightarrow{\varphi_2} & Y_2
\end{array}
$$

As Θ is a 1-block code, $\varphi_1' = \varphi_1 \circ \Theta$ is as well. Then, symplifying, we have a commuting diagram involving only 1-block codes and such that Ψ_X', φ_1' and φ_2 are e-right-resolving (because Ψ_X' is u-right-resolving and it can be verified φ_1' and φ_2 are e-right-resolving codes noting that X and Y are full-shifts) between irreducible SFT.

$$
\begin{array}{ccc}
X_1' & \xrightarrow{\varphi_1'} & Y_1 \\
\Psi_X' \downarrow & & \downarrow \Phi_Y \\
X_2 & \xrightarrow{\varphi_2} & Y_2
\end{array}
$$

Claim: Φ_Y is e-right-resolving. In fact, let $y_1^2 y_2^2 \in \mathcal{L}(Y_2)$ and $y_1^2 \in \mathsf{A}(Y_1)$ be such that $\phi_Y(y_1^1) = y_1^2$. As φ_1 is surjective (because it is a factor), there exists $x_1^1 \in \mathsf{A}(X_1)$ such that $\varphi_1(x_1^1) = y_1^1$. Then, by commutativity of the diagram, $y_1^2 = \phi_Y \circ \varphi_1(x_1^1) = \varphi_2 \circ \psi_X(x_1^1)$. On the other hand, it is easy to verify that the composition of two e-right-resolving 1-block codes is e-right-resolving as well. Next, we have that $y_1^2 y_2^2 \in \mathcal{L}(Y_2)$ and $\varphi_2 \circ \psi_X(x_1^1) = x_1^1$, and, by the e-right-resolving property of $\varphi_2 \circ \psi_X$, there exists $x_2^1 \in \mathsf{A}(X_1)$ such that $x_1^1 x_2^1 \in \mathcal{L}(X_1)$ and $\varphi_2 \circ \psi_X(x_1^1 x_2^1) = y_1^2 y_2^2$. Considering $y_2^1 = \varphi_2(x_2^1)$, the e-right-resolving property of Φ_X follows.

Finally, as Φ is surjective (because Ψ is right closing and Φ is a quotient of Ψ) and there are only conjugacies involved, we have that $h(Y_1) = h(Y_2)$, and, by Lemma 4, Φ_X is u-right-resolving. As Φ_X and Φ are conjugated, Φ is right closing, for a sliding block code is right closing if and only if it is conjugated to an u-right-resolving 1-block code. $\square$

Selfsimilarity, Simulation and Spacetime Symmetries

Vincent Nesme[1] and Guillaume Theyssier[2][†]

[1]*Freie Universität Berlin*
[2]*LAMA (CNRS, Université de Savoie),*
Campus Scientifique 73376 Le Bourget-du-Lac Cedex, France

We study intrinsic simulations between cellular automata and introduce a new necessary condition for a CA to simulate another one. Although expressed for general CA, this condition is targeted towards surjective CA and especially linear ones. Following the approach introduced by the first author in an earlier paper, we develop proof techniques to tell whether some linear CA can simulate another linear CA. Besides rigorous proofs, the necessary condition for the simulation to occur can be heuristically checked via simple observations of typical space-time diagrams generated from finite configurations. As an illustration, we give an example of linear reversible CA which cannot simulate the identity and which is 'time-asymmetric', i.e. which can neither simulate its own inverse, nor the mirror of its own inverse.

Keywords: cellular automata, simulation, reversibility, time symmetry, space symmetry, linear

1   Introduction and definitions

Cellular automata (CA) are well-known for the variety of behaviors they can exhibit. A lot of classification schemes where proposed in the literature, trying to make this variety of behaviors more intelligible [Wol84, Gil87, Kůr97]. Such classifications usually consist in a (finite) list of distinctive properties giving raise to a partition of the class of all CA. Another approach consists in defining a simulation relation between CA, and studying the ordered structure induced by the simulation. We follow this latter approach, and more precisely the simulation relation $\preccurlyeq_i$ defined in [DMOT11a, DMOT11b] giving rise to the notion of intrinsic universality [Oll08]. The intuition behind this simulation relation is simple: a CA is simulated by another if some rescaling of the first is a sub-automaton of a rescaling of the second.

More formally, we restrict ourselves to dimension 1 and the definition is as follows. A CA F is a *sub-automaton* of a CA G, denoted $F \sqsubseteq G$, if there is an injective map φ from A to B (state sets of F and G respectively) such that $\overline{\varphi} \circ F = G \circ \overline{\varphi}$, where $\overline{\varphi} : A^{\mathbb{Z}} \to B^{\mathbb{Z}}$ denotes the uniform extension of φ to configurations. We sometimes write $F \sqsubseteq_{\varphi} G$ to make φ explicit. This definition is standard but yields a very limited notion of simulation: a given CA can only admit a finite set of (non-isomorphic) CA as

sub-automata. Therefore, following works of J. Mazoyer, I. Rapaport and N. Ollinger [MR98, Oll02, DMOT11a, DMOT11b], we will add rescaling operations to the notion of simulation. The ingredients of rescaling operations are simple: packing cells into blocks, iterating the rule and composing with a translation (formally, we use shift CA σ_z, $z \in \mathbb{Z}$, whose global rule is given by $\sigma_z(c)_x = c_{x-z}$ for all $x \in \mathbb{Z}$). Given any state set $\mathcal{Q}$ and any $m \geq 1$, we define the bijective packing map $b_m : \mathcal{Q}^{\mathbb{Z}} \to \left(\mathcal{Q}^m\right)^{\mathbb{Z}}$ by:

$$\forall z \in \mathbb{Z} : \big(b_m(c)\big)(z) = \big(c(mz), \ldots, c(mz + m - 1)\big)$$

for all $c \in \mathcal{Q}^{\mathbb{Z}}$. The rescaling $F^{<m,t,z>}$ of F by parameters m (packing), $t \geq 1$ (iterating) and $z \in \mathbb{Z}$ (shifting) is the CA of state set $\mathcal{Q}^m$ and global rule:

$$b_m \circ \sigma_z \circ F^t \circ b_m^{-1}.$$

With these definitions, we say that F simulates G, denoted $G \preccurlyeq F$, if there are rescaling parameters m_1, m_2, t_1, t_2, z_1 and z_2 such that $G^{<m_1,t_1,z_1>} \sqsubseteq F^{<m_2,t_2,z_2>}$.

Determining whether some given CA simulates another given CA is hard (undecidable in general [DMOT11b, section 4.3]). For instance, looking at typical space time diagrams of two CA gives no clue on whether one simulates another, because the simulation can occur on a set of configurations of measure 0. Despite the general undecidability of the simulation relation, one can still hope to better understand its restriction to some specific classes of CA. For instance, the simulation relation is fully understood on products of shifts [DMOT11b, theorem 3.4] thanks to a 'characteristic sequence' which is essentially the sequence of ratio of translation vectors. Hence, if $F = \sigma_0 \times \sigma_1 \times \sigma_3$, one can prove that F cannot simulate $F^{-1} = \sigma_0 \times \sigma_{-1} \times \sigma_{-3}$ because they do not have the same characteristic sequence.

In this paper, we introduce a general necessary condition for a simulation between two CA to be possible. It focuses on surjective CA, but we will essentially use it on linear reversible CA. This condition is expressed as a characteristic set χ of points of the real half-plane which is decreasing w.r.t. $\preccurlyeq$ (Theorem 1 below):

$$F \preccurlyeq G \Rightarrow \chi(G) \subseteq \chi(F).$$

A striking property of χ is that it can be somewhat visualized on typical space time diagrams of linear CA. Moreover, the set χ is closely related to so-called 'Green functions' of linear CA for which systematic analysis techniques have been developed in [GNW10]. Hence, formal proofs of impossibility of simulation between two linear CA can be derived from heuristic observations of space-time diagrams in a quasi-automatic way.

The set of reversible CA is somewhat structured with respect to $\preccurlyeq$ since it possesses a maximal element (*i.e.,* a reversible universal CA [DMOT11b, theorem 4.5]) and verifies the following [DMOT11b, theorem 4.4]:

$$F \preccurlyeq G \Rightarrow F^{-1} \preccurlyeq G^{-1}$$

Therefore, a reversible CA is either $\preccurlyeq$-equivalent to its inverse, or $\preccurlyeq$-incomparable to it. The most complex reversible CA, reversible universal CA, are all $\preccurlyeq$-equivalent to their own inverse. Coming back to the example F above (product of shifts), we have that F and F^{-1} are $\preccurlyeq$-incomparable. Following [AN10], let us associate to every reversible CA F its dual $\tilde{F} = M \circ F^{-1} \circ M$, where M is the mirror transformation on configurations ($M(c)_z = c_{-z}$). Any product of shifts is self-dual, and generally speaking it seems to be hard to come up with CA that do not simulate their dual, while non-time-symmetric CA in the sense of [MG10] come in profusion.

An interesting question in this context is how different a reversible CA can be from its dual. As an illustration of the necessary condition for simulation between CA that is given by Theorem 1, we study in section 3 some reversible linear CA. The first one simulates its inverse, its mirror, its dual, but not the identity; the second one simulates neither the identity nor its inverse or its mirror image or its dual.

2   Simulation and geometry

The basic ingredient in this section is the collection of functions telling how a change of value of the center cell in the initial configuration will affect some other cell's value at some step in the future. Such functions are often studied for linear cellular automata (see section 3) and are sometimes called 'Green functions' in this context [Moo98].

Let F be any CA and fix some $x \in \mathbb{Z}$ and some $y \in \mathbb{N}$. For any configuration $c \in \mathcal{Q}^{\mathbb{Z}}$ and any $q \in \mathcal{Q}$, we denote by $\phi_c(q)$ the following configuration:

$$\phi_c(q)_z = \begin{cases} q & \text{if } z = 0, \\ c(z) & \text{else.} \end{cases}$$

We then denote by $F_{x,c}^y : \mathcal{Q} \to \mathcal{Q}$ the map $q \mapsto (F^y(\phi_c(q)))_x$.

For instance, if F is simply the identity, $F_{x,c}^y$ is the identity when $x = 0$, otherwise it is the constant function $q \mapsto c(x)$. For a less trivial example, consider the cas where $F = \bigoplus$ is the sum with neighborhood $\{0, -1\}$ over $\mathcal{Q} = \mathbb{Z}/2\mathbb{Z}$, i.e. $\bigoplus(c)_x = c(x) + c(x-1)$. Starting from a single nonzero cell, iterations of this automaton generate Pascal's triangle modulo 2. For $x \in \mathbb{N}$, let $x = \sum_{n \in \mathbb{N}} b_x(n) 2^n$, with $b_x(n) \in \{0, 1\}$, be its binary representation, and $B_x = \{n \in \mathbb{N} | b_x(n) = 1\}$. Then

$$\bigoplus_{x,c}^y (q) = \begin{cases} \bigoplus^y(c)_x + q & \text{if } x \geq 0 \text{ and } B_x \subseteq B_y \\ \bigoplus^y(c)_x & \text{else} \end{cases}.$$

We are interested in positions in space-time where the influence of the center cell is concentrated, whatever the initial configuration (see figure 1).

Definition 1 *F has the property $\mathrm{Spot}[x, y, l, r]$ for $x \in \mathbb{Z}$ and $y, l, r \in \mathbb{N}$ if*

- *$F_{x,c}^y$ is a bijection for all configurations c; and*

- *$F_{z,c}^y$ is a constant function for all c and all $z \in [x - l; x + r] \setminus \{x\}$.*

$\bigoplus$ thus fulfills $\mathrm{Spot}[x, y, l, r]$ if and only if, for any $z \in [x - l; x + r]$, $B_z \subseteq B_y$ is equivalent to $x = z$ (considering that "$B_z \subseteq B_y$" is a false statement when B_z is undefined).

Lemma 1 *If F has the property $\mathrm{Spot}[x, y, l, r]$, then $F^y(\mathcal{Q}^{\mathbb{Z}})$ contains all the words of size $\max(l, r) + 1$.*

Proof: Let us suppose, without loss of generality, that l is no larger than r. Let $\bar{q} = (q_0, \ldots, q_r) \in \mathcal{Q}^{r+1}$. We are going to construct $c \in \mathcal{Q}^{\mathbb{Z}}$ such that $F^y(c)_{0,\ldots,r} = \bar{q}$. Start with an arbitrary $c \in \mathcal{Q}^{\mathbb{Z}}$. We can first modify c_{r-x} in such way that $F^y(c)_r = q_r$; then we can change c_{r-x-1}, on which $F^y(c)_r$ does not depend, so that $F^y(c)_{r-1} = q_{r-1}$; and so on, until we choose c_{-x}, on which $F(c)_{1,\ldots,r}$ does not depend, so that $F(c)_0 = q_0$. □

For instance, $\bigoplus$ fulfills $\mathrm{Spot}[0, 1, +\infty, 0]$, which implies that it must be surjective.
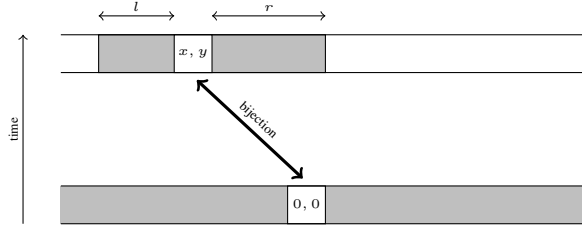
Fig. 1: Property $\mathrm{Spot}[x, y, l, r]$. Gray zones correspond to cells whose state does not change (either fixed in the initial configuration or kept constant in the y-th iteration of the CA).

Lemma 2 *Let $\mathcal{N}^y$ be the neighborhood of F^y. If F fulfills $\mathrm{Spot}[x, y, l, r]$ and $\mathrm{Spot}[x', y', l', r']$ with $[x' - l'; x' + r'] + \mathcal{N}^{y'} \subseteq [-l; r]$, then $\mathrm{Spot}[x + x', y + y', l', r']$ also holds.*

Proof: By definition of the neighborhood, $[x' - l'; x' + r'] + \mathcal{N}^{y'} \subseteq [-l; r]$ implies that $F^{y'}(c)_{[x'-l';x'+r']}$ is a function of $c_{[-l;r]}$. Applying that to $c = \sigma_{-x} \circ F^y(d)$, we get that the restriction of $F^{y+y'}(d)$ to $[x + x' - l'; x + x' + r']$ is a function of $F^y(d)_{[x-l;x+r]}$. Thus we get:

- $F^{y+y'}(d)_{[x+x'-l';x+x'+r'] \setminus \{x+x'\}}$ does not depend on d_0;

- $F^{y+y'}(d)_{x+x'}$ depends only on $F^y(d)_{[x-l;x+r]}$, which in turn, according to $\mathrm{Spot}[x, y, l, r]$, depends injectively on d_0.

$\square$

The central idea of the paper is to study the set of parameters (x, y, l, r) for which the property $\mathrm{Spot}[x, y, l, r]$ holds, and use that set to obtain necessary conditions for simulations between cellular automata. However, we won't use the set of parameters directly because the simulation relation is invariant by space-time rescalings and this set is not. Instead we will look at 'scale-free' structures inside this set of parameters. More precisely, given some integer p, we look for infinite geometric progressions of order p in the set of parameters. Hence we obtain a kind of fingerprint for each CA which is well-behaved with respect to space-time transformations involved in the simulation relation (Theorem 1 below). Moreover, as shown by examples developed latter in this paper, this fingerprint is closely related to the self-similar structure observed in typical space-time of some linear CA. Technically, this is how the definition goes.

Definition 2 *For a CA F and an integer $p \geq 2$, we denote $X_p(F)$ the set of points $(x, y) \in \mathbb{R} \times [0; +\infty)$ such that for some $k \in \mathbb{N}$, for every large enough $n \in \mathbb{N}$, F fulfills $\mathrm{Spot}[xp^n, yp^n, p^{n-k}, p^{n-k}]$.*

$X_2(\bigoplus)$ is for instance the set of points $(x, y) \in \mathbb{R} \times [0; +\infty)$ that can be written $x = \frac{a}{2^n}$ and $y = \frac{b}{2^n}$ with $a, b \in \mathbb{N}$ and $B_a \subseteq B_b$: its restriction to $\mathbb{R} \times [0; 1]$ is the dyadic part of a (shifted) Sierpiński triangle.

It can be noted that X_p is necessarily of measure 0, and is self-similar, since by Lemma 2 every point of X_p is the tip of a small copy of X_p within itself. One can also notice that if F is not surjective, then $X_p(F)$ is reduced to the singleton $\{(0, 0)\}$. Indeed, if $X_p(F)$ is not reduced to a singleton, then according to Lemma 1, the image of F contains every finite word, which implies, by compactness, that F is surjective.

We now detail, in a series of properties, how X_p is modified under the action of the transformations involved in the simulation of a CA by another. First, the shift. Let $\mathfrak{s}_z$ be the transformation of the plane defined by $\mathfrak{s}_z(x, y) = (x + zy, y)$. The following property is obvious, by definition of X_p.

Property 1 $X_p(\sigma_z \circ F) = \mathfrak{s}_z(X_p(F))$.

Let us now consider iteration and grouping. Let $\mathfrak{g}_t$ be the transformation of the plane defined by $\mathfrak{g}_t(x, y) = \left(x, \frac{y}{t}\right)$: notice that $\mathfrak{g}_p(X_p(F)) = X_p(F)$. Let $\mathfrak{f}_m$ be the transformation of the plane defined by $f_m(x, y) = \left(\frac{x}{m}, y\right)$.

Property 2 $X_p(F^t)$ *is a dense subset of* $\mathfrak{g}_t(X_p(F))$.

Proof: The inclusion is immediate from the definition. What might be slightly less immediate is why these sets are not obviously equal. Given the definition, $X_p(F)$ must be included in $\mathbb{R}_p^2$, where $\mathbb{R}_p$ is the set of real numbers having finite p-adic expansion. Actually, we do have $\mathfrak{g}_t(X_p(F)) \cap \mathbb{R}_p^2 = X_p(F^t)$, so the equality without the intersection is certainly true if $t \in \mathbb{R}_p$, not quite so in general. Let us now prove the density.

Let $(x, y) \in X_p(F)$. We want to find a sequence (x_n, y_n) of points of $X_p(F)$ converging to (x, y) such that for all n, $y_n \in t\mathbb{R}_p$. For a finite sequence of integers $0 = i_{n,0} < i_{n,1} < \ldots < i_{n,l}$ (l is a constant independent of n to be fixed later), we define $\eta_n = \sum_{j=0}^{l} p^{-i_{n,j}}$ and $(x_n, y_n) = \eta_n(x, y)$. We have three requirements:

- (x_n, y_n) must converge to (x, y): it is sufficient to have $\lim_{n \to +\infty} i_{n,1} = +\infty$

- (x_n, y_n) must be an element of $X_p(F)$. This is guaranteed as long as $i_{n,j+1} - i_{n,j}$ is always large enough. More precisely, by definition of X_p there exists some k such that for every large enough integer n, F fulfills $\mathrm{Spot}[xp^n, yp^n, p^{n-k}, p^{n-k}]$. Therefore, if $i_{n,l} - i_{n,l-1}$ is large enough (depending on k and the neighborhood of F), we get from Lemma 2 that $(1 + p^{-i_{n,l}+i_{n,l-1}})(x, y)$ is in $X_p(F)$. By recursion on l, we get ultimately $(x_n, y_n) \in X_p(F)$.

- y_n must be in $t\mathbb{R}_p$, which means the integer $p^{i_{n,l}} \sum_{j=0}^{l} p^{-i_{n,j}}$ must be a multiple of t.

So, it all boils down to finding increasing integer sequences $0 = i_0 < i_1 < \cdots < i_l$ where $i_{j+1} - i_j$ is arbitrary large, and such that t divides $p^{i_l} \sum_{j=0}^{l} p^{-i_j}$. That is clearly possible: the sequence of powers of p is ultimately periodic modulo t, so if we choose the i_j-s spaced by multiples of this period and $l = t$, we can easily meet the conditions. $\square$

Property 3 $X_p(b_m \circ F \circ b_m^{-1})$ *is a dense subset of* $\mathfrak{f}_m(X_p(F))$.

Proof: Let $G = b_m \circ F \circ b_m^{-1}$. Let $x \in \mathbb{Z}$ and $y, l, r \in \mathbb{N}$ with $l \geq 1$ and $r \geq 1$. First, it follows from definitions that, for any configuration c of F, $G^y_{x,b_m(c)}$ is constant if and only if, for all $z \in \{mx - m + 1, \cdots, mx + m - 1\}$, $F^y_{z,c}$ is constant. Moreover $G^y_{x,b_m(c)}$ bijective implies $F^y_{mx,c}$ bijective. This shows that if G has property $\mathrm{Spot}[x, y, l, r]$ then F has property $\mathrm{Spot}[mx, y, ml, mr]$.

Now suppose that F has property $\mathrm{Spot}[mx, y, ml, mr]$ and fix some configuration c of F. Then it is straightforward to check that $G^y_{x,b_m(c)}$ is bijective (because it sends each component of $\mathcal{Q}^m$ on itself) and $G^y_{z,b_m(c)}$ is constant for any $z \in [x - l; x + r] \setminus \{x\}$.

We have shown that F has property $\mathrm{Spot}[mx, y, ml, mr]$ if and only if G has property $\mathrm{Spot}[x, y, l, r]$. Thus we have

$$(x, y) \in X_p(G) \iff (mx, y) \in X_p(F).$$

This implies $X_p(G) \subseteq \mathfrak{f}_m(X_p(F))$. To prove the density, it is sufficient to prove that $X_p(F) \cap m\mathbb{R}_p$ is dense in $X_p(F)$ which can be done using the same argument as in the proof of property 2. $\qquad\square$

It only remains to consider the case of the sub-automaton.

Property 4 *If $G \sqsubseteq F$ then $X_p(F) \subseteq X_p(G)$.*

Proof: It is straightforward to check that if F has property $\mathrm{Spot}[x, y, l, r]$ then so does G. The property follows. $\qquad\square$

Properties 1, 2, 3 and 4 prove the following theorem.

Theorem 1 *If F simulates G, then there exist rational numbers β and $\alpha, \gamma > 0$ such that for every integer $p \geq 2$, $\pi_{\alpha,\beta,\gamma}(X_p(F)) \subseteq X_p(G)$, where $\pi_{\alpha,\beta,\gamma}(x, y) = (\alpha x + \beta y, \gamma y)$.*

The determination of X_p is not easy in general, but the following basic facts can be established straight-forwardly from the definitions:

- if F is a shift, then $\overline{X_p(F)}$ is a line passing through the origin;

- if F is nilpotent (*i.e.* $\exists t$ s.t. F^t is a constant function), then $X_p(F) = \{(0, 0)\}$;

- $X_p(F \times G) = X_p(F) \cap X_p(G)$.

Theorem 1 above shows that $X_p(F)$ represent obstructions for F to simulate other CA: the bigger $X_p(F)$ is, the smaller the family of CA F can simulate. Using the basic facts above, we can give some concrete formulations of this intuition.

Corollary 1 *Let $p \geq 2$ be an integer and F a CA. Then we have:*

- *If F simulates the identity, then $\overline{X_p(F)}$ must be included in a line passing through the origin;*

- *If F is intrinsically universal, then $X_p(F) = \{(0, 0)\}$;*

- *If F is reversible universal (i.e. it can simulate any reversible CA), then $X_p(F) = \{(0, 0)\}$;*

Proof: All items use Theorem 1. Item 1 and 2 are direct consequences of the computation of X_p for the identity and nilpotent CA (an intrinsically universal CA must simulate any nilpotent CA). Item 3 uses the fact that a reversible universal CA must simulate $\sigma \times \sigma^{-1}$ whose X_p is a singleton. $\qquad\square$

The purpose of the next section is to focus on a class of CA that generally have more interesting X_p: linear cellular automata.
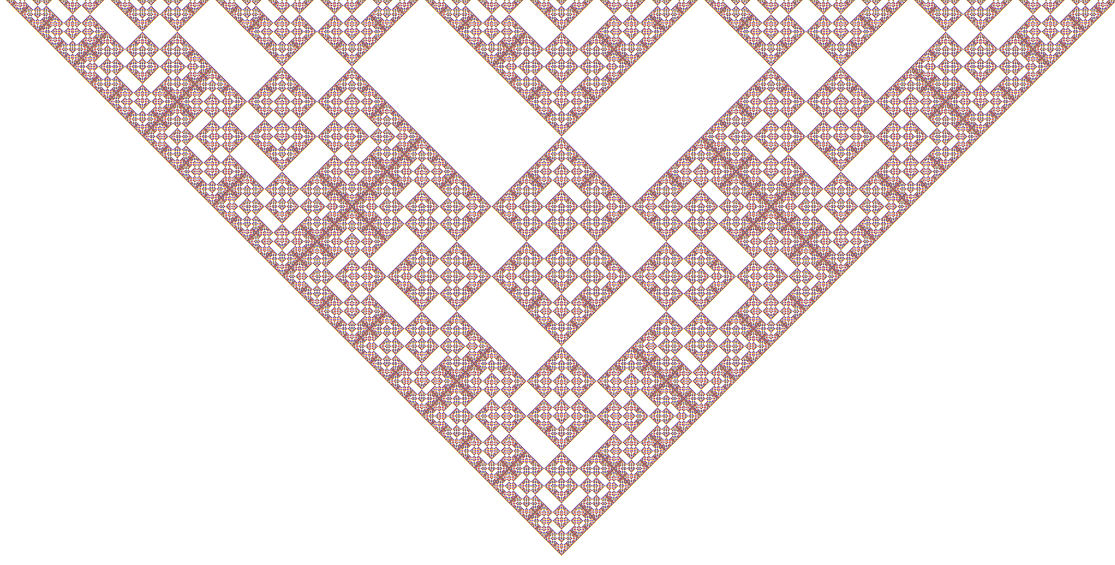
Fig. 2: Spacetime diagram of Θ up to a large power of 2. Also $\overline{X_2(\Theta)}$. Time goes from bottom to top

3 Linear Cellular Automata

More often than not, one can get a good idea about what $\overline{X_p}$ looks like just by examining the spacetime diagram. We think in particular of linear CA in the sense of [GNW10]. In this case, $\mathcal{Q} = R^d$, where R is a finite abelian ring, and d some positive integer. The algebra of CA that are homomorphisms of $\left(R^d\right)^{\mathbb{Z}}$ is then isomorphic to $\mathcal{M}_d(R)[u, u^{-1}]$: read section 1 of [GNW10] for details.

If F is such a linear CA and if 0 denotes the neutral element of R^d, the sets X_p can be derived from the functions $F^y_{x,\overline{0}}$ where $\overline{0}$ denotes the uniform configuration everywhere equal to 0. Indeed, for any configuration c, we have:

$$F^y_{x,c} \text{ bijective (resp. constant)} \iff F^y_{x,\overline{0}} \text{ bijective (resp. constant)}$$

In the sequel we denote $F^y_{x,\overline{0}}$ by F^y_x. The remainder of this section focuses on reversible cellular automata.

3.1 Θ: a reversible CA which cannot simulate the identity

Let us look at a more interesting example. The alphabet is now $\left(\mathbb{Z}_2\right)^2$, and the transition is given by

$$\Theta = \begin{pmatrix} 0 & 1 \\ 1 & u^{-1} + 1 + u \end{pmatrix}.$$

Since it already serves as a red thread through [GNW10], we will pass very quickly on it. Let us notice here that, since its determinant is 1, it is reversible, and that its inverse is $\Theta^{-1} = \begin{pmatrix} u^{-1} + 1 + u & 1 \\ 1 & 0 \end{pmatrix}$.

Obviously, Θ simulates its own inverse: in fact $\Theta^{-1} \sqsubseteq_\varphi \Theta$ with $\varphi = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

Figure 2 represents the spacetime diagram of Θ up to a large power of 2, for an initial configuration consisting of one single nonzero cell. Θ is "well-behaved" in the sense that these spacetime diagrams, for increasingly large powers of 2, converge to $\overline{X_2(\Theta)}$. It thus gives in a sense a purely visual proof of the fact that Θ does not simulate the identity. Of course, this requires actually some background knowledge, in order for the proof to be correct. One must know that Θ is a linear CA, and that $\overline{X_p}$ actually corresponds to its limit spacetime diagram, or at least is not limited to one line. While X_p is not defined in [GNW10], the information given there on the way to describe the limit spacetime diagram by means of a substitution system justifies this assertion. The crucial point is that any block that is not empty contains a reduced copy of the whole pattern, which means that in the neighborhood of any non-white point in the limit spacetime diagram, there is a copy of the whole thing, whose tip is then a point in $X_2(\Theta)$; therefore $X_2(\Theta)$ is dense in this pattern. And so, adding that Θ is its own mirror image, we get:

Proposition 1 Θ *simulates its mirror, its inverse and its dual, but cannot simulate the identity.*

3.2 Γ*: a life in pictures*

Let us now provide an example of a CA that is both space- and time-asymmetric, in the sense that it cannot simulate any of the CA derived from it by inverting space and/or time. This will be

$$\Gamma = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & u \\ 1 & u & 0 \end{pmatrix} \in \mathscr{M}_3(\mathbb{Z}_2)[u, u^{-1}].$$

Its inverse is given by $\Gamma^{-1} = \begin{pmatrix} u^2 & u & 1 \\ u & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$. We are going to give only the proof that it does not simulate its inverse: the proof of the two other results would add only length to this article, and can surely be left as an exercise to the reader.

(a) $X_2(\Gamma)$ (b) $X_2(\Gamma^{-1})$
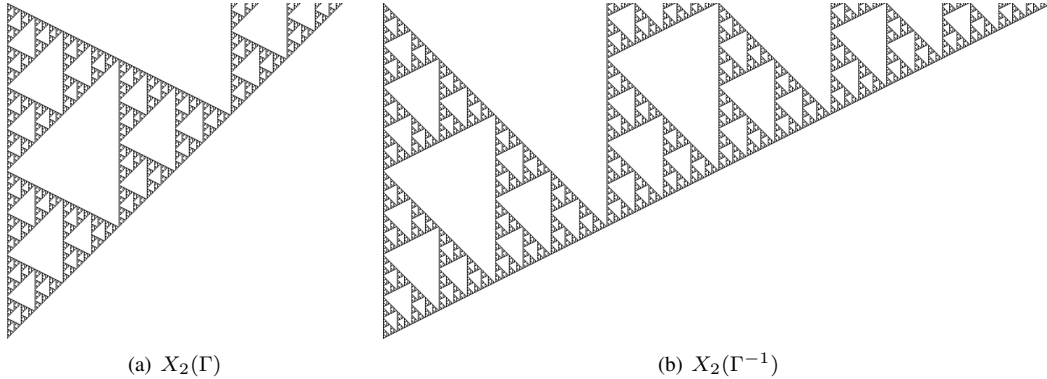
Fig. 3: X_2 with the second coordinate restricted to $[0, 1]$ (time goes from bottom to top)

Let us imagine for one blissful moment that we know $\overline{X_2}$ to be accurately represented by Figures 3(a)

and 3(b) (actually these figures are mirror images of spacetime diagrams up to a large power of 2). How do we conclude then?

Supposing that Γ simulates Γ^{-1}, we know from theorem 1 that for some $\alpha, \beta, \gamma, \pi_{\alpha,\beta,\gamma}(\overline{X_2(\Gamma)})$ should be included in $\overline{X_2(\Gamma^{-1})}$. Since there are only two lines passing through the origin in $\overline{X_2(\Gamma^{-1})}$, $\pi_{\alpha,\beta,\gamma}$ must send respectively $\mathbb{R}(0,1)$ and $\mathbb{R}(-1,1)$ on $\mathbb{R}(0,1)$ and $\mathbb{R}(-2,1)$, which implies $\beta = 0$. Now if we consider the lines joining these two axes, they have slope $\frac{1}{2}$ for Γ, 1 in Γ^{-1}, which means $\alpha = 2\gamma$. So, if Γ simulates its inverse, $\overline{X_2(\Gamma)}$ should be, modulo a change of scale, included into $\overline{X_2(\Gamma^{-1})}$, which is clearly not the case.

To make this proof rigorous, we need a tool to prove properties of X_2 for Γ and Γ^{-1}. We are going to follow section 3 of [GNW10], which gives a procedure to derive, from the transition matrix of the CA, a substitution system generating the Green functions (see Proposition 4 of [GNW10]). More precisely, we will associate to each CA F a 2×2 substitution system, that is a finite set E and a function $e : \mathbb{Z} \times \mathbb{N} \to E$ such that:

- F_x^y is a function of $e(x, y)$;

- for $i, j \in \{0, 1\}$, $e(2x + i, 2y + j)$ is a function of $e(x, y)$ and i and j.

The next two subsections give the substitution systems for Γ and Γ^{-1}, and subsection 3.2.3 uses them to formally prove negative result concerning simulation.

3.2.1 A substitution system for Γ

The minimal polynomial of Γ is $X^3 + X^2 + (1 + u^2)X + 1$, so we have the following recurrence relation.

$$\forall x \in \mathbb{Z} \forall n, y \in \mathbb{N} \qquad y < 3 \cdot 2^n \implies \Gamma_x^{3 \cdot 2^n + y} = \Gamma_x^{2^{n+1} + y} + \Gamma_x^{2^n + y} + \Gamma_{x-2^{n+1}}^{2^n + y} + \Gamma_x^y \qquad (1)$$

Now we define $\alpha_j(x, y)$ in the following way: these are the coefficients in $\mathbb{Z}/2\mathbb{Z}$ such that for every function $(x, y) \mapsto \Xi_x^y$ fulfilling equation (1) in lieu of Γ,

$$\Xi_x^y = \sum_{j=0}^{2} \sum_{i \in \mathbb{Z}} \alpha_j(x - i, y) \Xi_i^j. \qquad (2)$$

For every $x \in \mathbb{Z}$, $y \in \mathbb{N}$ and $s, t \in \{0, 1\}$, we have

$$\Xi_{2x+s}^{2y+t} = \sum_{i \in \mathbb{Z}} \alpha_0(x - i, y) \Xi_{2i+s}^t + \alpha_1(x - i, y) \Xi_{2i+s}^{2+t} + \alpha_2(x - i, y) \Xi_{2i+s}^{4+t}. \qquad (3)$$

In the case $s = t = 0$, we have the following derivation:

$$
\begin{aligned}
\Xi_{2x}^{2y} &= \sum_i \alpha_0(x - i, y) \Xi_{2i}^0 + \alpha_1(x - i, y) \Xi_{2i}^2 + \alpha_2(x - i, y) \Xi_{2i}^4 \\
&= \sum_i \alpha_0(x - i, y) \Xi_{2i}^0 + \alpha_1(x - i, y) \Xi_{2i}^2 + \alpha_2(x - i, y) \left(\Xi_{2i-2}^2 + \Xi_{2i-2}^1 + \Xi_{2i}^0 \right) \\
&= \sum_i \left(\alpha_0(x - i, y) + \alpha_2(x - i, y) \right) \Xi_{2i}^0 + \alpha_2(x - 1 - i, y) \Xi_{2i}^1 \\
&\qquad\qquad\qquad + \left(\alpha_1(x - i, y) + \alpha_2(x - 1 - i, y) \right) \Xi_{2i}^2
\end{aligned}
\qquad (4)
$$

which is to be compared with the definition of α_j:

$$\Xi_{2x}^{2y} = \sum_i \sum_{j=0}^{2} \alpha_j(2x - i, 2y)\Xi_i^j. \tag{5}$$

The comparison shows that $\alpha_j(2x, 2y)$ is a function of $\alpha_j(x - i, y)$ for some values of i. Γ is peculiar in that $\alpha_j(2x + 1, 2y) = 0$, which simplifies our work. The same operation now has to be performed for $\alpha_j(2x, 2y + 1)$.

$$
\begin{aligned}
\Xi_{2x}^{2y+1} &= \sum_i \alpha_0(x - i, y)\Xi_{2i}^1 + \alpha_1\Xi_{2i}^3 + \alpha_2(x - i, y)\Xi_{2i}^5 \\
&= \sum_i \alpha_0(x - i, y)\Xi_{2i}^1 + \alpha_1(x - i, y)\left(\Xi_{2i}^2 + \Xi_{2i}^1 + \Xi_{2i-2}^1 + \Xi_{2i}^0\right) \\
&\qquad\qquad + \alpha_2(x - i, y)\left(\Xi_{2i}^1 + \Xi_{2i-2}^1 + \Xi_{2i-4}^1 + \Xi_{2i-2}^0\right) \\
&= \sum_i \left(\alpha_1(x - i, y) + \alpha_2(x - 1 - i, y)\right)\Xi_{2i}^0 \\
&\quad + \left(\alpha_0(x - i, y) + \alpha_1(x - 1 - i, y) + \alpha_1(x - i, y) + \alpha_2(x - 2 - i, y)\right. \\
&\quad \left. + \alpha_2(x - 1 - i, y) + \alpha_2(x - i, y)\right)\Xi_{2i}^1 + \alpha_1(x - i, y)\Xi_{2i}^2
\end{aligned} \tag{6}
$$

Using the representation $\begin{array}{|c|} \hline \alpha_2 \\ \alpha_1 \\ \alpha_0 \\ \hline \end{array}$, we get the following substitution.

$$\boxed{\alpha.(x, y)}$$
$$\downarrow$$

$\alpha.(2x, 2y + 1)$	$\alpha.(2x + 1, 2y + 1)$
$\alpha.(2x, 2y)$	$\alpha.(2x + 1, 2y)$

$$\parallel$$

$\alpha_1(x, y)$	0
$\alpha_0(x, y) + \alpha_1(x - 1, y) + \alpha_1(x, y) + \alpha_2(x - 2, y) + \alpha_2(x - 1, y) + \alpha_2(x, y)$	0
$\alpha_1(x, y) + \alpha_2(x - 1, y)$	0
$\alpha_1(x, y) + \alpha_2(x - 1, y)$	0
$\alpha_2(x - 1, y)$	0
$\alpha_0(x, y) + \alpha_2(x, y)$	0

This needs some grouping; for instance, in the present situation, the substitution scheme uses $\alpha_1(x - 1, y)$, which is not an information contained in the initial cell. For instance, if we want to determine $\alpha_0(2x, 2y + t)$ for $t \in \{0, 1\}$, we need to know $\alpha_0(x, y)$, $\alpha_1(x, y)$, $\alpha_2(x, y)$ and $\alpha_2(x - 1, y)$. The smallest grouping that will allow us to carry all that information is

$$
\begin{array}{|cccc|}
\hline
\alpha_2(x - 3, y) & \alpha_2(x - 2, y) & \alpha_2(x - 1, y) & \alpha_2(x, y) \\
 & \alpha_1(x - 2, y) & \alpha_1(x - 1, y) & \alpha_1(x, y) \\
 & & \alpha_0(x - 1, y) & \alpha_0(x, y) \\
\hline
\end{array}.
$$

This gives us an alphabet of size $2^9 = 512$, and the substitution scheme is

$$
\begin{array}{cccc}
a & b & c & d \\
 & e & f & g \\
 & & h & i
\end{array}
$$

$$\downarrow$$

0	f	0	g	f	0	g	0
$a+b+c+e+f+h$	0	$b+c+d+f+g+i$		0	$b+c+d+f+g+i$	0	
	0	$c+g$			$c+g$	0	
0	$b+f$	0	$c+g$	$b+f$	0	$c+g$	0
	b	0	c		0	c	0
	0	$d+i$			$d+i$	0	

The initial state for this substitution system is $\cdots \boxed{0}\boxed{D}\boxed{0} \cdots$, and to a cell $\begin{array}{cccc} a & b & c & d \\ & e & f & g \\ & & h & i \end{array}$

in position (x,y) corresponds the Green function $\Gamma_x^y = \begin{pmatrix} d+i & c & g \\ c & b+d+i+g & c+f \\ g & c+f & b+d+i \end{pmatrix}$.

For a letter x in $\{a, b, \ldots, i\}$, let $\boxed{X}$ denote the cell where x has the value 1 whereas all other letters are set to 0. We can notice that A, E and H are completely equivalent: they all substitute to $\begin{array}{cc} E & 0 \\ 0 & 0 \end{array}$, and project onto 0 in the computation of Γ_x^y. We can therefore simplify this system a bit by putting $A = E = H = 0$:

Whereas we have a theoretical number of $2^5 = 32$ different states in the substitution scheme, only 11 of them are accessible from the initial state, namely 0 plus the ones represented in Figure 5. This graph has two strongly connected components, one composed of BD alone, the other of the remaining vertices. In particular, from any state of the substitution system that has been accessed from the initial state and that is neither 0 nor BD, there is a path to D; therefore there must be a point of X_2 in the corresponding square.

3.2.2 A substitution system for Γ^{-1}

We now have to perform the equivalent analysis for Γ^{-1}, which we will name Ω, in order to avoid possible confusions with negative exponents. The minimal polynomial of Ω is $X^3 + (1 + u^2)X^2 + X + 1$, so now the recurrence relation is

$$\forall x \in \mathbb{Z} \forall n, y \in \mathbb{N} \qquad y < 3 \cdot 2^n \implies \Omega_x^{3 \cdot 2^n + y} = \Omega_x^{2^{n+1}+y} + \Omega_{x-2^{n+1}}^{2^{n+1}+y} + \Omega_x^{2^n+y} + \Omega_x^y. \qquad (7)$$

We introduce β, which is to Ω what α was to Γ in Section 3.2.1.

$$\Xi_{2x+s}^{2y+t} = \sum_i \beta_0(x-i,y)\Xi_{2i+s}^t + \beta_1(x-i,y)\Xi_{2i+s}^{2+t} + \beta_2(x-i,y)\Xi_{2i+s}^{4+t} \qquad (8)$$
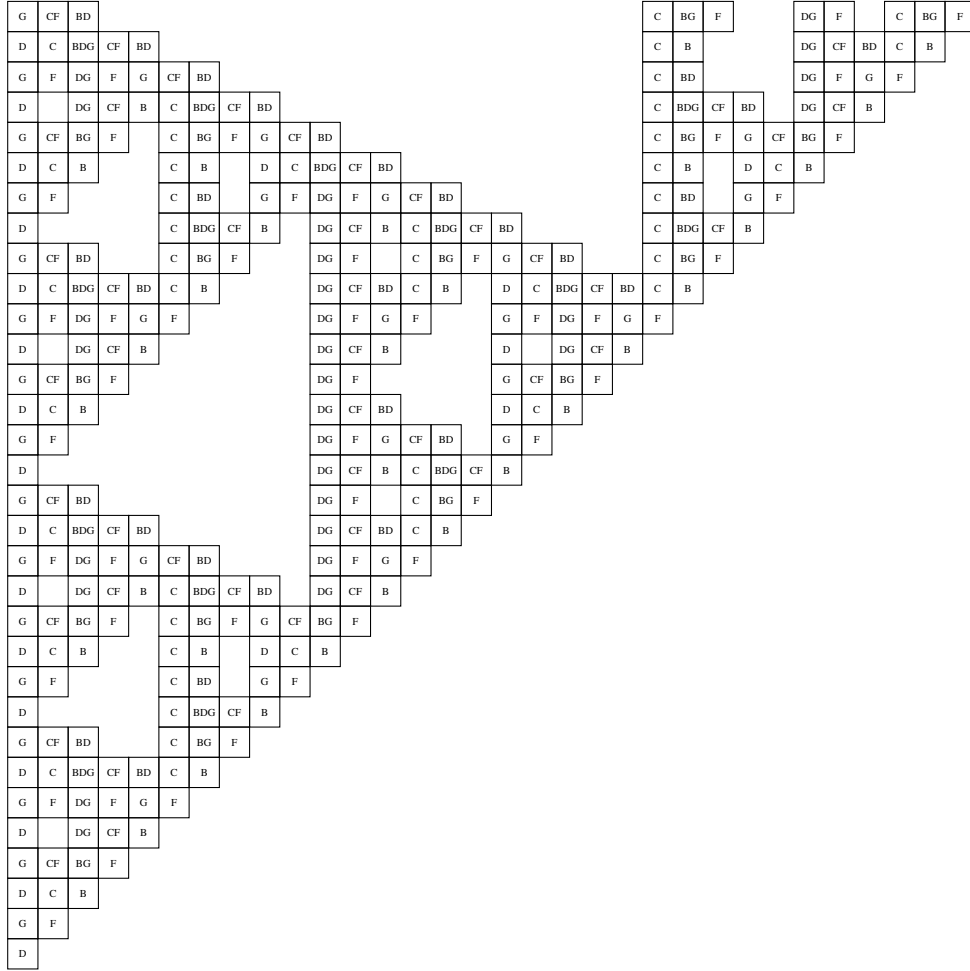
Fig. 4: Fifth step of Γ's substitution system (time goes from bottom to top).
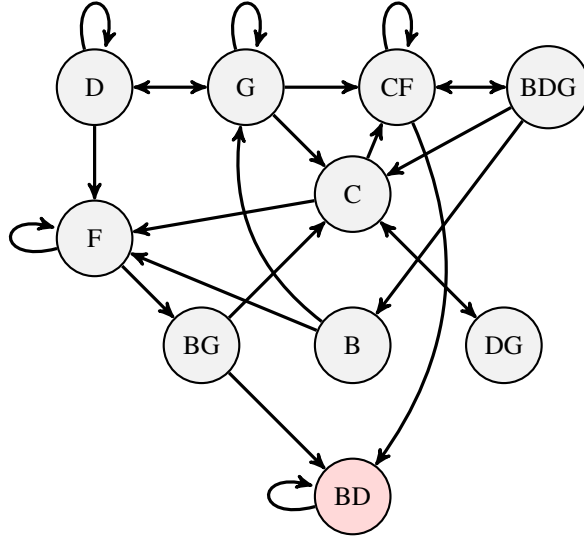
Fig. 5: Transition graph of the substitution system: an arrow from state s_1 to state s_2 means that s_2 can be obtained after a finite number of iterations starting from s_1.

We then get the following decompositions.

$$
\begin{aligned}
\Xi_{2x}^{2y} &= \sum_i \beta_0(x-i,y)\Xi_{2i}^0 + \beta_1(x-i,y)\Xi_{2i}^2 + \beta_2(x-i,y)\Xi_{2i}^4 \\
&= \sum_i \beta_0(x-i,y)\Xi_{2i}^0 + \beta_1(x-i,y)\Xi_{2i}^2 + \beta_2(x-i,y)\left(\Xi_{2i-4}^2 + \Xi_{2i-2}^1 + \Xi_{2i-2}^0 + \Xi_{2i}^0\right) \\
&= \sum_i \left(\beta_0(x-i,y) + \beta_2(x-1-i,y) + \beta_2(x-i,y)\right)\Xi_{2i}^0 + \beta_2(x-1-i,y)\Xi_{2i}^1 \\
&\quad + \left(\beta_1(x-i,y) + \beta_2(x-2-i,y)\right)\Xi_{2i}^2
\end{aligned}
$$

(9)

This is to be compared to this definition of β_j:

$$
\Xi_{2x}^{2y} = \sum_i \sum_{j=0}^{2} \beta_j(2x-i,2y)\Xi_i^j
$$

(10)

Likewise, for $y \mapsto 2y+1$, we get

$$
\begin{aligned}
\Xi_{2x}^{2y+1} &= \sum_i \left(\beta_1(x-i,y) + \beta_2(x-2-i,y)\right)\Xi_{2i}^0 \\
&\quad + [\beta_0(x-i,y) + \beta_1(x-i,y) \\
&\qquad + \beta_2(x-2-i,y) + \beta_2(x-1-i,y) + \beta_2(x-i,y)]\Xi_{2i}^1 \\
&\quad + [\beta_1(x-1-i,y) + \beta_1(x-i,y) \\
&\qquad + \beta_2(x-3-i,y) + \beta_2(x-2-i,y) + \beta_2(x-1-i,y)]\Xi_{2i}^2
\end{aligned}
$$

(11)

The minimal grouping is now...

$$\begin{array}{|cccccc|}
\hline
\beta_2(x-5,y) & \beta_2(x-4,y) & \beta_2(x-3,y) & \beta_2(x-2,y) & \beta_2(x-1,y) & \beta_2(x,y) \\
 & & \beta_1(x-3,y) & \beta_1(x-2,y) & \beta_1(x-1,y) & \beta_1(x,y) \\
 & & & & \beta_0(x-1,y) & \beta_0(x,y) \\
\hline
\end{array}$$

. . . and the corresponding substitution scheme is given by

$$\begin{array}{|cccc|}
\hline
a & b & c & d \quad e \quad f \\
 & g & h & i \quad j \\
 & & k & l \\
\hline
\end{array}$$

$$\downarrow$$

0	$a+b+c+g+h$	0	$b+c+d+h+i$	0	$c+d+e+i+j$	$a+b+c+g+h$	0	$b+c+d+h+i$	0	$c+d+e+i+j$	0
		0	$c+d+e+i+k$	0	$d+e+f+j+l$			$c+d+e+i+k$	0	$d+e+f+j+l$	0
				0	$d+j$					$d+j$	0
0	$b+h$		0	$c+i$	0	$d+j$	$b+h$	0	$c+i$	0	$d+j$	0
			0	d	0	e			d		e	0
				0	$e+f+l$					$e+f+l$	0

The initial state is $\boxed{\cdots \mid 0 \mid L \mid K \mid 0 \mid \cdots}$, and $\left(\Gamma^{-1}\right)^y_x$ is given by

$$\begin{pmatrix}
l+h+f+d+b & i+e+c & j+d \\
i+e+c & l+j+f+d & e \\
j+d & e & l+f
\end{pmatrix}.$$

A being equivalent to G, B to H and F to L, we get the simpler

$$\begin{array}{|ccc|}
\hline
c & d & e \\
g & h & i \quad j \\
 & k & l \\
\hline
\end{array}$$

$$\downarrow$$

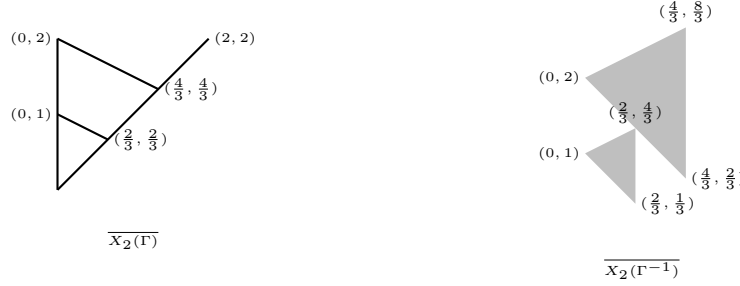| 0 | $c+d+h+i$ | 0 | | $c+d+h+i$ | 0 | $c+d+e+i+j$ | |
0	$d+e+g+h+i+k$	0	$d+e+j+l$	$d+e+g+h+i+k$	0	$d+e+j+l$	0
		0	$c+e+i$			$d+j$	0
0	$c+i$	0		$c+i$	0	$d+j$	
0	$d+h$	0	e	$d+h$	0	e	0
		0	$d+e+j+l$			$e+l$	0

This first simplification makes G equivalent to K, C to IK and J to DH, so we finally get

$$\begin{array}{|cc|}
\hline
d & e \\
h & i \\
k & l \\
\hline
\end{array}$$

$$\downarrow$$

$e+h+i+l$	0	0	$d+e+i$
$h+i+k+l$	0	0	$e+h+l$
0	$e+i$	$d+k$	0
$e+i$	0	0	d
$d+e+h$	0	0	$e+i$
0	$d+e+l$	$d+e+h+i+l$	0

. . . which results after five steps in Figure 6.

Fig. 6: Fifth step of Γ^{-1}'s substitution system (time goes from left to right).

Fig. 7: Partial knowledge about $\overline{X_2(\Gamma)}$ and $\overline{X_2(\Gamma^{-1})}$. Points in black are known to belong to the set while points in gray are known not to belong to the set. Remember that X_2 is invariant by homothetic transformations of center $(0,0)$ and factor 2^i (with i any integer).

3.2.3 Final arguments

It now remains to be proven that Figure 3 does represent X_2 for Γ and Γ^{-1}. As such, this does not mean much; actually, we need to prove a few features of X_2 that would suffice in order to conclude that Γ does not simulate Γ^{-1}. Namely, we want to justify this series of assertions:

(i) $\overline{X_2(\Gamma)}$ contains the (half-)lines $\mathbb{R}_+(0,1)$ and $\mathbb{R}_+(1,1)$.

(ii) $\overline{X_2(\Gamma)}$ contains the segment $\left[(0,1);(\frac{2}{3},\frac{2}{3})\right]$.

(iii) No point of $X_2(\Gamma^{-1})$ lies in the interior of the triangle with vertices $(0,1)$, $(\frac{2}{3},\frac{1}{3})$ and $(\frac{2}{3},\frac{4}{3})$.

This is enough to conclude, because (iii) implies that the only possible half-lines starting at the origin and included in $\overline{X_2(\Gamma^{-1})}$ are the vertical axis and that of slope $\frac{1}{2}$; and the segments joining these lines, if they exist, must have slope -1. Therefore it is impossible to send $X_2(\Gamma)$ into $X_2(\Gamma^{-1})$ by a $\pi_{\alpha,\beta,\gamma}$ transformation (see figure 7) and Theorem 1 concludes. Since $X_2(\tilde{\Gamma})$ is just the symmetric of $X_2(\Gamma^{-1})$ with respect to the vertical axis passing through $(0,0)$, the same reasoning with Theorem 1 shows that Γ cannot simulate $\tilde{\Gamma}$.

Proposition 2 Γ *can neither $\preccurlyeq$-simulate its inverse Γ^{-1} nor its dual $\tilde{\Gamma}$.*

We now prove the three assertions above successively using the substitution systems derived earlier.

Property 5 $\overline{X_2(\Gamma)}$ *contains the (half-)lines $\mathbb{R}_+(0,1)$ and $\mathbb{R}_+(1,1)$.*

Proof: First, by looking at the images of D, G, B and F by the substitution system of Γ, we prove by recurrence that:

- $\Upsilon(0,n) = D$ if n is even and G else, and

- $\Upsilon(n,n) = B$ if $n \geq 1$ is even and F else,

where Υ is the fixed-point of the substitution. We deduce from the former observation that every letter in the substitution system, except for BD, contains a point in X_2, that the (half-)lines $\mathbb{R}_+(0,1)$ and $\mathbb{R}_+(1,1)$ are in $\overline{X_2(\Gamma)}$.   □

Property 6 $\overline{X_2(\Gamma)}$ *contains the segment* $\left[(0,1); (\frac{2}{3}, \frac{2}{3})\right]$.

Proof: The substitution system of Γ is such that:

$$BD \rightarrow \begin{array}{|c|c|} \hline 0 & 0 \\ \hline BD & 0 \\ \hline \end{array} \qquad\qquad CF \rightarrow \begin{array}{|c|c|} \hline BD & 0 \\ \hline BDG & CF \\ \hline \end{array}$$

$$BDG \rightarrow \begin{array}{|c|c|} \hline G & CF \\ \hline B & C \\ \hline \end{array} \qquad\qquad G \rightarrow \begin{array}{|c|c|} \hline G & CF \\ \hline D & C \\ \hline \end{array}$$

We deduce that any pattern of the form

where X is either BDG or G, is sent to a pattern of the form

Now, observing Figure (4), one can see a discrete segment of slope $-\frac{1}{2}$ made of the above pattern starting from the top-left position and reaching the upper-diagonal. Since we know that all the cells appearing on this discrete segment, namely G, CF and BDG (plus an end point that is, depending on the parity of the scale, B or F), contain a point of X_2, it just remains to show by recurrence that a segment of this form is present at every scale, which is immediate. $\square$

Property 7 *No point of $X_2(\Gamma^{-1})$ lies in the interior of the triangle of vertices* $(0, \frac{1}{2})$, $(\frac{1}{3}, \frac{1}{6})$ *and* $(\frac{1}{3}, \frac{2}{3})$.

Proof: By induction, we can prove that depending on the parity of the step, this triangle takes alternatively the forms presented in Figures 8 and 9, which represent the corresponding triangle in the substitution system, supposing the pair of initial blocks represents a rectangle of height 1. For instance, Figure 6, showing the fifth step, exhibits in this position a triangle of the form presented in Figure 8.

The proof that each of the figures substitutes into the other one is purely mechanical, and essentially done by the very existence of Figure 6, where the first five steps of substitution are readable. $\square$

4 Discussion

We gave a new necessary condition for the simulation of CA and applied it to solve a few open questions of the form 'Does there exist a reversible CA that simulates such and such but not such and such?'.
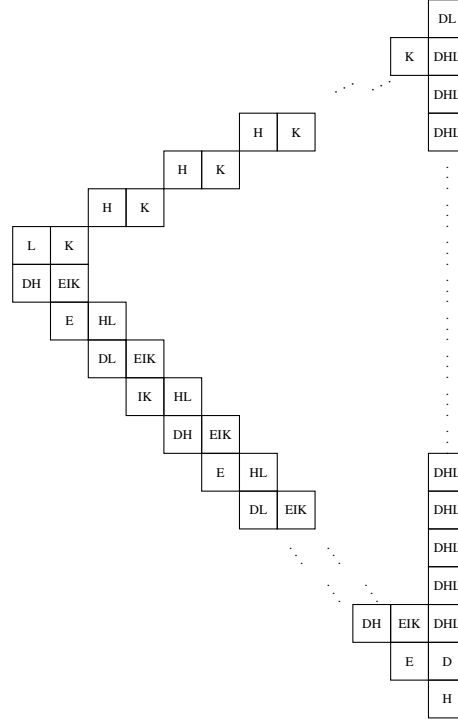
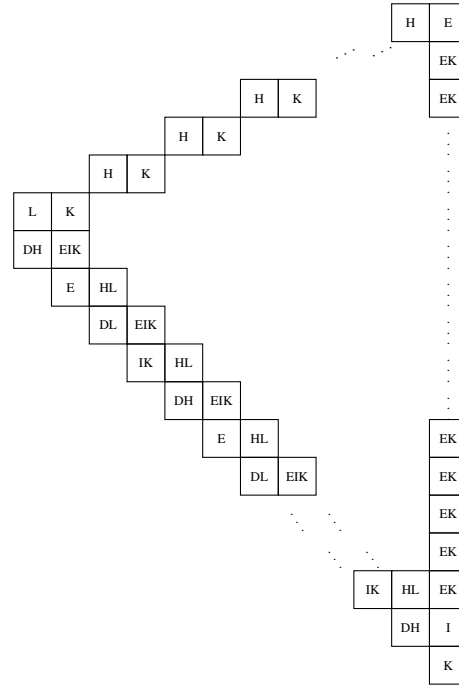**Fig. 8:** Odd steps (time goes from bottom to top).

Noticeably, each time we were able to answer this question, it was in the positive, which is one general reason why we would expect the same answer for other closely related questions of the same sort that remain open.

Our method is tailored to be applied to linear CA. Their practical advantage is that much of the information is present in their spacetime diagram, and therefore easy to access and comprehend. For instance, with our theorem in mind, a blink at Figure 2 is enough to suspect that Θ cannot simulate the identity. It then remains to check rigorously that the pattern does represent X_2 accurately, but that part is purely mechanical, if a bit tedious. Let us now finish with two questions.

Why did the authors resort to a 3×3 matrix? Couldn't they find anything simpler? No, they could not. Actually they conjecture that every 2×2 matrix simulates its inverse, which interestingly enough reduces to deciding whether every matrix simulates its transpose.

Does there exist a CA that can simulate the identity, but not its inverse/dual? The correct answer is 'probably, and $\Gamma \times \mathrm{id}$ is a good candidate'. However, our theorem is not really helpful in this case, since the X_p-s of this CA are trivial. Hopefully some hybrid can be created by merging it with [DMOT11b, theorem 3.4] and made available to the masses in the future.

Fig. 9: Even steps (time goes from bottom to top).

References

[AN10]   Pablo Arrighi and Vincent Nesme. The Block Neighborhood. In TUCS, editor, *Proceedings of JAC 2010*, pages 43–53, Turku, Finlande, December 2010.

[DMOT11a] Marianne Delorme, Jacques Mazoyer, Nicolas Ollinger, and Guillaume Theyssier. Bulking i: An abstract theory of bulking. *Theor. Comput. Sci.*, 412(30):3866–3880, 2011.

[DMOT11b] Marianne Delorme, Jacques Mazoyer, Nicolas Ollinger, and Guillaume Theyssier. Bulking ii: Classifications of cellular automata. *Theor. Comput. Sci.*, 412(30):3881–3905, 2011.

[Gil87]   Robert H. Gilman. Classes of linear automata. *Ergodic Theory and Dynamical Systems*, 7(1):105–118, 1987.

[GNW10]   Johannes Gütschow, Vincent Nesme, and Reinhard F. Werner. The fractal structure of cellular automata on abelian groups. In *Proceedings of Automata 2010*, pages 55–74, June 2010. Preprint: http://arxiv.org/abs/1011.0313.

[Kůr97]   Petr Kůrka. Languages, equicontinuity and attractors in cellular automata. *Ergodic Theory and Dynamical Systems*, 17:417–433, 1997.

[MG10]   Andrés Moreira and Anahí Gajardo. Time-symmetric cellular automata. In *JAC*, 2010.

[Moo98]   Cristopher Moore. Predicting nonlinear cellular automata quickly by decomposing them into linear ones. *Physica D: Nonlinear Phenomena*, 111(1-4):27–41, 1998.

[MR98]   Jacques Mazoyer and Ivan Rapaport. Inducing an order on cellular automata by a grouping operation. In *Proceedings of STACS*, pages 116–127, 1998.

[Oll02]   Nicolas Ollinger. *Automates Cellulaires : structures*. PhD thesis, École Normale Supérieure de Lyon, décembre 2002.

[Oll08]   Nicolas Ollinger. Universalities in cellular automata: a (short) survey. In B. Durand, editor, *Symposium on Cellular Automata Journées Automates Cellulaires (JAC'08)*, pages 102–118. MCCME Publishing House, Moscow, 2008.

[Wol84]   Stephen Wolfram. Computation theory of cellular automata. *Communications in Mathematical Physics*, 96(1):15–57, 1984.

Orbits of the Bernoulli measure in single-transition asynchronous cellular automata

Henryk Fukś and Andrew Skelton

Department of Mathematics and Statistics,
Brock University, St. Catharines, Canada.

We study iterations of the Bernoulli measure under nearest-neighbour asynchronous binary cellular automata (CA) with a single transition. For these CA, we show that a coarse-level description of the orbit of the Bernoulli measure can be obtained, that is, one can explicitly compute measures of short cylinder sets after arbitrary number of iterations of the CA. In particular, we give expressions for probabilities of ones for all three minimal single-transition rules, as well as expressions for probabilities of blocks of length 3 for some of them. These expressions can be interpreted as "response curves", that is, curves describing the dependence of the final density of ones on the initial density of ones.

Keywords: cellular automata. asynchronous rules, measure dynamics

1   Introduction

Mathematical theory of cellular automata can be developed using a variety of approaches. The most extensively used approach is the study of CA in the compact Cantor space $\mathcal{A}^{\mathbb{Z}}$ of symbolic sequences, where $\mathcal{A}$ is some finite alphabet. This approach proved to be very fruitful, and can now be considered a fully established sub-discipline of topological dynamics (Kůrka, 2009).

The aforementioned approach, however, is not without problems. Suppose, for example, that $F : \mathcal{A}^{\mathbb{Z}} \to \mathcal{A}^{\mathbb{Z}}$ is a CA rule with local function f, and σ is the shift map. Then F and $F\sigma$ determine different dynamical systems on $\mathcal{A}^{\mathbb{Z}}$, with possibly radically different properties. For instance, if F is the identity map, then it is obviously non-chaotic, yet $F\sigma = \sigma$ is chaotic. This is somewhat unsatisfactory in the view of the fact that σ is in some sense a "simple" map - it is, after all, just a translation.

To avoid this problem, one can study CA on non-compact spaces, and indeed this approach has been steadily gaining momentum in recent years (Formenti and Kůrka, 2009). Alternatively, the space of measures is often considered, or more precisely, the space $\mathcal{M}_{\mathcal{A}}$ of Borel shift-invariant probability measures on $\mathcal{A}^{\mathbb{Z}}$, equipped with the weak$\star$ topology. This space has the attractive property that F and $F\sigma$ determine the same dynamical system on $\mathcal{M}_{\mathcal{A}}$ and a number of interesting results have been established for dynamics of CA in $\mathcal{M}_{\mathcal{A}}$, *e.g.* by Kůrka and Maass (2000), Pivato (2002), Kůrka (2005), and others.

Among all measures in $\mathcal{M}_{\mathcal{A}}$, the uniform Bernoulli measure plays a special role in the dynamics of CA, first, because it is preserved by surjective CA and also, because it is a limit measure for linear CA (a

property known as "asymptotic randomization") – for a review, see Pivato (2009) and references therein. A very natural question is therefore to ask: what can we say about the orbit of the Bernoulli measure under a CA rule F? For linear CA, the asymptotic randomization result mentioned above answers this question to some extent, but what can be said about nonlinear rules?

The approach of the authors was to consider this problem in depth for concrete CA rules, starting from particularly simple cases. Even then, it is still difficult to fully characterize consecutive iterates of the Bernoulli measure. However, since any shift-invariant probability measure on $A^{\mathbb{Z}}$ is fully determined by its value on cylinder sets, it is often possible to compute measures of certain short cylinder sets after n iterations of F by taking advantage of the combinatorial structure of the CA rule. This works well for simple equicontinuous rules, as well as for almost-equicontinuous ones, as recently demonstrated for the case of almost-equicontinuous rule 172 (Fukś, 2010). Even for rules which are somewhat more complicated, such as the "traffic" rule 184 and its topological factor rule 142, significant results have been obtained (Fukś, 1999, 2006; Blank, 2003; Belitsky and Ferrari, 2005).

In this paper, we examine the same problem in the context of probabilistic rules. Can one compute iterates of the Bernoulli measure under simple probabilistic rules? Again, the situation appears to be similar as in the deterministic case. While the full characterization of iterates of the Bernoulli measure turns out to be very hard, measures of short cylinder sets can be computed explicitly if one takes advantage of the combinatorial structure of these rules. We will consider a special class of probabilistic rules, known as α-asynchronous CA. For the α-asynchronous version of a CA rule with local function f, one applies to each site rule f with probability α, or leaves the site unchanged with probability $1 - \alpha$, and this is done for each site simultaneously and independently. We will furthermore restrict our attention to particularly simple α-asynchronous rules, namely those for which the local function f differs from the local function of the identity rule only for one particular neighbourhood configuration.

2   Definitions

We first define probabilistic CA in a traditional way, as a stochastic process. Let $\mathcal{A} = \{0, 1\}$ be called a symbol set, and let elements of $\mathcal{A}^{\mathbb{Z}}$ be called configurations. Let $s(t) \in A^{\mathbb{Z}}$ denote a configuration at time t, where $t \in \mathbb{N}$. Suppose that we have a collection of random variables $X_{i,\mathbf{v}}$ taking values in A, indexed with $i \in \mathbb{Z}$ and $\mathbf{v} \in \mathcal{A}^3$.

We define define a nearest-neighbour binary probabilistic cellular automaton as a stochastic process

$$s_i(t + 1) = X_{i,\mathbf{v}(i,t)}, \tag{1}$$

where $\mathbf{v}(i, t) = \{s_{i-1}(t), s_i(t), s_{i+1}(t)\}$ will be called a *neighbourhood vector*. In general, the probability distribution of $X_{i,\mathbf{v}}$ is assumed to be independent of i, although it may (and normally does) depend on the neighbourhood vector $\mathbf{v}$.

We will be interested in a very special type of probabilistic CA, in which each cell is independently updated with some probability α. These rules were first studied experimentally by Fatès and Morvan (2005), and subsequently called called α-*asynchronous rules* (Fatès et al., 2006). They are formally defined as follows. Let $f : \mathcal{A}^3 \to \mathcal{A}$ be a given function and let $\alpha \in [0, 1]$ be a given parameter (called the *synchrony rate*). For these rules, random variables $X_{i,\mathbf{v}}$ take value in the set $\{f(v_1, v_2, v_3), v_2\}$ with

Wolfram code	Fatès code	Minimal rule
205	A	76
206	B	140
220	C	140
236	D	200
200	E	200
196	F	140
140	G	140
76	H	76

Tab. 1: Single transition rules.

probabilities, respectively, α and $1 - \alpha$, that is,

$$Pr\big(X_{i,\mathbf{v}} = f\left(v_1, v_2, v_3\right)\big) = \alpha, \tag{2}$$

$$Pr\big(X_{i,\mathbf{v}} = v_2\big) = 1 - \alpha, \tag{3}$$

for each $\mathbf{v} = \{v_1, v_2, v_3\} \in \mathcal{A}^3$ and $i \in \mathbb{Z}$. This can be understood as as probabilistic CA where at each site i we apply the local function f with probability α or leave the site unchanged with probability $1 - \alpha$, simultaneously and independently for all sites. For small α values and finite periodic configurations, this has an effect resembling asynchronous application of the rule f, hence the name (although in this paper we will be dealing with infinite configurations, so this feature will be irrelevant for us).

We wanted to understand at first only asynchronous rules for which the local function differs from the local function of the identity rule only for one neighbourhood configuration. These rules are shown in Table 2. Their Wolfram numbers are shown together with alternative designation as proposed by Fatès et al. (2006). The last column shows the so-called minimal rule number , that is, smallest rule number in the equivalency class which includes the given rule, its spatial reflection, the rule obtained by the interchange of 1's and 0s (Boolean conjugacy), and the rule obtained by the superposition of spatial reflection and Boolean conjugacy. Note that all these rules in Fatès notation are denoted by a single letter. Among them, only 76 (H), 140 (G), and 200 (E) are minimal and we will therefore consider only these rules. An asynchronous rule for which the local function f has Wolfram code W will be denoted by WA. We will therefore consider rules 76A, 140A, and 200A.

Note that the probabilistic cellular automaton can be fully defined if we specify the set of the so-called *transition probabilities*, to be denoted by

$$\omega\left(s_i(t+1)|s_{i-1}(t)s_i(t)s_{i+1}(t)\right), \tag{4}$$

and to be interpreted as the conditional probability that a site $s_i(t)$ with nearest neighbours $s_{i-1}(t)$ and $s_{i+1}(t)$ changes its state to $s_i(t+1)$ in a single time step. Using this concept, we can define a probabilistic cellular automaton as a dynamical system in the space of measures, as follows.

Let $\mathcal{M}_{\mathcal{A}}$ be a space of Borel shift-invariant probability measures on $\mathcal{A}^{\mathbb{Z}}$, equipped with the weak$\star$ topology. Let, for any block (word) $b = b_0 b_1 \ldots b_{r-1} \in \mathcal{A}^r$, $C_i(b)$ denote the cylinder set

$$C_i(b) = \{s \in \mathcal{A}^{\mathbb{Z}} : s_i = b_0, s_{i+1} = b_1, \ldots, s_{i+r-1} = b_{r-1}\}. \tag{5}$$

Since we are dealing with shift-invariant measures, we drop the spatial index i in expressions involving measures of cylinder sets. Note that the measure in $\mathcal{M}_\mathcal{A}$ is uniquely defined by its values on cylinder sets.

Suppose now that the function $\omega(\cdot|\cdot) : \mathcal{A} \times \mathcal{A}^3 \to [0,1]$ is given. We define the transformation $F : \mathcal{M}_\mathcal{A} \to \mathcal{M}_\mathcal{A}$ by defining, for any $\mu \in \mathcal{M}_\mathcal{A}$ and $c \in \mathcal{A}^r$,

$$(F\mu)\big(C(c)\big) = \sum_{b \in \mathcal{A}^{r+2}} \prod_{i=1}^{r} \omega(c_i|b_{i-1}b_ib_{i+1})\mu\big(C(b)\big), \tag{6}$$

where $r \in \mathbb{N}$.

For convenience, we also define *1-step block transition probability* $\boldsymbol{\omega}$ so that, for any $b = b_0b_1 \ldots b_rb_{r+1} \in \mathcal{A}^{r+2}$ and any $c = c_1c_2 \ldots c_{r-1}c_r \in \mathcal{A}^r$,

$$\boldsymbol{\omega}(c|b) = \prod_{i=1}^{r} \omega(c_i|b_{i-1}b_ib_{i+1}). \tag{7}$$

Moreover, we define a *n-step block transition probability* $\boldsymbol{\omega}$ recursively, so that, when $n \geq 2$ and for any blocks $b \in \mathcal{A}^{r+2n}$, $c \in \mathcal{A}^r$,

$$\boldsymbol{\omega}^n(c|b) = \sum_{b' \in \mathcal{A}^{r+2n-2}} \boldsymbol{\omega}(b'|b)\boldsymbol{\omega}^{n-1}(c|b'), \tag{8}$$

which may be written explicitly as

$$\boldsymbol{\omega}^n(c|b) = \sum_{\substack{b_{n-1} \in \mathcal{A}^{r+2(n-1)} \\ \vdots \\ b_1 \in \mathcal{A}^{r+2}}} \boldsymbol{\omega}(c|b_2) \left(\prod_{i=1}^{n-2} \boldsymbol{\omega}(b_i|b_{i+1})\right) \boldsymbol{\omega}(b_{n-1}|b). \tag{9}$$

Note that the n-step block transition probability $\boldsymbol{\omega}^n(c|b)$ can be intuitively understood as the conditional probability of seeing the block c on sites $[1,r]$ after n iterations of F, conditioned on the fact that the original configuration contained the block b on sites $[1-n, r+n]$.

3 Response Surface

Let us now suppose that the initial state $s(0)$ is not given explicitly, but that the state of each site is independently set to 1 with probability ρ or to zero with probability $1 - \rho$. This is equivalent to saying that the initial probability measure is a shift-invariant Bernoulli measure. We then apply our probabilistic CA n times, and ask: what is the resulting probability measure? Since it is well known that this measure is uniquely determined by its value on all cylinder sets, it would be sufficient to compute probabilities of occurrences of all finite blocks in order to describe the measure completely. This, however, is very difficult even in simple cases, thus we will restrict our attention to a much simpler problem, namely computing probabilities of short words, such as words of length one. One can say that such single-symbol probabilities provide only a very coarse description of the measure, yet they are often useful, just like knowledge of the first moment of some unknown distribution is often valuable. In many practical problems, e.g., in mathematical modeling, one wants to know how a CA rule iterated over an initial

configuration affects certain aggregate properties of the configuration, such as, for example, the density of 1's. For finite configurations, the density of ones is defined as the number of sites in state 1 divided by the total number of sites. For infinite configurations, which are the subject of this article, one could generalize this notion by taking the appropriate limit, but such limit may not always exist. Since we will be interested in orbits of translationally-invariant probability measures rather than individual configurations, it will be more convenient to define the density as the expected value of the cell state. For binary rules, if $P(0)$ and $P(1)$ are probabilities of occurrence of 0 and 1 in a configuration, the expected value of the cell state is $P(0) \cdot 0 + P(1) \cdot 1 = P(1)$. For this reason, we will use the term "density" interchangeably with the probability of occurrence of 1 in a configuration.

To be more precise, let μ_ρ be the Bernoulli measure such that $\mu_\rho(C(1)) = \rho$, $\mu_\rho(C(0)) = 1 - \rho$. Let us define *probability of occurrence* of block b after n iterations of F as

$$P_n(b) := (F^n \mu_\rho)\big(C(b)\big). \tag{10}$$

Using the concept of transition probabilities, we can write

$$P_n(b) = \sum_{a \in \mathcal{A}^{r+2n}} P_0(a)\boldsymbol{\omega}^n(b|a). \tag{11}$$

where the transition probability is defined in eq. (8) and (9). Note that $P_0(a)$ is easy to compute,

$$P_0(a) = \rho^{\# \text{ of 1's in } a}(1 - \rho)^{\# \text{ of 0's in } a}, \tag{12}$$

by the definition of Bernoulli measure.

Since some of the transition probabilities may be zero, we define, for any block $b \in G^r$, the set of *n-step block preimages*,

$$\operatorname{supp} \boldsymbol{\omega}^n(b|\cdot) = \{a \in \mathcal{A}^{r+2n} : \boldsymbol{\omega}^n(b|a) > 0\}. \tag{13}$$

Then we can write (11) as

$$P_n(b) = \sum_{a \in \operatorname{supp} \boldsymbol{\omega}^n(b|\cdot)} P_0(a)\boldsymbol{\omega}^n(b|a). \tag{14}$$

In what follows, we will show how to compute $P_n(1)$ for the three aforementioned asynchronous rules. For a given α and n, the graph of $P_n(1)$ versus $P_0(1)$ will be called *response curve*. We use this terminology analogous to signal processing theory: a probabilistic CA can be viewed as a black box, for which the input is given in the form of density of 1's in the initial measure ($P_0(1) = \rho$), and, after n iterations, we obtain output density, that is, $P_n(1)$. For the special case $\rho = 1/2$, we use the notation $P_n^{(s)}(1)$ which will be called a *symmetric response curve*. We will also plot $P_n(1)$ as a function of both ρ and the synchrony rate α, and this 3D graph will be called *response surface*. Most of the time, we will be interested in the limit $n \to \infty$, to be denoted as

$$P(1) = \lim_{n \to \infty} P_n(1), \tag{15}$$

$$P^{(s)}(1) = \lim_{n \to \infty} P_n^{(s)}(1). \tag{16}$$

4   Rule 200A

Consider an α-asynchronous rule defined as

$$\omega(1|b) = \begin{cases} 0 & \text{if } b \in \{000, 001, 100, 101\}, \\ 1 & \text{if } b \in \{011, 110, 111\}, \\ 1-\alpha & \text{if } b = 010, \end{cases} \tag{17}$$

and $\omega(0|b) = 1 - \omega(1|b)$ for all $b \in \mathcal{A}^3$. Note that if $\alpha = 1$, then this rule is equivalent to the deterministic Rule 200. In order to simplify notation, we define $\beta = 1 - \alpha$.

We wish to find a response surface for Rule 200A. In order to apply eq. (14), we begin by finding the set of all potential preimage blocks and their respective transition probabilities.

Proposition 4.1 *The set* $\operatorname{supp} \boldsymbol{\omega}^n(1|\cdot)$ *consists of all blocks of the form*

$$\{ \underbrace{\star \cdots \star}_{n} \ 1 \ \underbrace{\star \cdots \star}_{n} \}. \tag{18}$$

Proof: From eq. (17), we can see that an element in state 0 will always remain in state 0, so any block which does not have the above form will never be transformed to a single 1 under n iterations of Rule 200A. Similarly, a block in our set could produce a single 1, with some non-zero probability. $\quad\square$

We now define the following subset of $\operatorname{supp} \boldsymbol{\omega}^n(1|\cdot)$, $B_n = \{ \underbrace{\star \cdots \star}_{n-1} \ 010 \ \underbrace{\star \cdots \star}_{n-1} \}$.

Proposition 4.2 *For any block* $b \in \operatorname{supp} \boldsymbol{\omega}^n(1|\cdot) \setminus B_n$, *we have* $\boldsymbol{\omega}^n(1|b) = 1$.

Proof: In every element of the set $\operatorname{supp} \boldsymbol{\omega}^n(1|\cdot) \setminus B_n$, the central block will either be $011, 110$ or 111. From eq. (17), we can see that these blocks will always be preserved under application of Rule 200A. $\quad\square$

Proposition 4.3 *For any block* $b \in B_n$, *we have* $\boldsymbol{\omega}^n(1|b) = \beta^n$.

Proof: In each iteration, the 0s in the centre block will be preserved with probability 1, so we only need to consider the transition $010 \to 1$, which occurs in each iteration of Rule 200A with probability β. $\quad\square$

We may now use eq. (14) and consider the sets and transition probabilities described in Propositions 4.2 and 4.3, to conclude that

$$\begin{aligned} P_n(1) &= \sum_{b^* \in \operatorname{supp} \boldsymbol{\omega}^n(1|\cdot) \setminus B_n} P_0(b^*)\boldsymbol{\omega}^n(1|b^*) + \sum_{b^* \in B_n} P_0(b^*)\boldsymbol{\omega}^n(1|b^*) \\ &= 1 \cdot \left(2\rho^2(1-\rho) + \rho^3 \right) + \beta^n \cdot \rho(1-\rho)^2 \\ &= \rho^2(2-\rho) + \beta^n \rho(1-\rho)^2. \end{aligned} \tag{19}$$

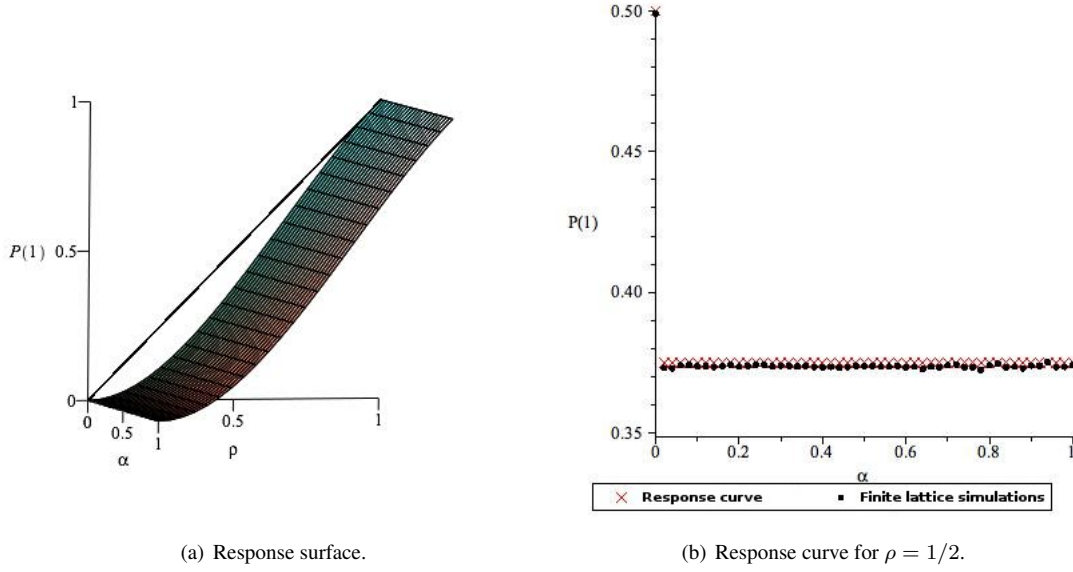(a) Response surface.

(b) Response curve for $\rho = 1/2$.

Fig. 1: Rule 200A - Graphs

Therefore, the asymptotic density of 1's is given by

$$P(1) = \lim_{n \to \infty} P_n(1) = \begin{cases} \rho & \text{if } \alpha = 0 \\ \rho^2(2 - \rho) & \text{if } \alpha \in (0, 1]. \end{cases}$$

Figure 1(a) shows the graph of $P(1)$ as a function of ρ and α.

When $\rho = 1/2$, the response curve is given by

$$P_n^{(s)}(1) = \frac{3}{8} + \frac{1}{8}\beta^n. \tag{20}$$

This response curve is plotted in Figure 1(b). together with results of computer simulations in which we measured density in an array of length 20000, iterated $10^5/\alpha$ times with $\alpha > 0.1$, assuming periodic boundary conditions, averaged over 100 runs. One can see that the response curve is remarkably close to the simulations curve for a finite lattice.

Basic Blocks For Rule 200A, we were also able to find explicit formulae for probabilities of each of the eight blocks in $\mathcal{A}^3$, to be called *basic blocks*. We once again use eq. (14). We omit tedious details of these calculations, which are very similar to what has been presented above for $P_n(1)$. We only present a summary of these findings in Table 2, where the set of all n-step preimage blocks of each basic block is shown, together with corresponding initial probabilities and respective transition probabilities. These results can be used to find formulae for probabilities of basic blocks, such as, for example,

$$P_n(000) = \rho^2(1 - \rho)^2(1 + \rho) + \rho(1 - \rho)^2(1 - 2\rho^2)\beta^n - \rho^2(1 - \rho)^3\beta^{2n}.$$

Tab. 2: Rule 200A - Initial and Transition Probabilities of Basic Blocks

$b \in \mathcal{A}^3$	$b^* \in \operatorname{supp} \boldsymbol{\omega}^n(b\vert\cdot)$	$\boldsymbol{\omega}^n(b\vert b^*)$	$P_0(b^*)$
000	$\star\cdots\star\,0\,1\,0\,1\,0\,\star\cdots\star$	$1 - 2\beta^n + \beta^{2n}$	$\rho^2(1-\rho)^3$
	$\star\cdots\star\star\,0\,0\,1\,0\,\star\cdots\star$	$1 - \beta^n$	$\rho(1-\rho)^3$
	$\star\cdots\star\star\,0\,1\,0\,\star\star\cdots\star$	$1 - \beta^n$	$\rho(1-\rho)^2$
	$\star\cdots\star\,0\,1\,0\,0\,\star\star\cdots\star$	$1 - \beta^n$	$\rho(1-\rho)^3$
	$\star\cdots\star\star\,0\,0\,0\,\star\star\cdots\star$	1	$(1-\rho)^3$
001	$\star\cdots\star\star\,0\,0\,1\,1\,\star\cdots\star$	1	$\rho^2(1-\rho)^2$
	$\star\cdots\star\,0\,1\,0\,1\,1\,\star\cdots\star$	$1 - \beta^n$	$\rho^3(1-\rho)^2$
	$\star\cdots\star\star\,0\,0\,1\,0\,\star\cdots\star$	β^n	$\rho(1-\rho)^3$
	$\star\cdots\star\,0\,1\,0\,1\,0\,\star\cdots\star$	$\beta^n - \beta^{2n}$	$\rho^2(1-\rho)^3$
010	$\star\cdots\star\star\,0\,1\,0\,\star\star\cdots\star$	β^n	$\rho(1-\rho)^2$
011	$\star\cdots\star\star\,0\,1\,1\,\star\star\cdots\star$	1	$\rho^2(1-\rho)$
100	$\star\cdots\star\,1\,1\,0\,0\,\star\star\cdots\star$	1	$\rho^2(1-\rho)^2$
	$\star\cdots\star\,1\,1\,0\,1\,0\,\star\cdots\star$	$1 - \beta^n$	$\rho^3(1-\rho)^2$
	$\star\cdots\star\,0\,1\,0\,0\,\star\star\cdots\star$	β^n	$\rho(1-\rho)^3$
	$\star\cdots\star\,0\,1\,0\,1\,0\,\star\cdots\star$	$\beta^n - \beta^{2n}$	$\rho^2(1-\rho)^3$
101	$\star\cdots\star\,0\,1\,0\,1\,0\,\star\cdots\star$	β^{2n}	$\rho^2(1-\rho)^3$
	$\star\cdots\star\,1\,1\,0\,1\,0\,\star\cdots\star$	β^n	$\rho^3(1-\rho)^2$
	$\star\cdots\star\,0\,1\,0\,1\,1\,\star\cdots\star$	β^n	$\rho^3(1-\rho)^2$
	$\star\cdots\star\,1\,1\,0\,1\,1\,\star\cdots\star$	1	$\rho^4(1-\rho)$
110	$\star\cdots\star\star\,0\,1\,1\,\star\star\cdots\star$	1	$\rho^2(1-\rho)$
111	$\star\cdots\star\star\,1\,1\,1\,\star\star\cdots\star$	1	ρ^3

We show below probabilities of all eight basic blocks in the special case when $\rho = 1/2$, together with asymptotic probabilities, assuming $\alpha \neq 0$.

$$P_n^{(s)}(000) = \frac{13}{32} - \frac{5}{16}\beta^n + \frac{1}{32}\beta^{2n}, \qquad P^{(s)}(000) = 13/32,$$

$$P_n^{(s)}(001) = \frac{3}{32} + \frac{1}{16}\beta^n - \frac{1}{32}\beta^{2n}, \qquad P^{(s)}(001) = 3/32,$$

$$P_n^{(s)}(010) = \frac{1}{8}\beta^n, \qquad P^{(s)}(010) = 0,$$

$$P_n^{(s)}(011) = \frac{1}{8}, \qquad P^{(s)}(011) = 1/8,$$

$$P_n^{(s)}(100) = \frac{3}{32} + \frac{1}{16}\beta^n - \frac{1}{32}\beta^{2n}, \qquad P^{(s)}(100) = 3/32,$$

$$P_n^{(s)}(101) = \frac{1}{32} + \frac{1}{16}\beta^n + \frac{1}{32}\beta^{2n}, \qquad P^{(s)}(101) = 1/32,$$

$$P_n^{(s)}(110) = \frac{1}{8}, \qquad P^{(s)}(110) = 1/8,$$

$$P_n^{(s)}(111) = \frac{1}{8}, \qquad P^{(s)}(111) = 1/8.$$

5 Rule 140A

The next rule to be considered has transition probabilities defined as

$$\omega(1|b) = \begin{cases} 0 & \text{if } b \in \{000, 001, 100, 101\}, \\ 1 & \text{if } b \in \{010, 011, 111\}, \\ 1 - \alpha & \text{if } b = 110, \end{cases} \tag{21}$$

and $\omega(0|b) = 1 - \omega(1|b)$ for all $b \in \mathcal{A}^3$. Note that if $\alpha = 1$, then this rule is equivalent to deterministic Rule 140.

We first find the set of all preimage blocks and their respective transition probabilities.

Proposition 5.1 *The set* $\operatorname{supp} \boldsymbol{\omega}^n(1|\cdot)$ *consists of all blocks of the form*

$$\{ \underbrace{\star \cdots \star}_{n} 1 \underbrace{\star \cdots \star}_{n} \}. \tag{22}$$

Proof: From eq. (21), we can see that a site in state 0 will always remain in state 0, so that for any block $b' \in \mathcal{A}^{2n+1} \setminus \operatorname{supp} \boldsymbol{\omega}^n(1|\cdot)$, we have $\boldsymbol{\omega}^n(1|b') = 0$. A block in $\operatorname{supp} \boldsymbol{\omega}^n(1|\cdot)$, however, could produce a single 1 with some non-zero probability. $\qquad \square$

To determine transition probabilities, we divide the set of preimage blocks into subsets. We start by defining $C_n^k \subset \operatorname{supp} \boldsymbol{\omega}^n(1|\cdot)$ to be the set of blocks of the form

$$\{ \underbrace{\star \cdots \star}_{n-1} 1 \quad \underline{1} \quad \underbrace{1 \cdots 1}_{k-1} 0 \underbrace{\star \cdots \star}_{n-k} \},$$

where $0 \leq k - 1 \leq n$. The value of $k - 1$ indicates the number of 1's before the first potential occurrence of 0 is located, counted to the right of the underlined central 1. Note that if $k - 1 = n$ then the block may not contain any 0's to the right of the centre. We also define the set

$$C_n = \bigcup_{k=0}^{n+1} C_n^k = \{ \underbrace{\star \cdots \star}_{n-1} 1\underline{1} \underbrace{\star \cdots \star}_{n} \},$$

and note that the complement of C_n is given by

$$\operatorname{supp} \boldsymbol{\omega}^n(1|\cdot) \setminus C_n = \{ \underbrace{\star \cdots \star}_{n-1} 0\underline{1} \underbrace{\star \cdots \star}_{n} \}.$$

Proposition 5.2 *For any block $c^* \in \operatorname{supp} \boldsymbol{\omega}^n(1|\cdot) \setminus C_n$, we have $\boldsymbol{\omega}^n(1|c^*) = 1$.*

Proof: From eq. (21), the centre block $0\underline{1}$ will be preserved for the first $(n-1)$-steps with probability 1. Finally, any block $0\underline{1}\star$ will be transformed to a single 1 with probability 1. □

Proposition 5.3 *For any block $c \in C_n^k$, we have*

$$\boldsymbol{\omega}^n(1|c) = \begin{cases} \beta^n & \text{if } k = 1, \\ \beta^n \left(\frac{\alpha}{\beta}\right)^{k-1} \binom{n-1}{k-1} + \beta^{n-k+1} \sum_{j=0}^{k-2} \binom{n-k+j}{j}\alpha^j & \text{if } 2 \leq k \leq n, \\ 1 & \text{if } k = n+1. \end{cases}$$

Proof: To simplify calculations let us use the notation $\gamma_n^k = \boldsymbol{\omega}^n(1|c)$. To calculate this transition probability, we will first write a formula for n-step transition probability recursively in terms of possible $(n-1)$-step transition probabilities. We do so by considering specific cases of the value of k.

1. When $k = 1$, consider the following transition:

$$
\begin{array}{ccccccccccc}
\star & \cdot & \cdot & \star & 1 & 1 & 0 & \star & \cdot & \cdot & \star \\
? & \cdot & ? & & 1 & 1 & 0 & ? & \cdot & ? &
\end{array}
$$

   The shaded transition will occur with probability β.

2. When $2 \leq k \leq n$, consider the following transition:

$$
\begin{array}{ccccccccccccccc}
\star & \cdot & \cdot & \star & 1 & 1 & 1 & \cdot & \cdot & 1 & 1 & 0 & \star & \cdot & \cdot & \star \\
? & \cdot & ? & & 1 & 1 & 1 & \cdot & \cdot & 1 & x & 0 & ? & \cdot & ? &
\end{array}
$$

   We know that $x = 1$ with probability β, resulting in a block in C_{n-1}^k, and $x = 0$ with probability α, resulting in a block in C_{n-1}^{k-1}.

3. When $k = n + 1$, consider the following transition:

$$
\begin{array}{ccccccccccc}
\star & \cdot & \cdot & \star & 0 & 1 & 1 & \cdot & \cdot & \cdot & 1 \\
? & \cdot & ? & & 0 & 1 & 1 & \cdot & \cdot & 1 &
\end{array},
$$

   which will occur with probability 1.

Combining these cases, we obtain the following recursive formula

$$\gamma_n^k = \begin{cases} \beta\gamma_{n-1}^1 & \text{if } k = 1, \\ \alpha\gamma_{n-1}^{k-1} + \beta\gamma_{n-1}^k & \text{if } 2 \le k \le n, \\ 1 & \text{if } k = n + 1. \end{cases} \tag{23}$$

This recursive equation can be solved to give the desired result. To see this, consider first the case of $k = 1$ or $k = n + 1$, when our formula follows trivially from eq. (23). When $2 \le k \le n$, our formula can be proved by induction with respect to n. When $n = 2$, we only have the case of $k = 2$, where

$$\gamma_2^2 = \beta^1\alpha^1 \binom{1}{1} + \beta^1 \sum_{j=0}^{0} \binom{j}{j}\alpha^j = \beta\alpha + \beta = 1 - \alpha^2.$$

Now, we consider the following inductive step, for $3 \le k \le n$,

$$\gamma_n^k = \alpha\gamma_{n-1}^{k-1} + \beta\gamma_{n-1}^k$$

$$= \alpha \left[\beta^{n-1} \left(\frac{\alpha}{\beta}\right)^{k-2} \binom{n-2}{k-2} + \beta^{n-k+1} \sum_{j=0}^{k-3} \binom{n-k+j}{j}\alpha^j \right] +$$

$$+ \beta \left[\beta^{n-1} \left(\frac{\alpha}{\beta}\right)^{k-1} \binom{n-2}{k-1} + \beta^{n-k} \sum_{j=0}^{k-2} \binom{n-k+j-1}{j}\alpha^j \right]$$

$$= \alpha\beta^{n-1} \left(\frac{\alpha}{\beta}\right)^{k-2} \binom{n-2}{k-2} + \beta\beta^{n-1} \left(\frac{\alpha}{\beta}\right)^{k-1} \binom{n-2}{k-1}$$

$$+ \alpha\beta^{n-k+1} \sum_{j=0}^{k-3} \binom{n-k+j}{j}\alpha^j + \beta\beta^{n-k} \sum_{j=0}^{k-2} \binom{n-k+j-1}{j}\alpha^j$$

$$= \beta^n \left(\frac{\alpha}{\beta}\right)^{k-1} \binom{n-1}{k-1} + \beta^{n-k+1} \left[\sum_{j=1}^{k-2} \binom{n-k+j-1}{j-1}\alpha^j + \beta + \beta \sum_{j=1}^{k-2} \binom{n-k+j-1}{j}\alpha^j \right]$$

$$= \beta^n \left(\frac{\alpha}{\beta}\right)^{k-1} \binom{n-1}{k-1} + \beta^{n-k+1} \sum_{j=0}^{k-2} \binom{n-k+j}{j}\alpha^j.$$

A similar procedure can be used to prove the formula when $k = 2$, thus completing the proof. $\qquad\square$

We may now use eq. (14) and consider the sets and transition probabilities described in Propositions 5.2 and 5.3, concluding that

$$P_n(1) = \sum_{c^* \in \text{supp}\, \boldsymbol{\omega}^n(1|\cdot) \backslash C_n} P_0(c^*)\boldsymbol{\omega}^n(1|c^*) + \sum_{c^* \in C_n} P_0(c^*)\boldsymbol{\omega}^n(1|c^*)$$

$$= \rho(1 - \rho) + \rho^2(1 - \rho)\beta^n + \rho^2(1 - \rho) \sum_{k=2}^{n} \rho^{k-1}\gamma_n^k + \rho^{n+2}, \tag{24}$$

where

$$\sum_{k=2}^{n} \rho^{k-1}\gamma_n^k = \beta^n \sum_{k=2}^{n} \binom{n-1}{k-1} \left(\frac{\rho\alpha}{\beta}\right)^{k-1} + \sum_{k=2}^{n}\sum_{j=0}^{k-2} \binom{n-k+j}{j} \alpha^j \left(\frac{\rho}{\beta}\right)^{k-1}. \tag{25}$$

Further simplification of eq. (24) and (25) is possible, using the following two summation identities.

Lemma 5.1

$$\sum_{k=2}^{n} \binom{n-1}{k-1} \left(\frac{\alpha\rho}{\beta}\right)^{k-1} = -1 + \left(1 + \frac{\alpha\rho}{\beta}\right)^{n-1}.$$

Proof: We use the binomial identity as follows

$$-1 + \left(1 + \frac{\alpha\rho}{\beta}\right)^{n-1} = -1 + \sum_{k=0}^{n-1} \binom{n-1}{k} \left(\frac{\alpha\rho}{\beta}\right)^{k} = \sum_{k=2}^{n} \binom{n-1}{k-1} \left(\frac{\alpha\rho}{\beta}\right)^{k-1}.$$
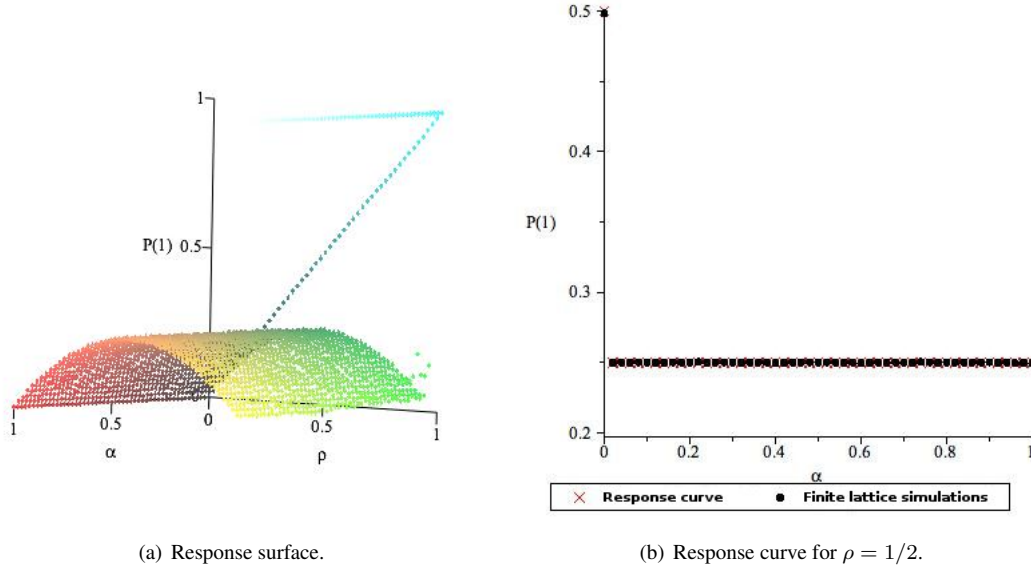
$$\square$$

Lemma 5.2 *When $\rho \neq 1$ and $\alpha \neq 0$, we have*

$$\beta^n \sum_{k=2}^{n}\sum_{j=0}^{k-2} \binom{n-k+j}{j} \alpha^j \left(\frac{\rho}{\beta}\right)^{k-1} = \frac{\rho}{1-\rho} \left[(\beta + \rho\alpha)^{n-1} - \rho^{n-1}\right]. \tag{26}$$

Proof: We prove this identity by induction. When $n = 2$, both sides of the identity equal to $\rho\beta$. If we denote by $h(n)$ the left hand side of eq. (26), then $h(n+1)$ is given by

$$\beta^{n+1} \sum_{k=2}^{n+1}\sum_{j=0}^{k-2} \binom{n+1-k+j}{j} \alpha^j \left(\frac{\rho}{\beta}\right)^{k-1}$$

$$= \rho\beta^n \sum_{m=1}^{n}\sum_{j=0}^{m-1} \binom{n-m+j}{j} \alpha^j \left(\frac{\rho}{\beta}\right)^{m-1} \qquad \text{(where we defined } m = k-1\text{)}$$

$$= \rho\beta^n \left(\sum_{m=2}^{n}\sum_{j=0}^{m-1} \binom{n-m+j}{j} \alpha^j \left(\frac{\rho}{\beta}\right)^{m-1} + 1\right)$$

$$= \rho\beta^n \left(\sum_{m=2}^{n}\sum_{j=0}^{m-2} \binom{n-m+j}{j} \alpha^j \left(\frac{\rho}{\beta}\right)^{m-1} + \sum_{m=2}^{n} \binom{n-1}{m-1} \left(\frac{\alpha\rho}{\beta}\right)^{m-1} + 1\right)$$

$$= \rho\beta^n \sum_{k=2}^{n}\sum_{j=0}^{k-2} \binom{n-k+j}{j} \alpha^j \left(\frac{\rho}{\beta}\right)^{k-1} + \rho\beta^n \left(\sum_{k=2}^{n} \binom{n-1}{k-1} \left(\frac{\alpha\rho}{\beta}\right)^{k-1} + 1\right).$$

(a) Response surface.

(b) Response curve for $\rho = 1/2$.

Fig. 2: Rule 140A - Graphs

Now, using the inductive hypothesis of eq. (26) and Lemma 5.1, we simplify $h(n+1)$ as follows,

$$h(n+1) = \rho\frac{\rho}{1-\rho}\left[(\beta+\rho\alpha)^{n-1} - \rho^{n-1}\right] + \rho\beta^n\left(-1 + \left(1 + \frac{\alpha\rho}{\beta}\right)^{n-1} + 1\right)$$
$$= \frac{\rho}{1-\rho}\left[(\beta+\rho\alpha)^n - \rho^n\right].$$

$\square$

Using Lemmas 5.1 and 5.2, we can now simplify eq. (24) and (25) to give

$$P_n(1) = \rho(1-\rho) + \rho^2\left(1 - (1-\rho)\alpha\right)^n. \tag{27}$$

The asymptotic density, therefore, is given by

$$P(1) = \lim_{n\to\infty} P_n(1) = \begin{cases} 1 & \text{if } \rho = 1, \\ \rho & \text{if } \alpha = 0, \\ \rho(1-\rho) & \text{otherwise.} \end{cases} \tag{28}$$

Figure 2(a) shows the graph of $P(1)$ vs. α and ρ.

In the special case when $\rho = 1/2$, we obtain

$$P_n^{(s)}(1) = \frac{1}{4} + \frac{1}{4}\left(1 - \frac{\alpha}{2}\right)^n. \tag{29}$$

In Figure 2(b), the graph of $P_n^{(s)}(1)$ is shown as a function of α, as given in eq. (29). The same figure shows results of computer simulation of iterations of Rule 140, in which this rule was applied to an array of length 20000, iterated $100000/\alpha$ times for $\alpha > 0.1$ and 1000000 times for $\alpha \le 0.1$, with periodic boundary conditions, and the results were averaged over 100 runs.

Basic Blocks   For Rule 140A, we were also able to find explicit formulae for probabilities each of the eight basic blocks. Once again omitting details, in Table 3 we show the set of n-step preimage blocks for four of the eight basic block, together with corresponding initial probabilities and respective transition probabilities. One can use this table together with consistency conditions for block probabilities to find

Tab. 3: Rule 140A - Initial and Transition Probabilities of Basic Blocks

$b \in \mathcal{A}^3$	$b^* \in \operatorname{supp} \boldsymbol{\omega}^n(b\vert\cdot)$	$\boldsymbol{\omega}^n(b\vert b^*)$	$P_0(b^*)$
001	$\underbrace{\star \cdots \star}_{n} 0\underline{0}1 \underbrace{\star \cdots \star}_{n}$	1	$\rho(1-\rho)^2$
	$\underbrace{\star \cdots \star}_{n-1} 1\ 1\underline{0}1 \underbrace{\star \cdots \star}_{n}$	$1-\beta^n$	$\rho^3(1-\rho)$
011	$\underbrace{\star \cdots \star}_{n} 0\underline{1}1 \underbrace{1 \cdots 1}_{k-1} 0 \underbrace{\star \cdots \star}_{n-k}$ where $1 \le k \le n+1$	see Prop. 5.3	see eq. (27)
101	$\underbrace{\star \cdots \star}_{n} 1\ 1\underline{0}1 \underbrace{\star \cdots \star}_{n}$	β^n	$\rho^3(1-\rho)$
	$\underbrace{\star \cdots \star}_{n-1} 0\ 1\underline{0}1 \underbrace{\star \cdots \star}_{n}$	1	$\rho^2(1-\rho)^2$
111	$\underbrace{\star \cdots \star}_{n} 1\underline{1}1 \underbrace{1 \cdots 1}_{k-1} 0 \underbrace{\star \cdots \star}_{n-k}$ where $1 \le k \le n+1$	see Prop. 5.3	see eq. (27)

formulae for probabilities of all eight basic blocks. We summarize these results as follows, where we assume that $\alpha \ne 0$.

$$P_n^{(s)}(000) = \frac{7}{16} - \frac{1}{4}\left(1-\frac{\alpha}{2}\right)^n + \frac{1}{16}\beta^n, \qquad P^{(s)}(000) = 7/16,$$

$$P_n^{(s)}(001) = \frac{1}{16} + \frac{1}{16}\beta^n, \qquad P^{(s)}(001) = 1/16,$$

$$P_n^{(s)}(010) = \frac{1}{4} - \frac{1}{8}\left(1-\frac{\alpha}{2}\right)^n, \qquad P^{(s)}(010) = 1/4,$$

$$P_n^{(s)}(011) = \frac{1}{8}\left(1-\frac{\alpha}{2}\right)^n, \qquad P^{(s)}(011) = 0,$$

$$P_n^{(s)}(100) = \frac{3}{16} - \frac{1}{16}\beta^n, \qquad P^{(s)}(100) = 3/16,$$

$$P_n^{(s)}(101) = \frac{1}{16} + \frac{1}{16}\beta^n, \qquad\qquad P^{(s)}(101) = 1/16,$$

$$P_n^{(s)}(110) = \frac{1}{8}\left(1 - \frac{\alpha}{2}\right)^n, \qquad\qquad P^{(s)}(110) = 0,$$

$$P_n^{(s)}(111) = \frac{1}{8}\left(1 - \frac{\alpha}{2}\right)^n, \qquad\qquad P^{(s)}(111) = 0.$$

6 Rule 76A

Rule 76 is the most difficult to analyze. Its transition probabilities are defined as

$$\omega(1|b) = \begin{cases} 0 & \text{if } b \in \{000, 001, 100, 101\}, \\ 1 & \text{if } b \in \{010, 011, 110\}, \\ 1 - \alpha & \text{if } b = 111, \end{cases} \tag{30}$$

and $\omega(0|b) = 1 - \omega(1|b)$ for all $b \in \mathcal{A}^3$. If $\alpha = 1$, then this rule is equivalent to deterministic Rule 76.

In this section, we will use the Kroenecker delta function, defined as

$$\delta_{(x,y)} = \begin{cases} 0 & \text{if } x \neq y \\ 1 & \text{if } x = y. \end{cases}$$

We now find the set of all potential preimage blocks and their respective transition probabilities. We start by defining $E_n^{k_1,k_2}$ to be the set of blocks of the form

$$\{\underbrace{\star \cdots \star}_{n-k_1-1} 0 \underbrace{1 \cdots 1}_{k_1} \underline{1} \underbrace{1 \cdots 1}_{k_2} 0 \underbrace{\star \cdots \star}_{n-k_2-1}\},$$

where $1 \leq k_1, k_2 \leq n$. The values of k_1, k_2 refer to the number of 1's to the left and right, respectively, of the centre $\underline{1}$ before the first potential occurence of a 0. Note that if $k_1 = n$ or $k_2 = n$, then the block may not contain any 0's to the left or right of the centre.

Proposition 6.1 *The set* $\operatorname{supp} \omega^n(1|\cdot)$ *consists of all blocks in*

$$E_n = \bigcup_{k_1,k_2=1}^{n} E_n^{k_1,k_2} = \{\underbrace{\star \cdots \star}_{n} 1 \underbrace{\star \cdots \star}_{n}\}.$$

Proof: From eq. (30), we can see that a site in state 0 will always remain in state 0, so that for any block $e' \in \operatorname{supp} \omega^n(1|\cdot) \setminus E_n$, we have $\omega^n(1|e') = 0$. A block in $\operatorname{supp} \omega^n(1|\cdot)$, however, could produce a single 1 with some non-zero probability. □

We were not able to obtain explicit formulae for transition probabilities for this rule, but we were able to find recursive formulae for them.

Proposition 6.2 *For any block belonging to $E_n^{k_1,k_2}$, to be denoted by $e_n^{k_1,k_2}$, we have*

$$\omega^n(1|e_n^{k_1,k_2}) = \begin{cases} \omega^{n-1}(1|e_{n-1}^{0,0}) & \text{if } k_1 = 0, k_2 = 0, \\[2ex] \displaystyle\sum_{i=0}^{k_2-1} \alpha^{1-\delta_{(i,k_2-1)}} \beta^i \omega^{n-1}(1|e_{n-1}^{0,i'}) & \text{if } k_1 = 0, 1 \leq k_2 \leq n, \\[3ex] \displaystyle\sum_{j=0}^{k_1-1} \alpha^{1-\delta_{(j,k_1-1)}} \beta^j \omega^{n-1}(1|e_{n-1}^{j',0}) & \text{if } 1 \leq k_1 \leq n, k_2 = 0, \\[3ex] \displaystyle\sum_{j=0}^{k_1-1}\sum_{i=0}^{k_2-1} \alpha^{2-\delta_{(j,k_1-1)}-\delta_{(i,k_2-1)}} \beta^{j+i+1} \omega^{n-1}(1|e_{n-1}^{j',i'}) & \text{if } 1 \leq k_1, k_2 \leq n, \end{cases}$$

where $j' = j + \delta_{(j,k_1-1)} - \delta_{(j,n-1)}$, $i' = i + \delta_{(i,k_2-1)} - \delta_{(i,n-1)}$, $\varepsilon_1^{1,1} = \varepsilon_1^{1,2} = \varepsilon_1^{2,1} = 1$, and $\varepsilon_1^{2,2} = \beta$.

Proof:

The proof of this proposition is rather long and tedious. It is similar in structure to the derivation of eq. (23). Since we were unable to derive a closed-form expression for the sums contained in the above transition probability, we omit the details of the derivation here. The full proof is available upon request. □

If we consider eq. (14) and Proposition 6.2, we conclude that

$$P_n(1) = \sum_{j=0}^{n}\sum_{i=0}^{n} \rho^{j+i+1}(1-\rho)^{2-\delta_{(j,n)}-\delta_{(i,n)}} \omega^n(1|e_n^{j,i}). \tag{31}$$

The response curve (Figure 3(a)) is plotted using eq. (31) for $n = 15/\alpha$ for $\alpha > 0.1$ and $n = 150$ when $\alpha \leq 0.1$.

The symmetric response curve is given by

$$P_n^{(s)}(1) = \sum_{j=0}^{n}\sum_{i=0}^{n} 2^{-i-j-3+\delta_{(j,n)}+\delta_{(i,n)}} \omega^n(1|e_n^{j,i}). \tag{32}$$

In Figure 3(b), $P_n^{(s)}(1)$, given by eq. (32) is plotted with together with results of directly simulated iterations of Rule 76. For the theoretical plot, we used $n = 15/\alpha$ for $\alpha > 0.1$ and $n = 150$ when $\alpha \leq 0.1$. For the simulated plot, an array of length 20000 was iterated $100000/\alpha$ times with $\alpha > 0.1$ and 1000000 times with $\alpha \leq 0.1$, with periodic boundary conditions, averaged over 100 runs. We can see that as before, there is a close agreement between the theoretical and experimental results.

7   Conclusion

We have demonstrated that for single-transition asynchronous rules it is possible to find explicit expressions for probabilities of 1 after n iterations of the rule, starting from the Bernoulli measure. In two cases these expressions are explicit, in the third case we found a recursive formula. Furthermore, for rules 200A

(a) Response surface.

(b) Response curve for $\rho = 1/2$.
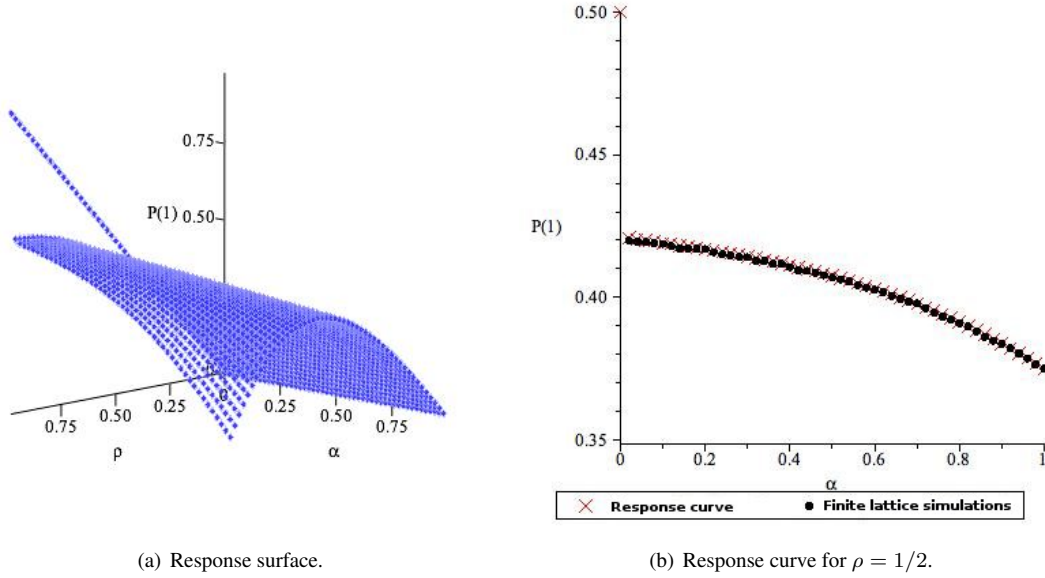
Fig. 3: Rule 76A - Graphs.

and 140A, one can also compute probabilities of blocks of length 3 (thus, by using consistency conditions, also of length 2). These results provide partial characterization of the orbit of Bernoulli measure under the action of single-transition asynchronous rules.

We hope that these results are useful in future research on probabilistic rules, in the following context. There exist various methods for computing approximate orbits of measures in CA and related system, such as, for example, mean-field approximation and its generalization, called a local structure theory (Gutowitz et al., 1987). The quality of these approximations is often judged by comparison of their predictions with computer experiments. This is not entirely satisfactory for a number of reasons, among them the fact that simulations are only possible for finite systems. Having some benchmark cases for which exact solutions are known, such as those presented here, will help to evaluate quality of approximate methods in a more rigorous fashion. Work in this direction is currently in progress.

Acknowledgements

References

V. Belitsky and P. A. Ferrari. Invariant measures and convergence properties for cellular automaton 184 and related processes. *Journal of Statistical Physics*, 118:589–623, 2005.

M. Blank. Ergodic properties of a simple deterministic traffic flow model. *Journal of Statistical Physics*, 111:903–930, 2003.

N. Fatès and M. Morvan. An experimental study of robustness to asynchronism for elementary cellular automata. *Complex Systems*, 16:1–27, 2005.

N. Fatès, D. Regnault, N. Schabanel, and É. Thierry. Asynchronous behavior of double-quiescent elementary cellular automata. In J. Correa, A. A. Hevia, and M. Kiwi, editors, *LATIN 2006: Theoretical Informatics*, volume 3887 of *LNCS*, pages 455–466, 2006.

E. Formenti and P. Kůrka. Dynamics of cellular automata in non-compact spaces. In R. A. Meyers, editor, *Encyclopedia of Complexity and System Science*. Springer, 2009.

H. Fukś. Exact results for deterministic cellular automata traffic models. *Phys. Rev. E*, 60:197–202, 1999.

H. Fukś. Dynamics of the cellular automaton rule 142. *Complex Systems*, 16:123–138, 2006.

H. Fukś. Probabilistic initial value problem for cellular automaton rule 172. *DMTCS proc.*, AL:31–44, 2010.

H. A. Gutowitz, J. D. Victor, and B. W. Knight. Local structure theory for cellular automata. *Physica D*, 28:18–48, 1987.

P. Kůrka. On the measure attractor of a cellular automaton. *Discrete and Continuous Dynamical Systems*, pages 524 – 535, 2005.

P. Kůrka. Topological dynamics of cellular automata. In R. A. Meyers, editor, *Encyclopedia of Complexity and System Science*. Springer, 2009.

P. Kůrka and A. Maass. Limit sets of cellular automata associated to probability measures. *Journal of Statistical Physics*, 100:1031–1047, 2000.

M. Pivato. Conservation laws in cellular automata. *Nonlinearity*, 15(6):1781, 2002.

M. Pivato. Ergodic theory of cellular automata. In R. A. Meyers, editor, *Encyclopedia of Complexity and System Science*. Springer, 2009.

Conservation Laws and Invariant Measures in Surjective Cellular Automata

Jarkko Kari[1][†] and Siamak Taati[2][‡]

[1]*Department of Mathematics, University of Turku, Finland*
[2]*Department of Mathematics, University of Groningen, the Netherlands*

We discuss a close link between two seemingly different topics studied in the cellular automata literature: additive conservation laws and invariant probability measures. We provide an elementary proof of a simple correspondence between invariant full-support Bernoulli measures and interaction-free conserved quantities in the case of one-dimensional surjective cellular automata. We also discuss a generalization of this fact to Markov measures and higher-range conservation laws in arbitrary dimension. As a corollary, we show that the uniform Bernoulli measure is the only shift-invariant, full-support Markov measure that is invariant under a strongly transitive cellular automaton.

Keywords: surjective cellular automata, conservation laws, invariant measures, statistical equilibrium

1 Introduction

Let $\Phi : S^{\mathbb{Z}} \longrightarrow S^{\mathbb{Z}}$ be a one-dimensional reversible cellular automaton, and let $\mu : S \longrightarrow \mathbb{R}$ be a quantity that is conserved by the evolution of Φ, in the sense that the average value of μ over any periodic configuration remains constant with time. Since adding a constant to μ does not change the latter condition, we may assume that μ is normalized in such a way that $\sum_{s \in S} 2^{-\mu(s)} = 1$.

Suppose that the cells are initialized randomly and independently so that each state $s \in S$ appears with probability $p(s) \triangleq 2^{-\mu(s)}$. In particular, if $w = w_1 w_2 \cdots w_n$ is a word of length n over S, the probability that n consecutive cells $i + 1, i + 2, \ldots, i + n$ take, respectively, the states $w_1, w_2, \ldots, w_n$ is $p(w) \triangleq p(w_1)p(w_2) \cdots p(w_n)$. There is a simple argument showing that p is a stationary distribution for Φ; that is, after any number of iterations of Φ, the state of the cells remain independent and with the same distribution p.

Namely, let u be a word of length l over S. If $\varphi : S^k \longrightarrow S$ is the local update rule of Φ, then there are a finite number of words v_i of length $k+l$ such that $\varphi(v_i) = u$. Let c_i be the periodic configuration obtained by repeating v_i on positions $\ldots, -(l + k), 0, l + k, \ldots$. The image of c_i is a periodic configuration e_i that is a repetition of a word of the form $x_i\, u\, y_i$ on positions $\ldots, -(l + k), 0, l + k, \ldots$. Conservation of μ implies that

$$\mu(v_i) = \mu(x_i) + \mu(u) + \mu(y_i) . \tag{1}$$
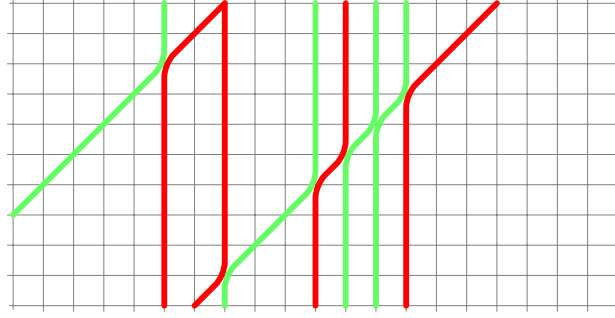
Fig. 1: Particles moving on the discrete line. Time goes downwards.

Raising 2 to the power of minus this value and summing over all i we have

$$\sum_i p(v_i) = \left[\sum_i p(x_i)\, p(y_i)\right] p(u) \,. \tag{2}$$

But since Φ is bijective, as i varies, the combination $y_i\, x_i$ takes all the possible values in S^k, each precisely once. Therefore, the summation part of the right-hand side adds up to 1, and we obtain

$$\sum_i p(v_i) = p(u) \,. \tag{3}$$

This is true for any finite word u, which means the joint probability distribution of the cell states remains unchanged under iterations of Φ.

As an example, consider the following discrete mechanical system in one dimension, consisting of particles moving on the discrete line $\mathbb{Z}$ and interacting with each other. Each particle is either standing still, or moving to the left, with constant speed 1. Each site may contain up to two particles, one standing and one moving. Upon collision (i.e., when a moving particle meets a standing one) the moving particle stops and the standing particle starts moving. To make the model more interesting, let us assume that the particles are of two distinguishable types — red and green (Figure 1). Obviously, the total number of red or green particles is conserved with time. Furthermore, the total number of moving particles never changes. It follows from the above argument that if we choose the number and color of particles in each cell independently, according to any fixed non-vanishing probability distribution p, then in any future instant of time, the number and color of the particles at different sites remain independent and with the same distribution p.

The above observation states that for any (interaction-free) additive conserved quantity in a reversible cellular automaton, there corresponds a Bernoulli distribution that is stationary. The converse is also true: every stationary Bernoulli distribution corresponds to an additive conserved quantity. More specifically, suppose that the Bernoulli distribution with cell marginal distribution $p : S \longrightarrow (0, 1]$ is stationary for Φ. The claim is that the quantity $\mu(s) \triangleq -\log p(s)$ is conserved by Φ.

The argument is quite similar to the previous one. Let c_v be the periodic configuration obtained by repeating a word v of length l at positions $\dots, -l, 0, l, \dots$. Let $c_u = \Phi(c_v)$ be the image of c_v, which is

again a repetition of a word u of length l at positions $\ldots, -l, 0, l, \ldots$. For an integer $n > 0$, consider the word u^n. Since Φ is injective, the pre-images of u^n under φ are words of the form $x_i \, v^{n-2} \, y_i$. (That is, each of the pre-images has several copies of v in the middle and constant-size turbulence on the borders. We have assumed that l is sufficiently large.) Since p is stationary, we have

$$p(u^n) = \sum_i p(x_i \, v^{n-2} \, y_i) \;, \tag{4}$$

or

$$p(u)^n = \left[\sum_i p(x_i) \, p(y_i) \right] p(v)^{n-2} \;. \tag{5}$$

Taking the logarithm on both sides, dividing by n, and letting $n \longrightarrow \infty$ we obtain that

$$\mu(u) = \mu(v) \;. \tag{6}$$

This is the case, for any periodic configurations c_v and $c_u = \Phi(c_v)$, which means that μ is conserved by Φ.

It turns out that the above correspondence between conserved quantities and stationary Bernoulli distributions generalizes to any number of dimensions. The cellular automaton is merely required to be surjective. Finally, the conserved quantity may involve contributions from the interactions between nearby cells, in which case the corresponding probability distribution becomes a Markov measure.

This property can be proved using the variational principle of equilibrium statistical mechanics and the properties of the pre-injective factor maps on strongly irreducible shifts of finite type. It can be seen as a generalization of the balance property of the surjective cellular automata.

In the present paper we give an elementary proof of the correspondence between conserved quantities and stationary Bernoulli distributions in surjective one-dimensional cellular automata. We also state the general theorem, but the complete proof will appear elsewhere [KT]. In Section 5, we provide an example of how this theorem allows us to transmit results concerning conservation laws over to invariant measures.

2 Preliminaries

2.1 One-dimensional cellular automata

Let S be a finite set of *states*. A one-dimensional *cellular automaton* (CA) over S is a translation commuting continuous map $\Phi : S^{\mathbb{Z}} \longrightarrow S^{\mathbb{Z}}$. Equivalently, according to the Curtis-Hedlund-Lyndon theorem [Hed69], Φ is defined by a parallel application of a *local rule* $f : S^{2r+1} \longrightarrow S$ at all sites, where r is the neighborhood *radius* of Φ: for every $c \in S^{\mathbb{Z}}$ and every $i \in \mathbb{Z}$,

$$\Phi(c)_i = f(c_{i-r}, c_{i-r+1}, \ldots, c_{i+r}).$$

Elements of $S^{\mathbb{Z}}$ are the *configurations* of the CA. For any word $u \in S^l$ of length $l \geq 1$, we denote by $^\omega u^\omega$ the *periodic configuration* in which the word u is repeated, starting in positions $\ldots, -l, 0, l, 2l, \ldots$.

A *cylinder* determined by word u and position $i \in \mathbb{Z}$ is the set

$$[u]_i \triangleq \{c \in S^{\mathbb{Z}} \mid c_i c_{i+1} \ldots c_{i+|u|-1} = u\}$$

of configurations that contain word u, starting in position i. Cylinders form a basis of the standard topology we use on $S^{\mathbb{Z}}$. Many concepts we consider are indifferent to the exact position i of the cylinder, and in those cases we use the simpler notation $[u]$. This can be interpreted as any $[u]_i$; for example as $[u]_0$.

A cellular automaton Φ is *injective*, *surjective* or *bijective* if it is one-to-one, onto or a bijection, respectively, as a function $S^{\mathbb{Z}} \longrightarrow S^{\mathbb{Z}}$. A cellular automaton Φ is called *reversible* if it is bijective and its inverse is a cellular automaton. It follows from the compactness of $S^{\mathbb{Z}}$ that every bijective CA is, in fact, reversible.

Two configurations $c, e \in S^{\mathbb{Z}}$ are *asymptotic* if the difference set $\{i \in \mathbb{Z} \mid c_i \neq e_i\}$ is finite. A CA Φ is called *pre-injective* if for any asymptotic configurations c, e holds that $c \neq e \implies \Phi(c) \neq \Phi(e)$. The celebrated Garden-of-Eden theorem by E. F. Moore and J. Myhill states that Φ is surjective if and only if it is pre-injective [Moo62, Myh63]. It then directly follows that every injective CA is also surjective. We see that injectivity, bijectivity and reversibility are equivalent concepts on cellular automata, and they imply surjectivity which is equivalent to pre-injectivity.

Let Φ be defined by a radius-r local rule. The local rule can be applied on finite words in a natural way, so that it defines functions $S^{k+2r} \longrightarrow S^k$ for every k. We use the same symbol Φ also for these functions on words.

All surjective CA have the following *balance property* [Hed69]. Every word u of length k has precisely $N \triangleq |S|^{2r}$ pre-images $v_1, v_2 \ldots, v_N \in S^{k+2r}$ under the function $\Phi : S^{k+2r} \longrightarrow S^k$. In terms of cylinders this means that the pre-image of every cylinder $[u]_i$ determined by $u \in S^k$ is the disjoint union of the N cylinders $[v_1]_{i-r}, [v_2]_{i-r}, \ldots, [v_N]_{i-r}$ determined by words v_j of length $k + 2r$.

2.2   Conserved quantities

Let $\mu : S \longrightarrow \mathbb{R}$ be an assignment of real numbers to the states. For a word $v \in S^*$ we define $\mu(v) \triangleq \mu(v_1) + \mu(v_2) + \ldots + \mu(v_k)$, where $v = v_1 v_2 \ldots v_k$ and $v_i \in S$. Function μ is an (interaction-free) *additive quantity*.

A cellular automaton $\Phi : S^{\mathbb{Z}} \longrightarrow S^{\mathbb{Z}}$ *conserves* μ if for every $u, v \in S^k$ such that $\Phi(^{\omega}u^{\omega}) = {}^{\omega}v^{\omega}$ holds

$$\mu(u) = \mu(v).$$

In other words, we require the (well defined) average value over periodic configurations to remain unchanged under the application of Φ. Other equivalent characterizations exist (see e.g. [HT91, Piv02, DFR03, MBG04]). In particular, there are simple algorithms for verifying whether a given cellular automaton conserves a given additive quantity. For the proofs we present here, the above characterization using periodic configurations is most convenient.

Conserved quantities play an important role in physics, and the concept has been studied in the context of cellular automata by several authors (see e.g. [HT91, BF98, Piv02, DFR03, FG03, MBG04, Ber07, FKTar]).

2.3   Invariant Bernoulli measures

Let $p : S \longrightarrow \mathbb{R}$ be a probability distribution on the state set, so that $\sum_{s \in S} p(s) = 1$. The *Bernoulli distribution* determined by p is the probability distribution of a random configuration $c \in S^{\mathbb{Z}}$ if the values c_i, for $i \in \mathbb{Z}$, are chosen randomly and independently, each with distribution p. It is identified by a Borel probability measure π that assigns probability $\pi([v]) \triangleq p(v_1)p(v_2)\ldots p(v_k)$ to each cylinder $[v]$,

where $v = v_1 v_2 \cdots v_k$ and $v_i \in S$. The measure π is translation-invariant. We only consider the case where $p(s) > 0$ for all $s \in S$, which is equivalent to requiring that the Bernoulli measure π is full-support.

A Borel measure π is *invariant* (or *stationary*) under a cellular automaton Φ if for all cylinders $[u]$ holds $\pi([u]) = \pi(\Phi^{-1}([u]))$. This means that π is a fixed point of the mapping $\eta \mapsto \Phi(\eta)$ where $\Phi(\eta)$ is the distribution of the configuration $\Phi(c)$ if configuration c is chosen randomly according to η.

The balance property of surjective CA can now be rephrased as follows: the uniform Bernoulli measure is invariant under all surjective CA.

The presence of natural invariant measures allows one to study CA as measure-preserving dynamical systems, applying results from ergodic theory (see e.g. [Wal82, Lin84, Piv09]). For instance, knowing the invariance of the uniform Bernoulli measure, one can apply Poincaré's recurrence theorem to infer that iterating a surjective CA on a uniformly random initial configuration, almost surely each finite word appearing in the initial configuration reappears infinitely many times on the same position.

3 A correspondence of conserved quantities and invariant measures on one-dimensional surjective CA

In this section, we state and give an elementary proof for a correspondence between (interaction-free) conserved quantities and invariant, full-support Bernoulli measures on surjective, one-dimensional CA. The balance property is a special case of our theorem, so it is not surprising that the proof is similar to a standard proof of the balance property. The second part of the proof was essentially presented in [Taa09] in a more general set-up.

Theorem 1 *Let $\Phi : S^{\mathbb{Z}} \longrightarrow S^{\mathbb{Z}}$ be a one-dimensional surjective cellular automaton over state set $S = \{s_1, s_2, \ldots, s_n\}$, and let $p_1, p_2, \ldots, p_n$ be n positive numbers such that $p_1 + p_2 + \ldots + p_n = 1$. Let π be the Bernoulli distribution on $S^{\mathbb{Z}}$ defined by $\pi(s_i) \triangleq p_i$ for all i, and let μ be the additive quantity defined by $\mu(s_i) \triangleq -\log p_i$. Then, π is invariant under Φ if and only if Φ conserves μ.*

Proof: The base of the logarithm does not matter — we use base 2 in the proof. Let Φ be defined by a radius-r local rule. Recall that Φ denotes both the CA function and the word functions $S^{k+2r} \longrightarrow S^k$. Observe also the correspondence

$$\pi([v]) = 2^{-\mu(v)}$$

for all words $v \in S^*$.

($\Longleftarrow$) Assume that Φ conserves the quantity μ. Let us first prove that there are positive constants m and M, independent of k, such that for all words $v \in S^{k+2r}$ and for $u \triangleq \Phi(v) \in S^k$ holds
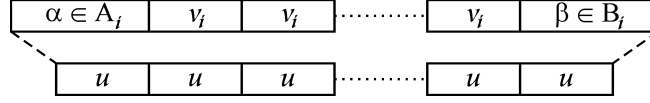
$$m \leq \pi([v])/\pi([u]) \leq M. \tag{7}$$

Indeed, let $x, y \in S^r$ be the words such that $\Phi(^{\omega}v^{\omega}) = {}^{\omega}(xuy)^{\omega}$. Then, due to the conservation of μ by Φ, we have that $\mu(v) - \mu(u) = \mu(xy)$. Hence

$$c \leq \mu(v) - \mu(u) \leq C \tag{8}$$

for constants $c \triangleq 2r \cdot \min\{\mu(s) \mid s \in S\}$ and $C \triangleq 2r \cdot \max\{\mu(s) \mid s \in S\}$. Raising two to the negative powers of the different sides of (8) gives

$$2^{-c} \geq \pi([v])/\pi([u]) \geq 2^{-C},$$

Fig. 2: The structure of the pre-images of word u^t.

which proves (7).

We can now prove that π is Φ-invariant. Suppose the contrary. Then there exists a word $u \in S^k$ such that $\pi(\Phi^{-1}([u])) < \pi([u])$. (Indeed, if instead we have $\pi(\Phi^{-1}([u])) > \pi([u])$ for some $u \in S^k$, then there necessarily is another element in S^k that has the required property.) Let $a \triangleq \pi(\Phi^{-1}([u]))/\pi([u]) < 1$.

The word u has $N \triangleq n^{2r}$ pre-images of length $k + 2r$, say $v_1, v_2, \ldots, v_N$. Let t be a positive integer parameter. Consider the cylinders of the form

$$[ux_1ux_2u \ldots x_{t-1}u]$$

where x_i vary over all words of length $2r$. Let U be the union of all such cylinders (for fixed t). We have that $\pi(U) = \pi([u])^t$. Let $V \triangleq \Phi^{-1}(U)$. Then V is the union of cylinders $[w]$ over all $w \in \{v_1, v_2, \ldots, v_N\}^t$, that is, over words w that are concatenations of t words v_i. We have that

$$\pi(V) = (\pi([v_1]) + \pi([v_2]) + \ldots + \pi([v_N]))^t = \pi(\Phi^{-1}([u]))^t = (a \cdot \pi([u]))^t = a^t \cdot \pi(U).$$

By choosing sufficiently large t we have that $\pi(V) < N \cdot m \cdot \pi(U)$, where m is the constant in (7). For some $w = ux_1ux_2u \ldots x_{t-1}u$ it must then be the case that $\pi(\Phi^{-1}([w])) < N \cdot m \cdot \pi([w])$. Because $\Phi^{-1}([w])$ is the disjoint union of N cylinders $[v]$ where v are such that $\Phi(v) = w$, we have that for some such v holds $\pi([v]) < m \cdot \pi([w])$, which violates (7).

($\implies$) Assume that π is invariant under Φ. Let y be an arbitrary periodic configuration, and let $x_1, x_2, \ldots x_k$ be its pre-images under Φ. Note that one-dimensional surjective CA are finite-to-one, so k is finite. It also follows that all x_i are periodic. Let $p > 2r$ be a sufficiently long common period of y and all x_i. We let $u \in S^p$ be such that $y = {}^\omega u^\omega$. For each i we take $v_i \in S^p$ similarly to be the repeating period in x_i so that $x_i = {}^\omega v_i^\omega$.

If period p is chosen sufficiently long, there exist sets

$$A_i, B_i \subseteq S^{p+r}, \text{ for } i = 1, 2, \ldots, k,$$

such that for every $t \geq 3$ the pre-images of the word u^t are precisely the words

$$\alpha \, v_i^{t-2} \, \beta$$

for $i = 1, 2, \ldots, k$ and $\alpha \in A_i$, $\beta \in B_i$. See Figure 2 for an illustration. The existence of such prefix and suffix sets A_i and B_i for sufficiently large p is a simple compactness argument.

Let $t \geq 3$ be an integer parameter, and consider the cylinder $U \triangleq [u^t]$ and its pre-image. We have that $\pi(U) = \pi([u])^t$, and by the Φ-invariance of π, also

$$\pi(U) = \pi(\Phi^{-1}(U)) = \sum_{i=1}^{k} \sum_{\alpha \in A_i} \sum_{\beta \in B_i} \pi([\alpha])\pi([v_i])^{t-2}\pi([\beta]).$$

Dividing by $\pi(U) = \pi([u])^t$ gives

$$1 = \sum_i^k c_i \left(\frac{\pi([v_i])}{\pi([u])} \right)^t ,$$

where

$$c_i \triangleq \sum_{\alpha \in A_i} \sum_{\beta \in B_i} \pi([\alpha])\pi([\beta])\pi([v_i])^{-2}$$

are independent of t.

The following obvious fact now implies that $\pi([v_i]) = \pi([u])$: if positive numbers $c_1, \ldots, c_k$ and $z_1, \ldots z_k$ satisfy

$$1 = c_1 z_1^t + \ldots + c_k z_k^t$$

for all $t = 3, 4, \ldots$, then necessarily all $z_i = 1$.

It follows from $\pi([v_i]) = \pi([u])$ that $\mu(v_i) = -\log \pi([v_i]) = -\log \pi([u]) = \mu(u)$. This proves the conservation of μ on periodic configurations.

$\square$

4 Generalization

Theorem 1 can be generalized in several directions. First, suppose that rather than a Bernoulli distribution, π is the distribution of a bi-infinite Markov chain with memory m, given by the transition probabilities $P(au, ub)$ for every $a, b \in S$ and $u \in S^{m-1}$. That is, we have $\pi([waub]_0 \,|\, [wau]_0) = P(au, ub)$ for every $a, b \in S$, $u \in S^{m-1}$ and $w \in S^*$. Assuming that all the transition probabilities $P(au, ub)$ are non-zero, we can define an additive quantity of range $m + 1$ by $\mu(aub) \triangleq -\log P(au, ub)$. The average value of μ on a periodic configuration x is defined to be the (well defined) average value of μ on the words seen through a window of width $m + 1$ that slides over x, and we say that μ is conserved by cellular automaton $\Phi : S^{\mathbb{Z}} \longrightarrow S^{\mathbb{Z}}$ if the average value of μ on x and $\Phi(x)$ are the same for all periodic x. For every surjective cellular automaton Φ, it can then be shown that π is invariant under Φ if and only if Φ conserves μ.

In higher dimensions, Theorem 1 remains valid as is, relating the invariance of Bernoulli distributions under surjective cellular automata and the conservation of range-1 additive quantities.

To state the theorem in its full generality (arbitrary dimensions, arbitrary finite range) we need few definitions. Let $d \geq 1$. Let $W \subseteq \mathbb{Z}^d$ be a finite set. An additive quantity with interaction window W is given by an assignment $\mu : S^W \longrightarrow \mathbb{R}$ of real numbers to patterns over W. If $x \in S^{\mathbb{Z}^d}$ is a configuration and $D \subseteq \mathbb{Z}^d$ a finite set, let us define $\mu_D(x)$ as follows: we slide the window W over x along the elements of D and sum the values of μ over the patterns seen. That is, $\mu_D(x) \triangleq \sum_{i \in D} \mu((\sigma^i x)|_W)$, where $\sigma^i x$ denotes the translation of x by i, and $(\sigma^i x)|_W$ the restriction of $\sigma^i x$ to W. As before, we can say that a d-dimensional cellular automaton $\Phi : S^{\mathbb{Z}^d} \longrightarrow S^{\mathbb{Z}^d}$ conserves μ if $\mu_D(\Phi(x)) = \mu_D(x)$ for every finite hypercube D and every periodic configuration x with fundamental domain D.

A *Gibbs measure* corresponding to μ is a Borel probability measure π satisfying

$$\pi([y]_D) = 2^{-(\mu_D(y) - \mu_D(x))} \pi([x]_D) , \tag{9}$$

for every two asymptotic configurations $x, y \in S^{\mathbb{Z}^d}$ and all sufficiently large finite sets $D \subseteq \mathbb{Z}^d$. Here, $[x]_D$ denotes the cylinder of all configurations that agree with x on D. The set D should be taken large

enough so that for every $i \notin D$, the configurations x and y cannot be distinguished by looking through the translated window $i + W$; that is, $x|_{i+W} = y|_{i+W}$.

The Gibbs measures (as defined above) coincide with the full-support Markov measures (see e.g. [Pre74, Geo88]). In particular, the one-dimensional Gibbs measures are precisely the distributions of Markov chains with positive transition probabilities. However, in higher dimensions, the Gibbs measure corresponding to an additive quantity is not necessarily unique. While it can be shown that for every additive quantity there corresponds at least one Gibbs measure, distinct Gibbs measures could satisfy (9) for the same μ. In statistical mechanics, Gibbs measures are used to describe the state of a system in thermal equilibrium. The multiplicity of Gibbs measures is then interpreted as the possibility of distinct equilibrium states at the same temperature (e.g., water vs. gas). See [Geo88, Pre74, KS80] for details and examples.

Theorem 2 *Let $\Phi : S^{\mathbb{Z}^d} \longrightarrow S^{\mathbb{Z}^d}$ be a d-dimensional surjective cellular automaton. Let μ be an additive quantity on $S^{\mathbb{Z}^d}$, and let $\mathscr{G}_\sigma(\mu)$ denote the set of translation-invariant Gibbs measures corresponding to μ. Then, the following conditions are equivalent:*

a) *Φ conserves μ.*

b) *Φ maps $\mathscr{G}_\sigma(\mu)$ onto itself.*

c) *There exists an element of $\mathscr{G}_\sigma(\mu)$ whose Φ-image is also in $\mathscr{G}_\sigma(\mu)$.*

The proof will appear in [KT].

5 Strongly Transitive Cellular Automata

In this section we give an example of how Theorem 2 can be used to transmit results between conservation laws research and invariant measures.

A cellular automaton $\Phi : S^{\mathbb{Z}^d} \longrightarrow S^{\mathbb{Z}^d}$ is *strongly transitive* if the backward orbit $\bigcup_{i=0}^\infty \Phi^{-i}(c)$ of every configuration $c \in S^{\mathbb{Z}^d}$ is dense in $S^{\mathbb{Z}^d}$. Equivalently, Φ is strongly transitive if for every non-empty cylinder $U \subseteq S^{\mathbb{Z}^d}$, the forward orbit $\bigcup_{i=0}^\infty \Phi^i(U)$ is the whole configuration space $S^{\mathbb{Z}^d}$. A strongly transitive CA is clearly surjective, and all positively expansive CA are strongly transitive.

Every CA has trivial conserved quantities: we call a quantity μ *trivial* if it assigns the same average value to all periodic configurations. For example, the interaction-free constant valuation ($\mu(s) = 1$ for all $s \in S$) is trivial. According to (9), the uniform Bernoulli measure is the unique Gibbs measure that corresponds to trivial conserved quantities.

The following theorem states that strongly transitive cellular automata only satisfy the trivial conservation laws.

Theorem 3 ([FKTar]) *Let $\Phi : S^{\mathbb{Z}^d} \longrightarrow S^{\mathbb{Z}^d}$ be a strongly transitive cellular automaton. Then Φ does not conserve any non-trivial additive quantities.*

Theorems 2 and 3 now immediately give the following result:

Corollary 4 *Let $\Phi : S^{\mathbb{Z}^d} \longrightarrow S^{\mathbb{Z}^d}$ be a strongly transitive cellular automaton. The uniform Bernoulli measure is the only translation-invariant Gibbs measure that is invariant under Φ.*

References

[Ber07]   Vincent Bernardi. *Lois de conservation sur automates cellulaires*. PhD thesis, Université de Provence, 2007.

[BF98]   Nino Boccara and Henryk Fukś. Cellular automaton rules conserving the number of active sites. *Journal of Physics A: Mathematical and General*, 31(28):6007–6018, 1998.

[DFR03]   Bruno Durand, Enrico Formenti, and Zsuzsanna Róka. Number conserving cellular automata I: decidability. *Theoretical Computer Science*, 299:523–535, 2003.

[FG03]   Enrico Formenti and Aristide Grange. Number conserving cellular automata II: dynamics. *Theoretical Computer Science*, 304:269–290, 2003.

[FKTar]   Enrico Formenti, Jarkko Kari, and Siamak Taati. On the hierarchy of conservation laws in a cellular automaton. *Natural Computing*, To appear.

[Geo88]   Hans-Otto Georgii. *Gibbs Measures and Phase Transitions*. Walter de Gruyter, 1988.

[Hed69]   G. A. Hedlund. Endomorphisms and automorphisms of the shift dynamical system. *Mathematical System Theory*, 3:320–375, 1969.

[HT91]   Tetsuya Hattori and Shinji Takesue. Additive conserved quantities in discrete-time lattice dynamical systems. *Physica D*, 49:295–322, 1991.

[KS80]   Ross Kindermann and J. Laurie Snell. *Markov Random Fields and Their Applications*. American Mathematical Society, 1980.

[KT]   Jarkko Kari and Siamak Taati. In preparation.

[Lin84]   D. A. Lind. Applications of ergodic theory and sofic systems to cellular automata. *Physica D: Nonlinear Phenomena*, 10(1–2), 1984.

[MBG04]   Andrés Moreira, Nino Boccara, and Eric Goles. On conservative and monotone one-dimensional cellular automata and their particle representation. *Theoretical Computer Science*, 325:285–316, 2004.

[Moo62]   Edward F. Moore. Machine models of self-reproduction. In *Proceedings of Symposia in Applied Mathematics*, pages 17–33. AMS, 1962.

[Myh63]   John Myhill. The converse of Moore's Garden-of-Eden theorem. *Proceedings of the American Mathematical Society*, 14:685–686, 1963.

[Piv02]   Marcus Pivato. Conservation laws in cellular automata. *Nonlinearity*, 15:1781–1793, 2002.

[Piv09]   Marcus Pivato. The ergodic theory of cellular automata. In *Encyclopedia of Complexity and System Science*. Springer, 2009.

[Pre74]   Christopher J. Preston. *Gibbs states on countable sets*. Cambridge University Press, 1974.

[Taa09]   Siamak Taati. *Conservation Laws in Cellular Automata.* PhD thesis, University of Turku, 2009.

[Wal82]   Peter Walters. *An Introduction to Ergodic Theory.* Springer-Verlag, 1982.

Projective subdynamics and universal shifts

Pierre Guillon[12][†]

[1]*CMM, Universidad de Chile*
[2]*CNRS & IML, Marseille, France*

We study the projective subdynamics of two-dimensional shifts of finite type, which is the set of one-dimensional configurations that appear as columns in them. We prove that a large class of one-dimensional shifts can be obtained as such, namely the effective subshifts which contain positive-entropy sofic subshifts. The proof involves some simple notions of simulation that may be of interest for other constructions. As an example, it allows us to prove the undecidability of all non-trivial properties of projective subdynamics.

Keywords: multidimensional symbolic dynamics, effective dynamics, tilings, simulation, undecidability

1   Introduction

Computation in dynamical systems has shown an increasing interest in the last decade. One of the questions that arises is the computational power of some models defined dynamically, where the computation result is seen as the "trace" of the system evolution or (equivalently) as a smaller system that it dynamically simulates. For cellular automata, this can be the limit set [Hur87, Maa95] or the column factors [Kůr97, CFG07]. For general effective dynamical systems, this can be observation problems with respect to some partitions [DKB05].

The setting of multidimensional symbolic dynamics is one of the most natural and elegant models with full computational power, as suggested by more recent results [Hoc09a, DRS10, AS10]. These works can be interpreted both as taking shifts of finite type as a model and subaction projections as a computing process or sofic shifts as a model and projective subdynamics as a process.

Independently, [PS10] presents some realization constructions as well as impossibility results in the weaker, yet natural case of projective subdynamics of shifts of finite type. Here, we also prove in this setting the constructability of a large class of effective shifts. To achieve this, we connect the problem to some simple notions of simulations over shifts.

Section 2 is devoted to the main definitions; Section 3 defines simulation and characterizes universality; Section 4 defines the main concept of the article, that of projective subdynamics, and recalls the known characterization in the sofic case; Section 5 introduces the intermediary notion of polyfactor, and gives a construction of it in the SFT case; Section 6 simulates it as projective subdynamics of SFT; finally Section 7 presents an independent application of the construction by proving a "Rice theorem" over projective subdynamics.

2 Preliminaries

We note $[\![i, j]\!]$ the set of integers $i, i+1, \ldots, j$, and $[\![i, j[\![= [\![i, j-1]\!]$. We also define $\mathbb{N}_1 = \mathbb{N} \setminus \{0\}$.

Let A be an alphabet (with $2 \leq |A| < \infty$) and $d \in \mathbb{N}_1$ the *dimension*. A *shape* is a subset $K \Subset \mathbb{Z}^d$, *i.e.*, $K \subset \mathbb{Z}^d$ and $|K| < \infty$. A *pattern* is a finite d-dimensional word $u = (u_i)_{i \in K} \in A^K$, where $K \Subset \mathbb{Z}^d$. A *configuration* is an infinite one $x = (x_i)_{i \in \mathbb{Z}^d} \in A^{\mathbb{Z}^d}$. For any $K \subset \mathbb{Z}^d$ and any configuration $x \in A^{\mathbb{Z}^d}$, we note x_K its restriction to K.

A *dynamical system* is a compact metric space X, on which some group G acts continuously. The *full* $\mathbb{Z}^d$-*shift* on alphabet A is the set of d-dimensional configurations $x \in A^{\mathbb{Z}^d}$, endowed with the product of the discrete topology, and with the action σ of $\mathbb{Z}^d$ defined for any $c, i \in \mathbb{Z}^d$ by $\sigma^c(x)_i = x_{c+i}$. We will mainly deal with subsystems of this, *i.e.*, closed subsets $\Sigma \subset A^{\mathbb{Z}^d}$ such that $\sigma^c(\Sigma) = \Sigma$ for any $c \in \mathbb{Z}^d$, which will be refered to as $\mathbb{Z}^d$-*shifts*.

Equivalently, a $\mathbb{Z}^d$-*shift* is a set $\Sigma \subset A^{\mathbb{Z}^d}$ of configurations defined via a collection of finite forbidden patterns $\mathcal{F}$, in the sense that $\Sigma = \left\{ x \in A^{\mathbb{Z}^d} \,\middle|\, \forall c \in \mathbb{Z}^d, \forall K \Subset \mathbb{Z}^d, \sigma^c(x)_K \notin \mathcal{F} \right\}$. Σ is *of finite type* (SFT) if $\mathcal{F}$ can be taken finite, *effective* if $\mathcal{F}$ can be taken recursively enumerable.

The topological closure of the orbit $\bigcup_{c \in \mathbb{Z}^d} \sigma^c(Z)$ of $Z \subset A^{\mathbb{Z}^d}$ will be denoted $\overline{Z}$. For instance, in dimension 1, a word u shall be seen as a map $i \mapsto u_i$ from $[\![0, |u|[\![$ to alphabet A. Then $\overline{{}^\infty u^\infty}$ will denote the set $\left\{ z \in A^{\mathbb{Z}} \,\middle|\, \exists k \in [\![0, |u|[\![, \forall j \in \mathbb{Z}, \sigma^{k+j|u|}(z)_{[\![0,|u|[\![} = u \right\}$ of configurations periodically equal to u.

If Σ is a $\mathbb{Z}$-shift over alphabet A, its *language of shape* $K \Subset \mathbb{Z}^d$ is the set $\mathcal{L}_K(\Sigma) = \{ z_K \,|\, z \in \Sigma \}$ of extendable patterns for this shape. These languages completely characterize Σ; moreover, by compactness, if Σ, Λ are two disjoint $\mathbb{Z}$-shifts, then there exists a finite shape $K \Subset \mathbb{Z}^d$ such that $\mathcal{L}_K(\Sigma) \cap \mathcal{L}_K(\Lambda) = \emptyset$.

We will actually essentially deal with $\mathbb{Z}^2$-shifts, but the generalization to higher dimensions is obvious.

3 Simulations

We define here some operations over shifts that can be seen as simulation rules, and which will help us to make constructions in the next sections. Similar compositions of operations have been recently studied in various settings [AS09, Hoc09b].

Let X and Y be dynamical systems corresponding to actions of the same group $\mathbb{Z}^d$, noted $\gamma^c : X \to X$ and $\delta^c : Y \to Y$ for $c \in \mathbb{Z}^d$. We note $X \succeq_f Y$ if there is a *factor map* $\Phi : X \to Y$, *i.e.*, an onto continuous map such that $\Phi \gamma^c = \delta^c \Phi$ for any $c \in \mathbb{Z}^d$; Y is then called a *factor* of X, and if Φ is bijective, X and Y are called *conjugate*. We note $X \succeq_i Y$ if the action δ on Y is conjugate to the action γ^k on X for some power $k \in \mathbb{N}_1^d$, where $(\gamma^k)^c = \gamma^{kc}$ for any $c \in \mathbb{Z}^d$ (and coordinatewise multiplication of vectors). We note $X \succeq_s Y$ if, up to conjugacy, $Y \subset X$ and δ is the restriction of γ to Y.

Let Σ and Γ be $\mathbb{Z}^d$-shifts over alphabets A and B, respectively. The previously-defined simulations can be visualized in a symbolic way. First note that $\Sigma \succeq_i \Gamma$ means that Γ is essentially the *bulking* $\Sigma^{[K]} = \left\{ (\sigma^{Kj}(x)_K)_{j \in \mathbb{Z}^d} \,\middle|\, x \in \Sigma \right\}$ (with coordinatewise multiplication) of Σ for some interval product $K \Subset \mathbb{Z}^d$, which is a shift over alphabet A^K. It can also be seen that for any nonempty $K \Subset \mathbb{Z}^d$, Σ is conjugate to its K-*block representation* $\Sigma^{(K)} = \left\{ (\sigma^i(x)_K)_{i \in \mathbb{Z}^d} \,\middle|\, x \in \Sigma \right\}$, which is a $\mathbb{Z}^d$-shift over alphabet A^K. A particular class of shift factor maps is that of *parallelizations* $\tilde{\Phi} : \Sigma \to \Gamma$ of *alphabet projections* $\Phi : A \to B$, *i.e.*, $\tilde{\Phi}(x)_i = \Phi(x_i)$ for any $x \in \Sigma$ and $i \in \mathbb{Z}^d$. It is also known that $\Sigma \succeq_f \Gamma$ if and only if $\Gamma = \tilde{\Phi}(\Sigma^{(K)})$ for some shape $K \Subset \mathbb{Z}^d$ and some parallel application $\tilde{\Phi}$ of some alphabet projection $\Phi : A^K \to B$.

Γ is called a $\mathbb{Z}^d$-*sofic* if it is a factor of a $\mathbb{Z}^d$-SFT. Equivalently from the last point, it is sofic if and only if it is the image of a $\mathbb{Z}^d$-SFT by some parallelization map. The classes of SFT, sofic shifts and effective shifts are closed under conjugacy, bulking and block representations. By the characterization above, that of sofic shifts is also closed under factor. On the contrary, $\succeq_s$ does not preserve any relevant property, which is why the simulation notion below will be very weak (it can be strengthened by taking the intersection with some SFT [AS09]); this will allow us to deal with a rather simple notion of universality, but will be compensated by the fact that our simulating shifts already have some structure (PSD in 9).

If $z : I \times J \to A$, for some intervals I and J of $\mathbb{Z}$ and $i \in I$, then we note $\pi_i(z) = (z_{i,j})_{j \in J}$. If $I = [\![0, m[\![$ and $J = \mathbb{Z}$, it gives a factor map π_i from any $\mathbb{Z}$-shift over alphabet A^m onto some $\mathbb{Z}$-shift over alphabet A. We also note $\pi_{I'}(z) = ((z_{i,j})_{i \in I'})_{j \in J}$ if $I' \subset I$.

The product $X \times Y$ of two sets $X \in A^{\mathbb{Z}^d}$ and $Y \in B^{\mathbb{Z}^d}$ will be abusively assimilated to the set $\left\{ w = (x_i, y_i)_{i \in \mathbb{Z}^d} \in (A \times B)^{\mathbb{Z}^d} \,\middle|\, (x_i)_{i \in \mathbb{Z}^d} \in X \text{ and } (y_i)_{i \in \mathbb{Z}^d} \in Y \right\}$ (which is a $\mathbb{Z}^d$-shift if X and Y are). We note $X^{<1>} = X$ and $X^{<n+1>} = X^{<n>} \times X$ for $n \in \mathbb{N}$. Essentially, $X^{<n>} = \left\{ y \in (A^n)^{\mathbb{Z}^d} \,\middle|\, \forall j \in [\![0, n[\![, \pi_j(y) \in X \right\}$. We note $\Sigma \succeq_p \Gamma$ if Γ is conjugate to some subshift of $\Sigma^{<n>}$ for some power $n \in \mathbb{N}$.

Each of these relations are not that interesting intrinsically, but can be associated together; the compositions will be noted $\succeq_{ps}$, $\succeq_{ifs}$, $\succeq_{pfs}$, *etc.*.. We can see, thanks to some commutation properties, that they are transitive whenever $\succeq_i$ and $\succeq_p$ are applied before $\succeq_s$.

We say that a $\mathbb{Z}$-shift Σ is *universal* if it simulates any other $\mathbb{Z}$-shift Γ in the sense that $\Sigma \succeq_{is} \Gamma$. This property is easily understood in the sofic case: indeed, uncountable $\mathbb{Z}$-sofic are exactly those that have positive entropy, and they can be represented on a graph with a non-cyclic strongly connected component (equivalently, they include some infinite transitive subshift).

Proposition 1 *Let Σ be a $\mathbb{Z}$-shift. The following statements are equivalent:*

1. *Σ includes some positive-entropy sofic subshift.*

2. *There are two words u and v with $u_0 \neq v_0$, $|u| = |v|$, and $\overline{^\infty \{u, v\}^\infty} \subset \Sigma$.*

3. *$\Sigma \succeq_{is} \{0, 1\}^{\mathbb{Z}}$.*

4. *Σ is universal.*

Proof:

$1 \Rightarrow 2$ If Σ includes a positive-entropy sofic subshift, then the graph of this subshift contains a non-cyclic strongly connected component, *i.e.*, there exists a vertex from which two arcs leave with two distinct labels, and which start paths that come back to the same vertex. Denoting $\tilde{u}, \tilde{v}$ the labels of these two paths, we can see that $u = \tilde{u}^{|\tilde{v}|}$ and $v = \tilde{v}^{|\tilde{u}|}$ satisfy the wanted conditions.

$2 \Rightarrow 3$ The $[\![0, |u| [\![$-bulking of $\overline{^\infty \{u, v\}^\infty}$ is a subshift that includes the full shift over alphabet $\{u, v\}$, which is essentially $\{0, 1\}^{\mathbb{Z}}$.

$3 \Rightarrow 1$ Remark that a non-trivial full shift is sofic and has positive entropy, as well as any of its iterations.

3⇒4 For any $\mathbb{Z}$-shift Σ on some alphabet A, we have $\{0,1\}^{\mathbb{Z}} \succeq_i A^{\mathbb{Z}} \succeq_s \Sigma$, since the letters of A are in bijection with some subset of $\{0,1\}^{\lceil \log |A| \rceil}$. Hence the full shift itself is universal, and the notions of simulation are transitive.

4⇒3 This is by definition of universality.

<div align="right">□</div>

In particular, the class of universal shifts is preserved by closing factor maps. Note that dealing with simulation $\succeq_{ifs}$ instead may widen the notion of universality to other subshifts. For our purpose though, this notion would be difficult to handle in the following.

Clock. Let $\mathcal{C}_n$ denote the n-cycle, *i.e.*, the dynamical system $[\![0,n[\![$ on which $\mathbb{Z}$ acts by $i \mapsto i + c \bmod n$ for any $c \in \mathbb{Z}$.

We have seen a definition of simulation that involves temporal delay, and one that involves spacial sprawl. The following lemma gives a transformation from the former to the latter.

Lemma 2 *If Γ is a $\mathbb{Z}$-shift, $A^{\mathbb{Z}}$ a full shift and $n \in \mathbb{N}$ such that $\Gamma^{[\![0,n[\![} \succeq_{fs} A^{\mathbb{Z}}$, then $\mathcal{C}_n \times \Gamma^{<n>} \succeq_{fs} A^{\mathbb{Z}}$.*

Proof: Let $\Lambda \subset \Gamma$ be closed and σ^n-invariant, and $\Phi : \Lambda \to A^{\mathbb{Z}}$ such that $\Phi\sigma^n = \sigma\Phi$. Let $\Lambda' = \left\{ (i,y) \in [\![0,n[\![\times \Gamma^{<n>} \mid \forall j \in [\![0,n[\![, \pi_j(y) \in \sigma^{j-i \bmod n}(\Gamma) \right\}$.

$$\Psi : \begin{array}{rcl} \Lambda' & \to & A^{\mathbb{Z}} \\ (i,y) & \mapsto & (\Phi(\sigma^j(\pi_{i+j \bmod n}(y)))_0)_{j \in \mathbb{Z}} \end{array}$$

is also a factor map, since for any $(i,y) \in \Gamma'$, we have:

$$\tilde{\Phi}(i+1 \bmod n, \sigma(y)) = (\Phi(\sigma^{j+1}(\pi_{i+1+j \bmod n}(y)))_0)_{j \in \mathbb{Z}} = \sigma(\tilde{\Phi}(i, y_0, \ldots, y_{n-1})) .$$

Moreover, Ψ is onto $A^{\mathbb{Z}}$ since, if $z \in A^{\mathbb{Z}}$, for $0 \le i < n$, the surjectivity of Φ and σ gives some $y^i \in Y$ such that $\Phi(\sigma^n(y^i)) = (z_{nj+i})_{j \in \mathbb{Z}}$. By construction, for any $j \in \mathbb{Z}$ and any $i \in [\![0,n[\![$, we have:

$$\Psi(0, \sigma^n(y^0), \sigma^{n-1}(y^1), \ldots, \sigma(y^{n-1}))_{nj+i} = \Phi(\sigma^{nj+i}(\sigma^{n-i}(y^i)))_0 = \sigma^{j+1}(\Phi(y^i))_0 = z_{nj+i} .$$

<div align="right">□</div>

Proposition 3 *If Σ is a $\mathbb{Z}$-shift, the following are also equivalent to universality (and to properties of Proposition 1):*

5. $\Sigma \succeq_{ps} \{0,1\}^{\mathbb{Z}}$.

6. *For any subshift Γ, $\Sigma \succeq_{ps} \Gamma$.*

Proof:

6⇒5⇒1 Positive entropy is preserved by product and supersystem, *i.e.*, the entropy of a system is more than that of any of its subsystems.

5⇒6 It is clear that every subshift can essentially be seen on an alphabet of the form $\{0,1\}^n$ with $n \in \mathbb{N}$, that is to say as being included in $(\{0,1\}^{\mathbb{Z}})^{<n>}$.

3&2$\Rightarrow$5 One can $\underline{\text{verify that}}$ if $u_0 \neq v_0$ and $|u| = |v| = n$, then $^\infty(uuv)^\infty$ is a word of smallest period $3n$, and that $\overline{^\infty(uuv)^\infty}$ is then conjugate to $\mathcal{C}_{3n}$, which factors $\underline{\text{onto}}$ $\mathcal{C}_n$. It is also clear that simulations are compatible with the product of systems. It results that $\overline{^\infty(uuv)^\infty} \times \Gamma^{<n>} \succeq_f \mathcal{C}_n \times \Gamma^{<n>} \succeq_{fs} \{0,1\}^{\mathbb{Z}}$ by Lemma 2 and, by hypothesis, $\overline{^\infty(uuv)^\infty} \subset \Gamma$, which gives $\Gamma^{<n+1>} \succeq_{fs} \{0,1\}^{\mathbb{Z}}$.

$\square$

4   Projective subdynamics

If $y = (y_{k,i})_{k,i\in\mathbb{Z}^2} \in A^{\mathbb{Z}^2}$ is a configuration and $k \in \mathbb{Z}$, then $\tau^k(y) = (y_{k,i})_{i\in\mathbb{Z}}$ will denote the projected k^{th} column. We note $\tau = \tau^0$. The *projective subdynamics* (PSD) of some $\mathbb{Z}^2$-shift X is the $\mathbb{Z}$-shift $\tau(X)$. The (vertical) *subaction* is the dynamical system where X is seen as being acted on by the restriction of σ to the subgroup $\{0\} \times \mathbb{Z}$ (only shifting vertically). Note that the PSD is a factor of the subaction by the map Φ defined by $\Phi(x)_i = x_{0,i}$ for $x \in X$ and $i \in \mathbb{Z}$.

The notions of PSD and subactions can of course be defined with respect to any dimension and any direction (subgroups of $\mathbb{Z}^d$), and all the following results will be adaptable in the general setting, but, for the sake of clarity, we will stick to the simple case of columns in bidimensional configurations.

If $\Sigma \subset A^{\mathbb{Z}}$ is a $\mathbb{Z}$-shift, let $\Sigma^{\mathbb{Z}}$ denote the $\mathbb{Z}^2$-shift $\left\{ x \in A^{\mathbb{Z}^2} \middle| \forall j \in \mathbb{Z}, \tau^j(x) \in \Sigma \right\}$. Remark that if Σ is an SFT, then so is $\Sigma^{\mathbb{Z}}$. If Φ is a factor map between the $\mathbb{Z}$-shifts Λ and Σ, then we can define a *parallelization* $\tilde{\Phi}$ from $\Lambda^{\mathbb{Z}}$ onto $\Sigma^{\mathbb{Z}}$ such that $\tau^k(\tilde{\Phi}(x)) = \Phi(\tau^k(x))$ for any $x \in \Lambda^{\mathbb{Z}}$ and any $k \in \mathbb{Z}$.

In the following sections, we will be interested in the PSD of $\mathbb{Z}^2$-SFT.

Proposition 4 *The class of PSD of $\mathbb{Z}^2$-SFT is invariant by product, conjugacy, and by SFT factor preimages, i.e., if Φ is a factor map from a $\mathbb{Z}$-SFT Σ onto a $\mathbb{Z}$-shift Λ and X a $\mathbb{Z}^2$-SFT with $\tau(X) \subset \Lambda$, then $\Phi^{-1}(\tau(X))$ is the PSD of some $\mathbb{Z}^2$-SFT.*

Proof:

- Clearly, the PSD of a product $\mathbb{Z}^2$-shift is the product $\mathbb{Z}$-shift of their two PSD.

- Assume that Φ is a conjugacy between a $\mathbb{Z}$-shift Σ and $\tau(X)$ for some $\mathbb{Z}^2$-SFT X, and $\tilde{\Phi} : \Sigma^{\mathbb{Z}} \to \Lambda^{\mathbb{Z}}$ its parallelization such that $\forall x \in \Sigma^{\mathbb{Z}}, \tau(\tilde{\Phi}(x)) = \Phi(\tau(x))$. It is clear that $\tilde{\Phi}$ is a conjugacy between $\Sigma^{\mathbb{Z}}$ and $\tau(X)^{\mathbb{Z}}$, and that $Y = \tilde{\Phi}^{-1}(X)$ is a $\mathbb{Z}^2$-SFT with $\tau(Y) = \Phi^{-1}(\tau(X) = \Sigma$.

- Let Φ be as in the statement. As above, its parallelization $\tilde{\Phi} : \Sigma^{\mathbb{Z}} \to \Lambda^{\mathbb{Z}}$ satisfies that the preimage $Y = \tilde{\Phi}^{-1}(X)$ is a $\mathbb{Z}^2$-SFT, since $\Sigma^{\mathbb{Z}}$ and X both are; by construction, $\tau(Y) = \Phi^{-1}(\tau(X))$.

$\square$

Before dealing further with the PSD of $\mathbb{Z}^2$-SFT, let us state what is known about PSD of $\mathbb{Z}^2$-sofic.

Proposition 5 *Let Σ be a $\mathbb{Z}$-shift. The following are equivalent.*

1. *Σ is the PSD of some $\mathbb{Z}^2$-sofic.*

2. *Σ is a factor of the PSD of some $\mathbb{Z}^2$-SFT.*

3. *Σ is the factor of the subaction of some $\mathbb{Z}^2$-SFT.*

Proof:

1⇒2 If $\Sigma = \tau(\tilde{\Phi}(X)) \subset A^{\mathbb{Z}}$ for some $\mathbb{Z}^2$-SFT X over alphabet B and some factor map $\tilde{\Phi}$ based on an alphabet projection $\Phi : B \to A$, then $\Sigma = \tilde{\Phi}(\tau(X))$ if we define $\tilde{\Phi}(x)_i = \Phi(x_i)$ for any $i \in \mathbb{Z}$.

2⇒1 Assume $\Sigma = \Phi(\tau(X))$ for some factor map Φ and some $\mathbb{Z}^2$-SFT X. Then the parallelization $\tilde{\Phi} : \tau(X)^{\mathbb{Z}} \to \Sigma^{\mathbb{Z}}$ is such that $\tau(\tilde{\Phi}(X)) = \Phi(\tau(X)) = \Sigma$, and $\tilde{\Phi}(X)$ is sofic as a factor of a $\mathbb{Z}^2$-SFT.

2⇒3 The PSD is a factor of the subaction, and the relation of factor is transitive.

3⇒2 Let Φ be a factor map from the vertical subaction of X onto Σ for some $\mathbb{Z}^2$-SFT X. By a standard uniform-continuity argument, some block representation of X will (while still being SFT) transform Φ into a simple projection to the central cell.

$\square$

As a consequence of this, the class of PSD of $\mathbb{Z}^2$-sofic is invariant by factor; it actually admits the following elegant characterization.

Let us define for any $\mathbb{Z}$-configuration $x \in A^{\mathbb{Z}}$ the $\mathbb{Z}^2$-configuration $\mathcal{A}(x)$ by $\tau^j(\mathcal{A}(x)) = x$, for any $j \in \mathbb{Z}$. If Σ is a $\mathbb{Z}$-shift over alphabet A, then $\mathcal{A}(\Sigma)$ is a $\mathbb{Z}^2$-shift.

Theorem 6 ([DRS10, AS10]) *The following are equivalent (and are thus also equivalent to the statements in Proposition 5).*

4. Σ *is effective.*

5. $\mathcal{A}(\Sigma)$ *is sofic.*

The problem of finding a similar characterization of $\mathbb{Z}$-shifts that can be obtained as PSD of $\mathbb{Z}^2$-SFT remains. It is clear that this class contains all $\mathbb{Z}$-SFT. In [CFG10], some constructions are given of cellular automata defined over $\mathbb{Z}$-SFT and that have specific *ultimate traces*, which actually give projective subdynamics of some $\mathbb{Z}^2$-SFT: in particular, all positive-entropy sofic subshifts can be obtained that way. On the other hand, [GR10] gives some impossibility results in that particular subsetting. In [PS10], both more constructions and impossibility results are presented, in the general setting. In particular, a full characterization of $\mathbb{Z}$-sofic PSD of $\mathbb{Z}^2$-SFT is given, emphasizing moreover on the difference between the so-called *stable* and *unstable* PSD. Note that the $\mathbb{Z}^2$-SFT can realize strictly fewer $\mathbb{Z}$-shifts than $\mathbb{Z}^2$-sofic do. There are even $\mathbb{Z}$-sofic that are not realizable as PSD of $\mathbb{Z}^2$-SFT, such as the shift of the configurations having state 0 everywhere except for at most one cell. The next two sections are devoted to realizing some class of $\mathbb{Z}$-shifts that goes further than the sofic case.

5 Polyfactors

The *polyfactor* of some $\mathbb{Z}^2$-shift X over alphabet A^m, with $m \in \mathbb{N}_1$, is the union $\overset{\circ}{\tau}(X) = \bigcup_{0 \le i < m} \pi_i(\tau(X))$, which can be seen as the projective subdynamics of some system which is invariant by some powers of the shift, but not by the whole action (periodic local constraints). Note that the notion of polyfactor depends on the interpretation of the alphabet as a power of another alphabet. In particular, the projective subdynamics of some shift is also its polyfactor (if we interpret $m = 1$).

Let us see conditions on the subshift that allow it to be the polyfactor of some SFT.

Lemma 7 *If Σ is a $\mathbb{Z}$-shift and X a $\mathbb{Z}^2$-shift such that $\Sigma \succeq_{ps} \tau(X) \succeq_f \Sigma$, then Σ is the polyfactor of some $\mathbb{Z}^2$-shift Y conjugate to X.*

Proof: Let $n \in \mathbb{N}$ be such that $\tau(X) \subset \Sigma^{<n>}$, Ψ be a factor map from $\tau(X)$ onto Σ, and $\tilde{\Psi}$ its parallelization, *i.e.*, $\forall x \in X, \tau(\tilde{\Psi}(x)) = \Psi(\tau(x))$. The product $Y = \tilde{\Psi}(X) \times X$ is conjugate to X (they are linked by maps $\Psi \times$ id and π_1). Moreover, it can be seen as a $\mathbb{Z}^2$-shift with $1 + n$ columns; the first one is equal to Σ, and the other n are included in it. $\quad\square$

The interest of introducing polyfactors of $\mathbb{Z}^2$-SFT is that their class is more robust than that of their projective subdynamics, as illustrated by the following remarks (all of which are not useful for our main construction).

Proposition 8 *The class of polyfactors of $\mathbb{Z}^2$-SFT is invariant by projection union (if the alphabet is a power), product, union, conjugacy, weak iteration (reading every n letters), and by SFT factor preimages (see Proposition 4).*

Proof:

- If $\Sigma = \overset{\circ}{\tau}(X)$ where X is a $\mathbb{Z}^2$-SFT over alphabet $(A^m)^n$, with $m, n \in \mathbb{N}_1$, then $\bigcup_{0 \leq i < m} \pi_i(\Sigma)$ is also the polyfactor of X, seen as a $\mathbb{Z}^2$-SFT over alphabet A^{mn}.

- Assume $\Sigma = \overset{\circ}{\tau}(X) \subset A^{\mathbb{Z}}$ and $\Gamma = \overset{\circ}{\tau}(Y) \subset B^{\mathbb{Z}}$, *i.e.*, there are $m, n \in \mathbb{N}$ such that $\tau(X) \subset \Sigma^{<m>}$ and $\tau(Y) \subset \Gamma^{<m>}$. Then $\Sigma \times \Gamma$ can be seen as the polyfactor of the $\mathbb{Z}^2$-shift $\{z| \exists x \in \Sigma, y \in \Gamma, \forall i \in [\![0,m[\![, \forall j \in [\![0,n[\![, \pi_{i+jm}(z) = (\pi_i(x), \pi_j(m))\}$ over alphabet $(A \times B)^{mn}$.

- The previous two points give the union.

- If a $\mathbb{Z}$-shift Σ over alphabet A is conjugate to $\overset{\circ}{\tau}(X)$ for some $\mathbb{Z}^2$-SFT over alphabet A^m for some $m \in \mathbb{N}$, then it is clear that this conjugacy can be parallelized into a conjugacy $\tilde{\Phi} : \Sigma^{<m>} \to \overset{\circ}{\tau}(X)^{<m>}$; by Proposition 4, $\tilde{\Phi}^{-1}(\tau(X))$ can be obtained as $\tau(Y)$ for some $\mathbb{Z}^2$-SFT Y. One can see that $\overset{\circ}{\tau}(Y) = \bigcup_{0 \leq i < m} \pi_i(\tilde{\Phi}^{-1}(\tau(X))) = \bigcup_{0 \leq i < m} \Phi^{-1}(\pi_i(\tau(X))) = \Phi^{-1}(\overset{\circ}{\tau}(X)) = \Sigma$.

- Invariance by SFT factor preimage comes from 4 in the same way as conjugacy.

- If $K \in \mathbb{Z}^d$ and X is a $\mathbb{Z}^2$-SFT, then the bulking $X^{[K]}$ is one also; its polyfactor $\overset{\circ}{\tau}(X^{[K]})$ consists exactly of all weak iterations of $\overset{\circ}{\tau}(X)$.

$\quad\square$

Proposition 9 *If $\Sigma \succeq_{ps} \Lambda \succeq_{fs} \tau(X) \succeq_f \Sigma$ for some $\mathbb{Z}$-SFT Λ and some $\mathbb{Z}^2$-SFT X, then Σ is the polyfactor of some $\mathbb{Z}^2$-SFT.*

Proof: Let $n \in \mathbb{N}$ be such that $\Lambda \subset \Sigma^{<n>}$, $\Gamma \subset \Lambda$ and $\Phi : \Gamma \to \tau(X)$ a factor map, which can actually be extended to Λ. Then $\Phi^{-1}(\tau(X))$ is the PSD of some $\mathbb{Z}^2$-SFT Y thanks to Proposition 4. We obtain $\Sigma \succeq_{ps} \Lambda \succeq_s \tau(Y)$ and $\tau(Y) \succeq_f \tau(X) \succeq_f \Sigma$, which gives $\Sigma \succeq_{ps} \tau(Y) \succeq_f \Sigma$; Lemma 7 allows to conclude. $\quad\square$

Corollary 10 *If Σ is an effective $\mathbb{Z}$-shift and contains some positive-entropy sofic subshift, then it is the polyfactor of some $\mathbb{Z}^2$-SFT.*

Proof: It is enough to use Proposition 9 with Λ some full shift simulated by Σ (see Proposition 1) and X some $\mathbb{Z}^2$-sofic whose projective subdynamics is Σ (see Theorem 6). $\qquad\square$

6 Projective subdynamics of SFT

Let us see a construction that turns the polyfactor of some SFT into the projective subdynamics of a modified SFT.

Consider a $\mathbb{Z}$-shift S over alphabet A^m for some $m \in \mathbb{N}$. S is *marking* if $S_0 \cap S_i = \emptyset$ for $0 < i < m$ and $S_i = \left\{ w \in (A^{2m-1})^{\mathbb{Z}} \middle| \pi_{[\![i,i+m[\![}(w) \in S \right\}$. By compactness, we have that S is marking if and only if there exists some length $l \in \mathbb{N}$ such that the languages $\mathcal{L}_{[\![0,l[\![}(S_i)$ are pairwise disjoint for $i \in [\![0,m[\![$.

This definition can be seen as the impossibility to interpret a two-dimensional configuration into two distinct juxtapositions of stripes of S. Of course, any subshift of a marking shift is also marking.

Here are two classes of examples of marking shifts.

Example 11 *If Γ and Λ are two disjoint $\mathbb{Z}$-shifts over alphabet A and $m \in \mathbb{N}$, then the following set is marking:*
$$S_{\Gamma,\Lambda}^m = \left\{ w \in (A^{2m+2})^{\mathbb{Z}} \middle| \forall i \in [\![m,2m]\!], \pi_i(w) \in \Gamma \text{ and } \pi_{2m+1}(w) \in \Lambda \right\} \ .$$

Example 12 *If u and v are two distinct words with same length over alphabet A and $m \in \mathbb{N}$, then the following set is marking:*
$$S_{u,v}^m = \left\{ w \in (A^{2m+2})^{\mathbb{Z}} \middle| \exists j \in [\![0,|u|]\!], \forall i \in [\![m,2m]\!], \pi_i(w) = \sigma^j(^{\infty}u^{\infty}) \text{ and } \pi_{2m+1}(w) = \sigma^j(^{\infty}v^{\infty}) \right\} \ .$$

For $x \in (A^m)^{\mathbb{Z}^2}$, we define the *$m$-unbulking* of x with shift $i \in \mathbb{Z}$ as the configuration $y = \boxtimes_m^i(x)$ over alphabet A defined by $\tau^{[\![i+km,i+(k+1)m[\![}(y) = \tau^k(x)$. For X a $\mathbb{Z}^2$-shift over alphabet A^m, we define the *m-unbulking* of X as the $\mathbb{Z}^2$-shift $\boxtimes_m(X) = \bigcup_{i \in [\![0,m[\![} \boxtimes_m^i(X)$ over alphabet A. It flattens the configurations by alternating the layers (like the contrary of a $[\![0,m[\![\times\{0\}$-bulking). In particular, $\tau(\boxtimes_m(X)) \overset{\circ}{=\tau} (X)$. If X is an SFT, then $\boxtimes_m(X)$ need not be so, but this is where marking shifts is useful.

Lemma 13 *Let X be a $\mathbb{Z}^2$-SFT over alphabet A^m for some $m \in \mathbb{N}$, such that $\tau(X)$ is marking. Then $\boxtimes_m(X)$ is a $\mathbb{Z}^2$-SFT over alphabet A, and $\tau(\boxtimes_m(X)) \overset{\circ}{=\tau} (X)$. Moreover, its local constraints can be effectively computed from that of X.*

Proof: Since the $S_i = \left\{ w \in (A^{2m-1})^{\mathbb{Z}} \middle| \pi_{[\![i,i+m[\![}(w) \in \tau(X) \right\}$ are disjoint for $i \in [\![0,m[\![$, by compactness they actually differ on patterns of a bounded height. Hence some local constraints can impose patterns of this height to cycle through the S_i. Moreover, for each i, it is easy to check locally the constraints of X (that may have a larger range) with respect to this unique interpretation. $\qquad\square$

Lemma 14 *Let X be a $\mathbb{Z}^2$-SFT over alphabet A^m, and Y, Z two nonempty $\mathbb{Z}^2$-SFT over alphabet A such that $\tau(Y) \cap \tau(Z) = \emptyset$. Then $X' = \boxtimes_{2m+2}(X \times Y^{<m+1>} \times Z)$ is a $\mathbb{Z}^2$-SFT. It is empty if X is, otherwise $\tau(X') \overset{\circ}{=\tau} (X) \cup \tau(Y) \cup \tau(Z)$. Moreover, its local constraints can be effectively computed from that of X, Y, Z.*

Proof: The PSD $\tau(X')$ is included in $S^m_{\tau(Y),\tau(Z)}$, which is marking by Example 11. Hence Lemma 13 gives the result. It is clear that everything is effective. $\qquad\square$

Proposition 15 *Let Σ be the polyfactor of some $\mathbb{Z}^2$-SFT X over alphabet A^m for some $m \in \mathbb{N}_1$, and Y, Z two nonempty $\mathbb{Z}^2$-SFT over alphabet A such that $\tau(Y) \cap \tau(Z) = \emptyset$. Then $\Sigma \cup \tau(Y) \cup \tau(Z)$ is the PSD of some $\mathbb{Z}^2$-SFT over A.*

Proof: If $\Sigma \neq \emptyset$, then Lemma 14 gives the result; otherwise we can apply the same lemma while fixing $m = 0$. $\qquad\square$

The interesting case will actually be when $\tau(Y)$ and $\tau(Z)$ are contained in Σ.

Theorem 16 *Any effective $\mathbb{Z}$-shift including some positive-entropy sofic subshift is the PSD of some $\mathbb{Z}^2$-SFT.*

Proof: This directly comes from Proposition 15, Corollary 10, and the fact that any positive-entropy $\mathbb{Z}$-sofic contains two disjoint periodic orbits, which are trivially realizable as PSD of periodic $\mathbb{Z}^2$-SFT. $\qquad\square$

Any effective universal $\mathbb{Z}$-shift is then realizable in that sense, and note that their class is preserved by closing maps.

Another consequence of this construction is the following.

Corollary 17 *Let $(X_i)_{0 \leq i < m}$ be a finite family of $\mathbb{Z}^2$-SFT among which two have disjoint PSD, say $\tau(X_0) \cap \tau(X_1) = \emptyset$. Then $\bigcup_{0 \leq i < m} \tau(X_i)$ is the PSD of some $\mathbb{Z}^2$-SFT.*

Proof: By Proposition 8, $\bigcup_{2 \leq i < m} \tau(X_i)$ is the polyfactor of some $\mathbb{Z}^2$-SFT. Then Proposition 15 gives the result. $\qquad\square$

The simple cases of application of this corollary are in the case of two PSD which are either disjoint or contain two distinct periodic orbits. We can be more precise: by using Example 12 in Lemma 14, we can reprove [PS10, Proposition 5.3]: if $(X_i)_{0 \leq i < m}$ is a finite family of $\mathbb{Z}^2$-SFT such that $\Sigma = \bigcup_{0 \leq i < m} \tau(X_i)$ contains two distinct periodic configurations, then Σ is the PSD of some $\mathbb{Z}^2$-SFT. This is not a direct corollary of the previous statement, since the two distinct periodic configurations could here be in the same non-uniform periodic orbit.

7 Undecidability

The transformation of polyfactors into projective subdynamics allows the following theorem à la Rice. This is largely inspired by [CG07, CFG10], but note that it is not a direct corollary of the corresponding result on traces of cellular automata, since we deal here with more non-trivial properties.

Theorem 18 *For any property $\mathcal{P}$ satisfied by the PSD of some $\mathbb{Z}^2$-SFT over alphabet $\{0, 1\}$, but not all of them, the following problem is undecidable:*

> **Input:** *a $\mathbb{Z}^2$-SFT over alphabet $\{0, 1\}$.*
> **Problem:** $\tau(X) \in \mathcal{P}$?

Proof: Assume that the full shift $\{0,1\}^{\mathbb{Z}}$ satisfies $\mathcal{P}$ (otherwise consider the complement of $\mathcal{P}$); let Y be some $\mathbb{Z}^2$-SFT such that $\tau(Y)$ does not satisfy $\mathcal{P}$, and $w \in \{0,1\}^{[\![0,l[\![^2}$ a forbidden pattern for Y, with $l \in \mathbb{N}$. We can consider the (periodic) $\mathbb{Z}$-SFT $\overline{\infty w^\infty}$ over alphabet A^l, as a periodic vertical superposition of these blocks w. Let us prove that, if the problem above was decidable, then we could decide the emptiness of binary $\mathbb{Z}^2$-shifts. Indeed, let us be given an arbitrary $\mathbb{Z}^2$-shift X over alphabet $\{0,1\}$. We can compute the $\mathbb{Z}^2$-SFT $X' = (\overline{\infty w^\infty})^{\mathbb{Z}} \times X \times \{0,1\}^{\mathbb{Z}^2}$, that we must see over alphabet $\{0,1\}^{l+2}$, with l layers representing periodic superpositions of blocks w, a layer representing X and a layer with a full shift. Then from Lemma 14, we can compute the $\mathbb{Z}^2$-SFT $X'' = \boxtimes_{2l+6}(X' \times \{\infty 0^\infty\}^{<l+3>} \times \{\infty 1^\infty\})$. Now, since pattern w appears periodically in configurations of X'', we have that $Y' = Y \sqcup X''$ is still a $\mathbb{Z}^2$-SFT. If X is empty, then so is X', and so is X'', hence $Y' = Y$ and $\tau(Y') = \tau(Y) \notin \mathcal{P}$. Otherwise, $\tau(Y') \supset \tau(X'') \overset{\circ}{\supset} \tau(X') \supset \{0,1\}^{\mathbb{Z}}$, hence $\tau(Y') = \{0,1\}^{\mathbb{Z}} \in \mathcal{P}$. As a consequence, if we could decide whether the PSD of the $\mathbb{Z}^2$-SFT Y satisfied $\mathcal{P}$ or not, then we would be able to decide whether X is empty. Yet, this problem is known to be undecidable (see [Ber66] for a proof on Wang tile model, which can easily be simulated effectively by binary $\mathbb{Z}^2$-SFT). □

Taking Y sofic instead of SFT allows the same statement for $\mathbb{Z}^2$-sofic.

8 Conclusion

We have presented a construction of SFT that have a given PSD among a large class. It is clear that it could be adapted to PSD corresponding to dimensions higher than 2, codimensions higher than 1, and other subgroups than vertical columns.

However, this construction leaves as open problems a general characterization of PSD of SFT. Some effective positive-entropy shifts may not include positive-entropy sofic subshifts. A difficult case is also the case of null-entropy shifts. It was well understood by [PS10] in the sofic case; this construction can be thought of as some kind of simulation, and maybe Proposition 9 could involve simulations performed by non-universal shifts. Impossibility results are also lacking outside the sofic one-dimensional case [PS10].

What could be interesting too is to study the case of deterministic SFT (or, equivalently, with some given expansiveness directions). But it is already difficult to understand the case of deterministic sofic (for instance whether the construction of [AS10] could be "determinized"). Cellular automata (which correspond to deterministic SFT with additional regularity properties, or seen as actions of $\mathbb{N} \times \mathbb{Z}$ rather than $\mathbb{Z}^2$) have been the subject of independent works, still far from characterizations, be it the limit set (PSD orthogonal to the expansiveness direction) [Hur87, Maa95] or the trace (parallel) [CFG07, CFG10].

Another question was asked by E. Jeandel and R. Pavlov [Pav10, Question 2]: in a similar flavor to Theorem 6, what can we say about the class of $\mathbb{Z}$-shifts Σ such that $\Sigma^{\mathbb{Z}}$ is a $\mathbb{Z}^2$-sofic? It is clear that Σ is SFT if and only if $\Sigma^{\mathbb{Z}}$ is, showing that this kind of dimension increase leaves much less freedom than $\mathcal{A}$. It seems there are no example of non-sofic Σ with $\Sigma^{\mathbb{Z}}$ sofic.

References

[AS09]   Nathalie Aubrun and Mathieu Sablik. An order on sets of tilings corresponding to an order on languages. In Susanne Albers and Jean-Yves Marion, editors, 26th *International Symposium on Theoretical Aspects of Computer Science (STACS'09)*, Freiburg, Germany, February 2009. IBFI Schloss Dagstuhl.

[AS10]   Nathalie Aubrun and Mathieu Sablik. Simulation of effective subshifts by two-dimensional sft and a generalization. preprint, 2010.

[Ber66] Robert Berger. The undecidability of the domino problem. *Memoirs of the American Mathematical Society*, 66:72, 1966.

[CFG07] Julien Cervelle, Enrico Formenti, and Pierre Guillon. Sofic trace of a cellular automaton. In S. Barry Cooper, Benedikt Löwe, and Andrea Sorbi, editors, *Computation and Logic in the Real World, 3rd Conference on Computability in Europe (CiE07)*, volume 4497 of *Lecture Notes in Computer Science*, pages 152–161, Siena, Italy, June 2007. Springer-Verlag.

[CFG10] Julien Cervelle, Enrico Formenti, and Pierre Guillon. Ultimate traces of cellular automata. In Jean-Yves Marion, editor, 27th *International Symposium on Theoretical Aspects of Computer Science (STACS'10)*, Nancy, France, March 2010.

[CG07]   Julien Cervelle and Pierre Guillon. Towards a Rice theorem on traces of cellular automata. In Ludek Kučera and Antonín Kučera, editors, 32nd *International Symposium on the Mathematical Foundations of Computer Science*, volume 4708 of *Lecture Notes in Computer Science*, pages 310–319, Český Krumlov, Czech Republic, August 2007. Springer-Verlag.

[DKB05] Jean-Charles Delvenne, Petr Kůrka, and Vincent Blondel. Decidability and universality in symbolic dynamical systems. *Fundamenta Informaticæ*, XX:1–25, 2005.

[DRS10] Bruno Durand, Andrei Romashchenko, and Alexander Shen. Fixed-point tile sets and their applications. draft, September 2010.

[GR10]   Pierre Guillon and Gaétan Richard. Asymptotic behavior of dynamical systems. preprint, April 2010.

[Hoc09a] Michael Hochman. On the dynamics and recursive properties of multidimensional symbolic systems. *Inventiones Mathematicæ*, 176(1):131–167, April 2009.

[Hoc09b] Michael Hochman. On universality in multidimensional symbolic dynamics. *Discrete & Continuous Dynamical Systems*, 2(2), 2009.

[Hur87] Lyman P. Hurd. Formal language characterizations of cellular automaton limit sets. *Complex Systems*, 1:69–80, 1987.

[Kůr97] Petr Kůrka. Languages, equicontinuity and attractors in cellular automata. *Ergodic Theory & Dynamical Systems*, 17(2):417–433, April 1997.

[Maa95] Alejandro Maass. On the sofic limit set of cellular automata. *Ergodic Theory & Dynamical Systems*, 15:663–684, 1995.

[Pav10] Ronnie Pavlov. A class of nonsofic $\mathbb{Z}^d$ shift spaces. preprint, 2010.

[PS10]   Ronnie Pavlov and Michael Schraudner. Classification of sofic projective subdynamics of multidimensional shifts of finite type. preprint, 2010.

NOCAS : A Nonlinear Cellular Automata Based Stream Cipher

Sandip Karmakar[1][†] and Dipanwita Roy Chowdhury[1][‡]

[1]*Indian Institute of Technology, Kharagpur, WB, India*

LFSR and NFSR are the basic building blocks in almost all the state of the art stream ciphers like Trivium and Grain-128. However, a number of attacks are mounted on these type of ciphers. Cellular Automata (CA) has recently been chosen as a suitable structure for crypto-primitives. In this work, a stream cipher is presented based on hybrid CA. The stream cipher takes 128 bit key and 128 bit initialization vector (IV) as input. It is designed to produce 2^{128} random keystream bits and initialization phase is made faster 4 times than that of Grain-128. We also analyze the cryptographic strength of this cipher. Finally, the proposed cipher is shown to be resistant against known existing attacks.

Keywords: Cellular Automata, Stream Cipher, NMix, Hybrid Nonlinear Cellular Automata

1   Introduction

The mass use of hand-held devices/PDA has popularized the use of stream ciphers. Stream ciphers are much less power consuming, requires small space for their operations and are faster in operation than other cryptographic algorithms. Generally, in stream ciphers a secret key and a public IV are input. Keystream bits are generated by the cipher per cycle of operation. The plain-text is XORed on the encryption side with the generated keystream to produce the cipher-text. Decryption is carried out by simply XORing the cipher-text with the keystream. The eStream project which started in year 2004 was an attempt to standardize stream ciphers. A large number of stream ciphers were submitted to this project. After a cryptanalysis phase ranging over 4 years, stream ciphers were filtered in 3 phases by their performance and security. At the final stage has Trivium [CP], Grain [HJM] and MICKEY [BD] which are hardware efficient and Rabbit [BVCZ], Salsa20/12 [Ber], HC-128 [Wu], SOSEMANUK [BBC[+]] that are software based stream ciphers.

The eStream project categorized stream ciphers in two sections, hardware based and software based. Software based ciphers are expected to have optimized software performance, while hardware based ciphers are optimized for hardware. The submitted software based ciphers had a nonlinear filter function which combines LFSR (Linear Feedback Shift Register) and NFSR (Nonlinear Feedback Shift Register)

[†]Email: `sandip1kk@gmail.com`.
[‡]

bits. Trivium [CP] is reported as the fastest cipher providing hardware performance. Grain-128 [HJM] is the next cipher in terms of hardware performance. It combines a LFSR and a NFSR bits by a nonlinear function. However, Grain-128 has been subjected to many attacks, like, dynamic cube attack [DS11], fault attacks [BCC+09], [KC11]. [BCC+09] breaks the cipher by inducing faults in the LFSR of Grain-128, while [KC11] breaks the cipher by injecting faults in the NFSR of the cipher. Our design of NOCAS follows the structure of Grain-128, it replaces the LFSR and NFSR by a maximum length CA and a hybrid nonlinear CA. The nonlinear filter function is replaced by NMix, a nonlinear key mixing function used for block ciphers. NOCAS is shown to be resistant against fault attack and initialization becomes 4 times faster than Grain.

Cellular Automata were studied as a good pseudorandom sequence generator. The main requirement of a stream cipher is good pseudorandom generation. Also parallel operations of CA, which may give high troughput to ciphers. Rule-30 based CA was studied by Wolfarm as a pseudorandom generator. But it was later cryptanalyzed by Miere and Stafflebach [MS91] mainly due to its correlation. This shows that only nonlinear CA needs to be operated to reduce its correlation. [KMC10] studied few hybrid CA structures for cryptographic applications. It is shown that those CA can provide good cryptographic characteristics. In this paper, we have chosen one such hybrid CA rule for nonlinear mixing of key bits and a maximum length CA for linear mixing and high period. The cipher takes 128 bit key and 128 bit initialization vector (IV). It initializes in 64 cycles. Bits from hybrid nonlinear CA and the maximum length linear CA are combined with a nonlinear filter function NMix to produce output bit. In the current paper, we show that NOCAS is expected to have high security and provides security against known attacks.

The paper is organized as follows. Following the introduction, we briefly discuss the basic definitions regarding cellular automata (CA) and give a brief specification of Grain-128 cipher in section 2. NOCAS is proposed in section 3. Security analysis of NOCAS is studied in section 4. The hardware implementations of NOCAS and Grain-128 are compared in section 5. Finally, the paper is concluded in section 6.

2   Preliminaries

In this section, we provide definitions relating CA and cryptographic properties. We also give a brief specification of the Grain-128 stream cipher.

2.1   Basics of Cellular Automata

A cellular automaton is a finite array of cells. Each cell is a finite state machine $C = (\{0, 1\}, f)$ where, f is a mapping $f : \{0, 1\}^n \to \{0, 1\}$. The mapping f, called local transition function. n is the number of cells the local transition function depends on. On each iteration the CA each cell of the CA updates itself with respective f.

The number of neighbouring cells, f depends on, may be same or different on different directions of the automaton. f may be same or different for cells across the automaton. The array of cells may be multi-dimensional. A 1-dimensional CA, each of whose rule depends on left and right neighbour and the cell itself is called a 3-neighbourhood CA. Similarly, if each cell depends on 2 left and 2 right neighbours and itself only, it is called 5-neighbourhood CA. A CA whose cells depend on 1 left and 2 right neighbouring cells is called a 4-neighbourhood right skew CA. A left skewed 4-neighbourhood CA can be defined similarly.

Tab. 1: Truth table for $f = q_{i-1}(t) \oplus q_i(t)$

Input	Output
000	0
001	0
010	1
011	1
100	1
101	1
110	0
111	0
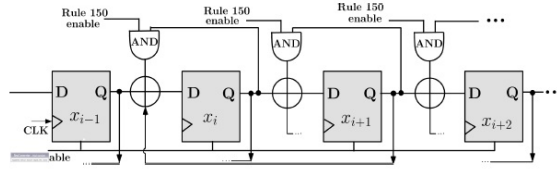
Fig. 1: A 4 Cell Linear Hybrid Cellular Automata based on Rules 90, 150

The state of the ith cell at time $(t + 1)$ depends on states of $(i - 1)^{th}$, i^{th} and $(i + 1)^{th}$ cells at time t. So, the local transition function for a 3-neighbourhood CA cell can be expressed as follows:

$$q_i(t + 1) = f[q_i(t), q_{i+1}(t), q_{i-1}(t)]$$

where, f denotes the local transition function realized with a combinational logic, and is known as a rule of CA [CCNC]. The decimal value of the truth table of the local transition function is defined as the *rule number* of the cellular automaton. For example, consider, $f = q_{i-1}(t) \oplus q_i(t)$. Its truth table is shown in tab. 1. Since the decimal equivalent of the output 00111100 is 60, rule number of f is, 60. Other examples are:
Rule 30: $f = q_{i-1}(t) \oplus (q_{i+1}(t) + q_i(t))$, where + is the Boolean 'or' operator and $\oplus$ is the Boolean 'xor' operator.
Rule 90: $f = q_{i-1}(t) \oplus q_{i+1}(t)$.
Rule 150: $f = q_{i-1}(t) \oplus q_i(t) \oplus q_{i+1}(t)$.

If the rule of all the cells are the same then it is called uniform cellular automata, otherwise, it is called hybrid cellular automata. A 4 cell linear hybrid cellular automata is shown in Fig. 1. This work employs both linear and nonlinear CA. We define linear and nonlinear cellular automata below, before proceeding further.

Definition 1 *Linear Cellular Automaton: A CA whose local transition function does not involve the '.' (Boolean and) operator in any of the cell is called the linear cellular automaton. For example, rule, $f = q_{i-1}(t) \oplus q_{i+1}(t)$ employed in each cell is a linear cellular automaton, where $q_{i-1}(t)$ and $q_{i+1}(t)$ denotes left and right neighbours of i-th cell at t-th instance of time.*

Definition 2 *Nonlinear Cellular Automaton: A CA whose local transition function is non-linear, i.e., involves at least one . operator, for at least one of the cells is a nonlinear cellular automaton. For example, rule, $f = q_{i-1}(t).q_{i+1}(t)$ employed in each cell is a nonlinear cellular automaton, where, $q_{i-1}(t)$ and $q_{i+1}(t)$ denotes left and right neighbours of the ith cell at t^{th} instance of time.*

2.2 Cryptographic Terms and Primitives

We next provide definitions of various terms and properties which Boolean functions should satisfy for cryptographic applications.

Definition 3 *Pseudorandom Sequence: An algorithmic sequence is pseudorandom if it cannot be distinguished from a truly random sequence by any efficient (polynomial time) probabilistic procedure or circuit.*

Definition 4 *Affine Function: A Boolean function which can be expressed as 'xor' ($\oplus$) of some or all of its input variables and a Boolean constant is an affine function.*

For example, $f(x_1, x_2) = x_1 \oplus x_2$ is an *affine function*, while the function, $f(x_1, x_2) = x_1 \oplus x_2 \oplus x_1.x_2$ is not an *affine function*, where, . is the Boolean 'and' operation and $\oplus$ is the Boolean 'xor' operation.

Definition 5 *Hamming Weight: Number of Boolean 1's in a Boolean function's truth table is called the Hamming weight of the function.*

Hamming weight of a function f is denoted as, $wt(f)$. For example, Hamming weight of $f(x_1, x_2) = x_1 \oplus x_2$ is, 2 and Hamming weight of $f(x_1, x_2) = x_1.x_2$ is 1.

Definition 6 *Balanced Boolean Function: If the Hamming weight of a Boolean function of n variables is 2^{n-1}, it is called a balanced Boolean function.*

Thus, $f(x_1, x_2) = x_1 \oplus x_2$ is balanced, while $f(x_1, x_2) = x_1.x_2$ is not balanced.

Definition 7 *Nonlinearity: Let, f be a Boolean function of variables, $x_1, x_2, \ldots x_n$ and A be the set of all affine functions in $x_1, x_2, \ldots x_n$. The minimum of the Hamming distances between f and the Boolean functions in A is the nonlinearity of f.*

Hence, nonlinearity of $f(x_1, x_2) = x_1.x_2$ is 1.

Definition 8 *Walsh Transform: Let $\bar{X} = (X_n, \ldots, X_1)$ and $\bar{\omega} = (\omega_1, \ldots \omega_n)$ both belong to $\{0,1\}^n$ and $\bar{X}.\bar{\omega} = X_n.\omega_1 \oplus \ldots X_1.\omega_n$. Let $f(\bar{X})$ be a Boolean function on n variables. Then the Walsh transform of $f(\bar{X})$ is a real valued function over $\{0,1\}^n$ that can be defined as $W_f(\bar{\omega}) = \Sigma_{\bar{X} \in \{0,1\}^n} (-1)^{f(\bar{X}) \oplus \bar{X}.\bar{\omega}}$. The Walsh transform is sometimes called the spectral distribution or simply the spectrum of a Boolean function.*

Definition 9 *Resiliency: A function $f(X_n \ldots X_1)$ is m-th order correlation immune (CI) iff its Walsh transform W_f satisfies $W_f(\bar{\omega}) = 0$; for $1 \leq wt(\bar{\omega}) \leq m$. Further, if f is balanced then $W_f(0) = 0$. Balanced m-th order correlation immune functions are called m-resilient functions. Thus, a function $f(X_n, \ldots, X_1)$ is m-resilient iff its Walsh transform W_f satisfies $W_f(\bar{\omega}) = 0$; for $0 \leq wt(\bar{\omega}) \leq m$.*

For example, resiliency of $f(x_1, x_2) = x_1 \oplus x_2$ is 1, but resiliency of $f(x_1, x_2) = x_1.x_2$ is 0.

Definition 10 *Algebraic Normal Form: Any Boolean function can be expressed as xor of conjunctions and a Boolean constant, True or False. This form of the Boolean function is called its Algebraic Normal Form (ANF).*

Every *Boolean function* can be expressed in *ANF*. As an example, $f(x_1, x_2, x_3) = x_1.x_2.x_3$ is in *ANF*, while $f(x_1, x_2, x_3) = (x_1 \oplus x_2).(x_2 \oplus x_3)$ is not in *ANF*. Its *ANF* representation is, $f(x_1, x_2, x_3) = x_1.x_2 \oplus x_1.x_3 \oplus x_2 \oplus x_2.x_3$.

Definition 11 *Algebraic Degree: The maximum number of literals in any conjunction of ANF of a Boolean function is called its degree. Ciphers expressible or conceivable as a Boolean function have algebraic degree which is the same as the degree of the ANF of the Boolean function.*

Thus, $f(x_1, x_2) = x_1 \oplus x_2 \oplus x_1.x_2$ has algebraic degree 2.

Next, we outline a test which has been developed to distinguish a given Boolean function from a truly random function.

2.3 d-Monomial Test

d-Monomial test is a statistical test for pseudorandomness proposed independently in [Saa] and [EJT]. It investigates the Boolean function representation of each output bit in terms of input bits. If a Boolean function of n Boolean variables is a good pseudorandom sequence generator, then it will have $\frac{1}{2} \binom{n}{d}$ d-degree monomials. A deviation will indicate non-randomness. For example, consider the function $f(x_1, x_2) = x_1 \oplus x_2$, it has 2, 1-degree monomials and 0, 2 degree monomial. It turns out that it has 1, 1-degree monomial more, hence it is expected to be non-pseudorandom. On the other hand $f(x_1, x_2) = x_1$ is expected to be a good pseudorandom generator.

In spite of its simplicity, this test gained huge appreciation in cryptography community. It proved to be a good tool in analyzing the degree of pseudorandomness of cryptographic systems. To the best of our knowledge, d-monomial test has not been applied to CA configurations previously. We explore different CA configurations under this test.

2.4 Specification of the Grain-128 Stream Cipher

Grain-128 is a hardware based stream cipher enlisted in the final list of the eStream [est] project. We briefly describe the specification of the Grain-128 stream cipher here. A detailed description may be found in [HJM].

The Grain-128 stream cipher consists of three main building blocks, namely, an NFSR, an LFSR and an output function $h(x)$(Fig. 2). The contents of the NFSR are denoted by $b_i, b_{i+1}, \ldots, b_{i+127}$ and the contents of the LFSR are denoted by, $s_i, s_{i+1}, \ldots, s_{i+127}$. The update function of the LFSR is given by,
$s_{i+128} = s_i + s_{i+7} + s_{i+38} + s_{i+70} + s_{i+81} + s_{i+96}$
The NFSR is updated by,

$$
\begin{aligned}
b_{i+128} & = s_i + b_i + b_{i+26} + b_{i+56} + b_{i+91} + b_{i+96} + b_{i+3}b_{i+67} + b_{i+11}b_{i+13} \\
& + b_{i+17}b_{i+18} + + b_{i+27}b_{i+59} + b_{i+40}b_{i+48} + b_{i+61}b_{i+65} + b_{i+68}b_{i+84}
\end{aligned}
$$

The NFSR and the LFSR together represent the internal state of the cipher. A nonlinear filter function h is defined with 2 input bits from the NFSR and 7 input bits from the LFSR. The function h is defined by:
$h = b_{i+12}s_{i+8} + s_{i+13}s_{i+20} + b_{i+95}s_{i+42} + s_{i+60}s_{i+79} + b_{i+12}b_{i+95}s_{i+95}$.
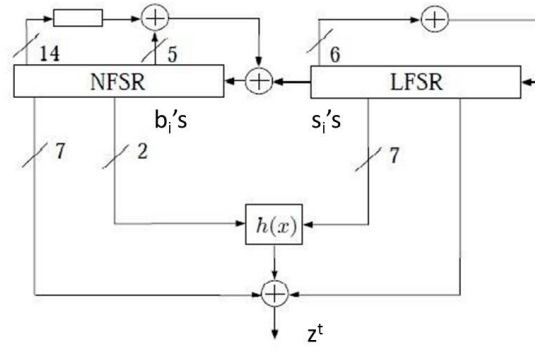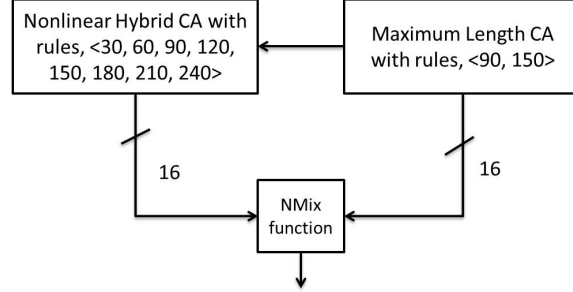
Fig. 2: Operation of Grain-128
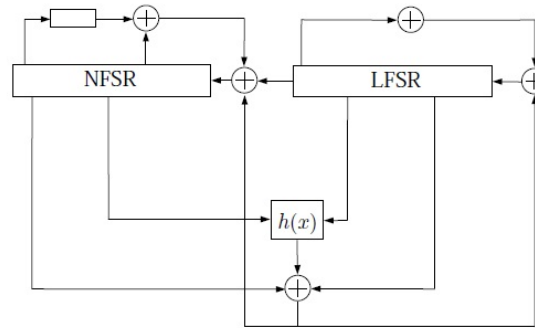
Fig. 3: Initialization of Grain-128

Fig. 4: Structure of NOCAS

The output function z^t is defined as,

$z^t = b_{t+2} + b_{t+15} + b_{t+36} + b_{t+45} + b_{t+64} + b_{t+73} + b_{t+89} + h + s_{t+93}$

An initialization phase is carried out before the cipher generates keystream bits. The 128 bit key, $k = (k_1, k_2, \ldots, k_{128})$ and the 96 bit initialization vector $IV = (IV_1, IV_2, \ldots, IV_{96})$ is loaded in the NFSR and the LFSR respectively as, $b_i = k_i, 1 \leq i \leq 128$ and $s_i = IV_i, 1 \leq i \leq 96$, rest of the LFSR bits, $(s_{97}, s_{98}, \ldots, s_{128})$ are loaded with 1. The cipher is run for 256 rounds without producing any keystream, during initialization the output function is fed back and xored with both the LFSR and the NFSR (Fig. 3).

3 NOCAS: A CA Based Stream Cipher

In the previous section, we have seen structure of Grain-128. The cipher is simple in design consisting of only a LFSR and an NFSR. It is a lightweight cipher with fast startup and high throughput. Unfortunately, a number of attacks have been mounted on it [DS11], [BCC+09], [KC11]. In [BCC+09], faults are injected in the LFSR to deduce full secret key in only 22 faults, while [KC11] induces faults in the NFSR to get back the secret key in maximum 256 faults. In this section we present the specification of the cipher NOCAS (Hybrid **NO**nlinear **CA** based **S**tream Cipher), with by replacing the LFSR with a linear maximum length CA and the NFSR with a hybrid nonlinear CA.

The building blocks of NOCAS are:

- A Hybrid Nonlinear CA of 128-bits with rules $< 30, 60, 90, 120, 150, 180, 210, 240 >$ repeated 16 times.

- A Linear Maximum Length CA of 128 bits with combinations of rules 90 and 150.

- The function NMix which is cryptographically suited nonlinear mixing function proposed in [BC09].

A block diagram of NOCAS is given in figure 4. Each of the building blocks are discussed in the following subsections.

Tab. 2: ANF of 3-nbd Rules used in Ruleset 5

Rule #	ANF	Linear?
30	$(x_2.x_3) \oplus x_1 \oplus x_2 \oplus x_3$	No
60	$x_1 \oplus x_2$	Yes
90	$x_1 \oplus x_3$	Yes
120	$x_1 \oplus (x_2.x_3)$	No
150	$x_1 \oplus x_2 \oplus x_3$	Yes
180	$x_1 \oplus x_2 \oplus (x_2.x_3)$	No
210	$x_1 \oplus x_3 \oplus (x_2.x_3)$	No
240	x_1	Yes

Tab. 3: d-**Monomial Characteristics of Hybrid Ruleset 5 CA [KMC10]**

	Number of nth degree terms			
Rules	*1*	*2*	*3*	*4*
Ideal	1,2,3	1,5,10	0,5,52	0,2,52
Ruleset 5	3,2,4	1,3,5	0,2,6	0,0,3

3.1 Hybrid Nonlinear CA

In [KMC10], a number of cellular automata have been synthesized and their cryptographic properties have been studied. The authors have identified six hybrid nonlinear hybrid CAs (ruleset 1 to 6) which are cryptographically robust. Among these rulesets we choose ruleset 5 i.e., $30, 60, 90, 120, 150, 180, 210, 240$. The algebraic normal form of the rules used in ruleset 5 is shown in table 2. We briefly discuss the cryptographic properties of ruleset 5 CA next. In our design, we use null-boundary ruleset 5 CA. Ruleset 5 CA consists of cells operating on rules $30, 60, 90, 120, 150, 180, 210, 240$ spaced alternatively. The nonlinear register of NOCAS is of 128 bits, hence, 16 such hybrid CA cells are repeated in the design.

In [KMC10] ruleset 5 is tested over three iterations for d-monomial test. We reproduce the result in table 3. Here, ruleset 5 is tested over three iterations for the cryptographic properties like, balancedness, nonlinearity, resiliency and algebraic degree (tab. 4).

It can be seen that over all the iterations, the CA generates balanced output and has a fast nonlinearity growth. Resiliency of the CA is constant, it has good algebraic degree which also increases fast with the iterations. Also, results of d-monomial test is satisfactory.

Tab. 4: Cryptographic Properties of Ruleset 5

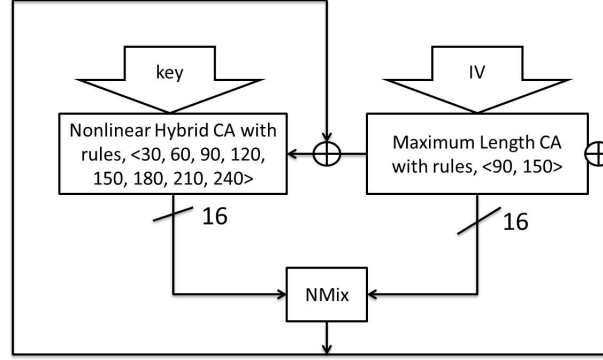| Iteration	Balancedness	Nonlinearity	Resiliency	Degree
1	Balanced	2	2	2
2	Balanced	8	2	3
3	Balanced	32	2	4

Fig. 5: Initialization of NOCAS

3.2   Maximum Length Linear CA

It is shown by researchers that $90, 150$ hybrid linear CA produces maximum length cycle for any CA length [CCNC]. In our design again we use null boundary 90, 150 hybrid CA, which is CA cells operating with rules 90 and 150 in such an arrangement so as to produce maximum length structure, and the end-cells are connected to nulls. It is known that such maximum length linear CA produces excellent pseudorandom sequences. The leftmost bit of this CA is fed to the rightmost bit position of hybrid CA in the structure of NOCAS. Clearly due to maximality of linear part and the design of NOCAS up to 2^{128} different states will be present NOCAS, which makes it possible to generate 2^{128} unique keystream bits.

3.3   NMix

NMix introduced in [BC09] is used to combine bits from hybrid CA and maximum length CA. The function possesses good cryptographic properties.

Definition 12 *For two n-bit inputs X and Y, the output Z given by NMix is defined as follows,*
$z_i = x_i \oplus y_i \oplus c_{i-1}$
$c_i = \oplus_{j=0}^{i} x_j y_j \oplus x_{i-1} x_i \oplus y_{i-1} y_i$
where, $0 \le i \le n - 1, c_{-1} = 0, x_{-1} = 0, y_{-1} = 0.$

We use 16 bits each from nonlinear part and linear parts of NOCAS as input to NMix and take the MSB as its output. Due to this design all 16 input bits from nonlinear and linear parts are mixed fully in the output. The output function is clearly a 32 variable bent function having degree 2, hence, providing high nonlinearity. The 16 input bits from nonlinear and linear parts are chosen as bits, $1, 10, 19, 28, 37, 46,$ $55, 64, 65, 74, 83, 92, 101, 110, 119, 128.$

3.4   Initialization

Key and initialization vector (IV) are input to the nonlinear and linear parts of the cipher in 128 bits each. So that $n_i = k_i, 1 \le i \le 128$, where n_i is the nonlinear register and k_i is the ith key bit, while, $l_i = v_i, 1 \le i \le 128$, where l_i is the linear register and v_i is the ith IV bit. Once, key and IV are setup in respective registers, the cipher is clocked for 64 cycles without producing any keystream and the keystream is XORed with both the MSB of nonlinear and linear registers fig. 5.

4   Security Analysis of NOCAS

In this section, we present the security analysis of NOCAS. We will see that the employment of hybrid nonlinear CA provides resistance against popular existing attacks.

- *Linear Cryptanalysis*: Nonlinearity and resiliency are the most important requirements for a cryptographic system. Good nonlinearity characteristics indicate that the cipher is expected to be safe against linear cryptanalysis and also from algebraic attacks. Table 5 shows the nonlinearity of NOCAS with pass of iteration. In only 4 cycles of operation nonlinearity of NOCAS reaches 12428. It can be noted that the growth rate of nonlinearity is very steep. As complexity of linear cryptanalysis is directly related to nonlinearity, it can be claimed that NOCAS is resistant against linear cryptanalysis.

- *Correlation Attack:* Good nonlinearity characteristics does not imply correlation immunity, ie, good nonlinear ciphers can display correlations among key, plain-texts and cipher-texts, which is the basis of correlation attack. Also, balancedness is an important factor to prevent correlation attack. Table 5 illustrates the balancedness of the NOCAS output bit with iterations. All the output bit expressions are balanced in the initial 4 iterations. Hence, the cryptographic property balancedness holds good for NOCAS. Table 5 also tabulates the resiliency of NOCAS output bit with iterations. It reveals that higher resiliency is achieved by NOCAS at much lower number of iterations. Due to the faster growth of resiliency of output bit of NOCAS and its balancedness, it is expected to show resistance against correlation attacks.

- *Algebraic Attacks*: Algebraic cryptanalysis is dependent on the algebraic degree of a cipher. The increase of number of nonlinear terms of a cipher also increase the attack complexity. Table 5 shows the growth of algebraic degree of the output bit of NOCAS with iterations, while table 6 shows d-monomial characteristics of NOCAS with iterations, which shows almost exponential growth in nonlinear terms. It can be observed that in NOCAS the algebraic degree increases linearly. The growth in number of terms in the resultant Boolean expression and the number of different degree terms in the output equation are both high. Considering table 6 once again, note that, at iteration 4 only the number of nonlinear terms in the expression of the output bit is more than 400, which is more than double the number of nonlinear terms at iteration 3, it can be expected that any attempt to linearize the expression for algebraic attack will have to deal with exponential number of nonlinear terms with pass of iterations. Hence, algebraic attacks are not expected to yield good result against NOCAS. Ciphers having large algebraic degrees are resistant against linearization and algebraic attacks. So, NOCAS is expected to be resistant to these attacks both in reduced round version and the full key-IV setup version.

- *Scan-based Side Channel Attack*: Scan-chain based attack works because of the invertibility of the states of the cipher. The same will not be possible for NOCAS because of the presence of non-invertible CA rule 30. Though rule 30 is partially reversible, presence of linear and nonlinear rules in the CA configuration reduces the probability of the reversion exponentially with iterations. Hence, scan-based side channel attack will not be successful on NOCAS.

- *Cube Attack/AIDA attack*: Till date the most successful attacks on stream ciphers were cube attack and dynamic cube attack [DS11]. This attack exploits the fact that the distribution of the d-degree

Tab. 5: Cryptographic Characteristics of NOCAS

Iteration	Balancedness	Nonlinearity	Algebraic Degree	Resiliency
1	Balanced	538	3	2
2	Balanced	1842	4	3
3	Balanced	5648	5	3
4	Balanced	12428	6	4

Tab. 6: d-monomial Test Result of NOCAS

Iteration	Deg.-1	Deg.-2	Deg.-3	Deg.-4	Deg.-5	Deg.-6
1	18	32	4	0	0	0
2	24	48	16	2	0	0
3	34	94	26	12	1	0
4	56	168	128	56	42	6

terms is far from ideal in d-monomial test. We tabulate in table 6 the d-monomial test values for the first 4 iterations of the output bit of NOCAS. This kind of distribution is expected to resist higher order differential attacks and distinguishers. The overall d-monomial characteristics of NOCAS is fairly good in view of the number of terms in middle degrees, presence of linear and highest degree terms. A large algebraic degree of a cipher will prevent the attack from practically being implemented. In case of NOCAS, the d-monomial test result is fairly good and the high algebraic degree growth rate is also an important factor in prevention of the attack on NOCAS. Hence, cube attack on NOCAS will not be successful on any reasonable number of rounds.

- *Fault Attack:* Fault attacks induce faults in the cipher registers and exploits the difference of faulty and fault-free cipher-text to deduce the secret key. In case of NOCAS, the design is such that it is difficult to produce linear or low-degree equations from faulty and fault-free cipher-texts. Hence, solving such a system is a hard problem. Therefore, fault attack is expected not to succeed against NOCAS.

5   Comparison of NOCAS with Grain-128

Both the ciphers, NOCAS and Grain-128 are synthesized on Xilinx 8.1 Vertex 4 FPGA. Table 7 compares the performances of NOCAS and Grain-128. The result shows that Grain-128 is hardware efficient than NOCAS while throughput is comparable. NOCAS achieves 4 times speedup in startup than Grain-128.

Tab. 7: Comparison of NOCAS and Grain-128

	No. of LUTs	Throughput	Setup
Grain-128	278	390 Mb/s	256 cycles
NOCAS	562	372 Mb/s	64 cycles

6   Conclusion

In the current paper, we have introduced a new stream cipher based on hybrid nonlinear CA called NO-CAS. The design produces fast initialization in only 64 cycles. We have analyzed the cryptographic properties like balancedness, nonlinearity, resiliency and algebraic degree of NOCAS, which show it is a cryptographically robust cipher. The d-monomial test also produce fairly good result against NOCAS. Finally, we have shown that NOCAS is expected to resistant against popularly known existing attacks. It achieves 4 times speedup in initialization than Grain-128.

References

[BBC$^+$]   Côme Berbain, Olivier Billet, Anne Canteaut, Nicolas Courtois, Henri Gilbert, Louis Goubin, Aline Gouget, Louis Granboulan, Cédric Lauradoux, Marine Minier, Thomas Pornin, and Herve Sibert. Sosemanuk, a fast software-oriented stream cipher. *eSTREAM, ECRYPT Stream Cipher Project*, 2006.

[BC09]   Jaydeb Bhowmik and Dipanwita Roy Chowdhury. Nmix : An Ideal Candidate for Key Mixing. *SecCrypt 2009*, pages 285–288, 2009.

[BCC$^+$]   Steve Babbage, Christophe De Canniere, Anne Canteaut, Carlos Cid, Henri Gilbert, Thomas Johansson, Matthew Parker, Bart Preneel, Vincent Rijmen, and Matthew Robshaw. The estream portfolio. *"http://www.ecrypt.eu.org/stream/portfolio.pdf"*.

[BCC$^+$09]   Alexandre Berzati, Cecile Canovas, Guilhem Castagnos, Blandine Debraize, Louis Goubin, Aline Gouget, Pascal Paillier, and Stephanie Salgado. Fault analysis of grain-128. *Hardware-Oriented Security and Trust, IEEE International Workshop on*, 0:7–14, 2009.

[BD]   Steve Babbage and Matthew Dodd. The stream cipher mickey 2.0. *eSTREAM, ECRYPT Stream Cipher Project*, 2006.

[Ber]   Daniel J. Bernstein. Salsa20. *eSTREAM, ECRYPT Stream Cipher Project*, 2006.

[BVCZ]   Martin Boesgaard, Mette Vesterager, Thomas Christensen, and Erik Zenner. The stream cipher rabbit. *eSTREAM, ECRYPT Stream Cipher Project*, 2006.

[CCNC]   P. Pal Chaudhuri, D. Roy Chowdhury, S. Nandi, and S. Chattopadhyay. CA and Its Applications: A Brief Survey, Additive Cellular Automata - Theory and Applications vol.-1, pages 6-25. *eSTREAM, ECRYPT Stream Cipher Project*, 1997.

[CP]   Christophe De Canniere and Bart Preneel. Trivium specifications. *eSTREAM, ECRYPT Stream Cipher Project*, 2006.

[DS11]   Ita Dinur and Adi Shamir. Dynamic Cube Attack on Full Grain-128. *ePrint Cryptology Archive*, 2011.

[EJT]   H. Englund, T. Johansson, and MS Turan. A Framework for Chosen IV Statistical Analysis of Stream Ciphers. *Progress in Cryptology - INDOCRYPT*, 2007:268–281.

[est]   The estream project. *"http://www.ecrypt.eu.org/stream/"*.

[HJM]   Martin Hell, Thomas Johansson, and Willi Meier. A stream cipher proposal: Grain-128. *eSTREAM, ECRYPT Stream Cipher Project*, 2006.

[KC11]   Sandip Karmakar and Dipanwita Roy Chowdhury. Fault Analysis of Grain-128 by Targeting NFSR. *AfricaCrypt 2011*, 2011.

[KMC10]   Sandip Karmakar, Debdeep Mukhopadhyay, and Dipanwita Roy Chowdhury. d-monomial Tests on Cellular Automata for Cryptographic Design. *ACRI 2010*, 2010.

[MS91]   Meier and Staffelbach. Analysis of Pseudo Random Sequences Generated by Cellular Automata. *"EUROCRYPT: Advances in Cryptology: Proceedings of EUROCRYPT"*, 1991.

[Saa]   Markku-Juhani O. Saarinen. Chosen IV Statistical Attacks on eStream Stream Ciphers. *http://www.ecrypt.eu.org/stream*.

[Wola]   Wolfram. Cryptography with Cellular Automata. *CRYPTO: Proceedings of Crypto*, 1985.

[Wolb]   S. Wolfram. Random Sequence Generation by Cellular Automata. *Advances in Applied Mathematics, vol.-7, pages 123-169*.

[Wu]   Hongjun Wu. Stream cipher hc-128. *eSTREAM, ECRYPT Stream Cipher Project*, 2006.

Cell damage from radiation-induced bystander effects for different cell densities simulated by cellular automata

Sincler Peixoto de Meireles[1][†] and Adriano Márcio dos Santos[1] and Maria Eugênia Silva Nunes[2] and Suely Epsztein Grynberg[1]

[1]*Centro de Desenvolvimento da Tecnologia Nuclear (CDTN/CNEN) - Av. Presidente Antonio Carlos 6627, 31270-901, Belo Horizonte, Minas Gerais, Brasil*
[2]*Universidade Federal de Ouro Preto (UFOP) - Rua Diogo de Vascomcelos, 122, 35400-000, Ouro Preto , Minas Gerais, Brasil*

During recent years, there has been a shift from an approach focused entirely on DNA as the main target of ionizing radiation to a vision that considers complex signaling pathways in cells and among cells within tissues. Several newly recognized responses were classified as the so-called non-target responses in which the biological effects are not directly related to the amount of energy deposited in the DNA of cells that were traversed by radiation. In 1992 the bystander effect was described referring to a series of responses such as death, chromosomal instability or other abnormalities that occur in non-irradiated cells that came into contact with irradiated cells or medium from irradiated cells. In this work, we have developed a mathematical model via cellular automata, to quantify cell death induced by the bystander effect. The model is based on experiments with irradiated cells conditioned medium (ICCM) which suggests that irradiated cells secrete molecules in the medium that are capable of damaging other cells. The computational model consists of two-dimensional cellular automata which is able to simulate the transmission of bystander signals via extrinsic route and via Gap junctions. The model has been validated by experimental results in the literature. The time evolution of the effect and the dose-response curves were obtained in good accordance to them. Simulations were conducted for different values of bystander and irradiated cell densities with constant dose. From this work, we have obtained a relationship between cell density and effect.

Keywords: Automata Cellular, Bystander Effect, Computer Simulation, Monte Carlo

1   Introduction

Several radiobiological studies over the past decade have profoundly challenged the dogma of classical radiobiology by which radiation effects would only be observed in cells that have undergone irradiation, or their descendants, through genetic damage produced directly by energy deposition in DNA. Currently,

there is compelling evidence suggesting that when a cell population is exposed to ionizing radiation, biological effects occur in a greater proportion compared to cells that have been actually irradiated **?**. Microbeam studies have shown unequivocally that non-hit cells respond to changes in gene expression, micronuclei formation, chromosomal aberrations, mutations and cell death **?**. Later experiments with irradiated cells conditioned medium (ICCM) were also able to confirm this effect and suggest its action by a factor released by irradiated cells that somehow communicates with their neighbors **?**. This phenomenon has been termed radiation-induced bystander effect. From the description of the bystander effect there were several attempts to establish models to understand it better. An attempt was made by **?** with the BaD (bystander and direct) model. It reviewed the radiobiological damage in directly irradiated cells and in the bystander cells. In the following years he extended the model to study the effects on human carcinogenesis **??**. **?** constructed a model for broad and microbeam, very similar to the BaD model, but considering that differentiating cell damage originates from specific signs of protein character. This bystander signal diffuses into the medium by Brownian motion and may cause cell inactivation, cell death and oncogenic transformation. Later, in a more extensive analysis of data, **?** showed that the model adjustment could be improved if a long latency period was considered (five or six years). The adjustment of the latter model is equivalent to a relative risk model with linear fit for age at exposure and attained age. The following year, a new stochastic model was developed using the Monte Carlo technique **?**, taking into account the spatial location, cell death and repopulation. The dose of ionizing radiation and time-response of this model were explored. Based on a model of tumor growth and direct irradiation **?** develop a model where: hyper-sensitivity at low doses and the bystander effect are considered. A cellular automata was used to simulate the diffusion of glucose and to describe cell growth. In this model the cell cycle phases were not taken into account in relation to the effects of radiation, taking advantage of its phases only to describe the cellular multiplication. Another proposed model describes the bystander effect as a result of two distinct processes: trigger signal output from irradiated cells and bystander cell response **?**. In this model, cells that received signals may have late effects and proliferate. The model emphasizes the dependence of the dose for the occurrence of the effects, and also suggests that increasing the quantity of the medium should cause approximately the same effect as a moderate reduction of the fraction of irradiated cells. In this paper, a computational model was written for the study of Radiation-Induced Bystander Effects based on harvesting medium experiments. This model focuses on reception and reemission of bystander signals by secondary sources, considering factors of signal activity loss and repair mechanisms actions.

2 Materials and Methods

2.1 Computational Model

The model consists of a two-dimensional cellular automata, consisted of two overlapping networks, where the first represents the cellular matrix and the second the medium in which cells are immersed. This model is able to simulate the transmission of bystander signals via the intracellular environment, and via cell junctions. We adopted the use of square sites, according to the relation of Moore neighborhood, where neighbors are considered the eight adjacent sites to the site in question. The sites can take the following states: healthy cell, the cell which received the bystander signal, dead cell, and absence of cell (empty space). The state transitions of cells can occur not only due to the bystander effect. Cells can also die because of increased competition for space and nutrients, or even multiply. At the start of the simulation cells are distributed randomly in the network, as well as the bystander signals. It is possible to use different geometries, varying the density of cell culture and also varying the lines to be simulated. The

state transitions of the cells are illustrated in Fig. 1. After irradiation (A), which lasts a time t_0, the cells stay at rest for a time t_1 after which (B) the medium transfer is carried out. The time in which the effects are measured after transfer of medium is called t_2. The number of signals is obtained through probability
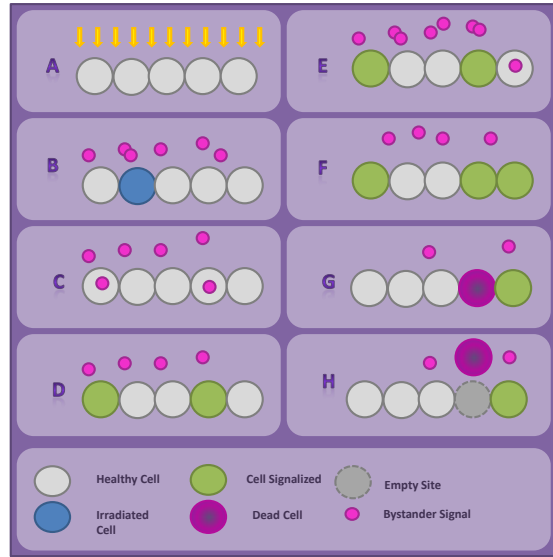
**Figure 1:** Changes in the model allowed state.

functions depending on the dose received by the donor culture. Each of the signals generated can lose its ability to interact over time. The half life for the bystander signal has not been determined, but the signal is still active for more than 60h **?**. The signals move freely through the medium, and their motion was simulated using the Monte Carlo technique (MC). They can interact with cells of the receiving culture medium from irradiated cells (C). When a signal interacts with a cell it disappears from the network and the cell becomes a cell signalized (D). The Monte Carlo technique is also employed in the state changes of cells. As suggested by **?**, a cell that received the signal can become a secondary source that triggers a chain reaction. The cells receiving the signal can generate new bystander signals and transmit them to neighboring cells by cell junctions (F), or release them in the intracellular medium (E). At the end of each time step of simulation it is possible to observe the number of signals generated and absorbed by the cells. For a cell, the greater the number of neighboring cells signaled, the higher the probability of receiving a signal. A cell that received the signal on bystander can evolve into two situations over time (G). In the first, the cell can return to its original state, admitting that it has the ability to eliminate or inactivate the bystander signal, or it can also return to its original state to repair the damage caused by the signal. In the second situation the cell may die because of the damage caused by the signal. Moreover, even in populations that induce these effects, not all cells respond to the signal and show the effect **?**. A dead cell can come off the culture plate (H), freeing up space on the network for a new cell takes its place.

2.2   Computational Resources

The model is based on experiments carried out with ICCM **?**. The logic was implemented through a program written in C language. The random number generator UNI was chosen for the simulations by having passed all the tests in Marsaglia's DIEHARD find in **?**. The simulations were performed on an XPS 8300 Intel Core ^{TM} i7 quad-core processors with Windows operating system 7. Data were analyzed using Origin v7.5 software and Microsoft Office Excel 2007. The images obtained in the simulation were generated by RasTop 2.2.

3   Results and discussions

To validate the computer model, MC simulations were performed to obtain the survivor number as a function of dose (Fig. 2). The simulations were performed using the parameters of dose, cell density and times used by **?**. Different seeds of random number generator UNI were used to obtain an mean value of the simulation results. Nine simulations were done for each dose value. The error bars are not displayed in the figure because of the accuracy of the simulation(their sizes are of the same order of the dots' sizes). Comparing the results from the simulations with the experimental data shown in Fig. 2, there is a very
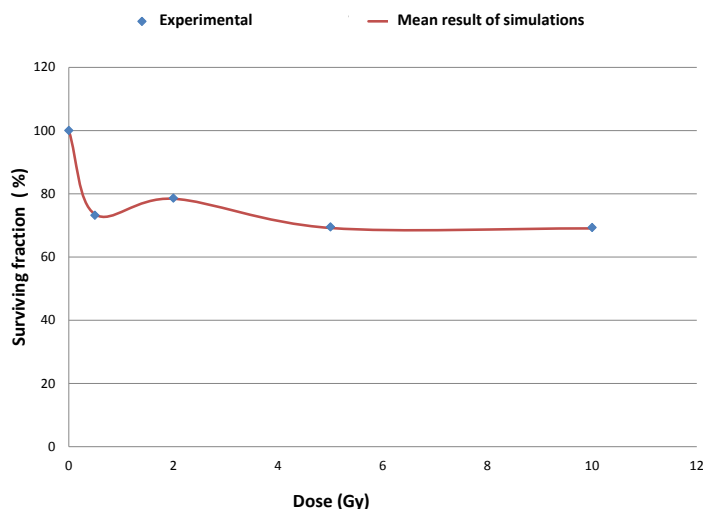

Figure 2: Dose-response curve for the bystander culture. Error bars are not shown because of their sizes being of the same order of the dots sizes

good agreement between them, with a standard deviation less than ± 0.3 (Table 1). This indicates that the developed model is able to reproduce with a good range of security, the experimental results presented in the literature.

Table 1. Standard deviation of the values found in the simulation

| Experimental Results(%)	Mean results of simulation (%)	Standard Deviation
100	100	0.0
73.2	73,5	0.2
78.6	78,5	0.2
69.5	69,2	0.3
69.3	69,1	0.2

The effect of conditioned medium over the non irradiated cell culture increases with the density of the irradiated culture **?**. Simulations were performed keeping the number of bystander cells in culture constant and varying density of irradiated cells. Nine simulations were done for each density value. The data are shown in Fig. 3.
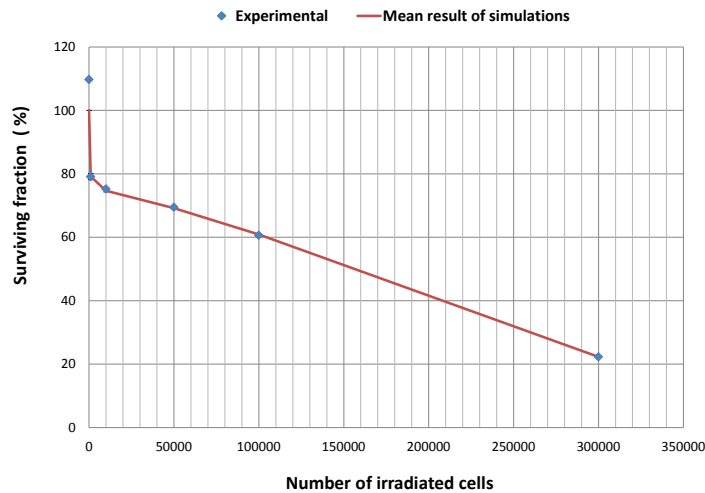
Figure 3: Graph comparing the surviving fraction obtained in the simulation to the experimental results described in the literature. Error bars are not shown because of their small sizes compared to the dots sizes.

No experimental data were found in the literature of the bystander effect for different densities of non-hit cells. To estimate this behavior, new simulations were performed fixing the number of irradiated cells and varying the bystander cells density. The results are shown in Fig. 4. The increasing density of bystander cells in culture generates only a small variation in the effect, where the higher the cells density, the greater the surviving fraction, noting an effect opposite to the variation of the irradiated cells density. The de-

crease of the effect with increasing bystander cell density is consistent because the number of signals per cell of the network becomes smaller .
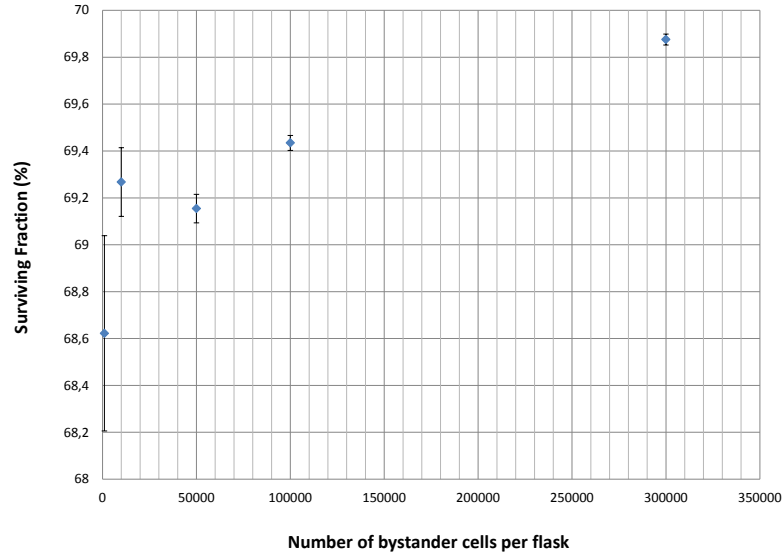
Figure 4: Fraction of surviving cells for different densities of cells in the bystander culture with the errors bars.

4   Conclusions

The model presented in this work can be a tool in understanding the bystander effect, since it agreed with the data documented in the literature. It can be used to simulate the behavior of cell lines for different cell densities, different cell doses and other different parameters found in the literature. The model also shows a behavior that was not experimentally explored yet which is the increase of the surviving fraction as a function of the number of bystander cells per flask, showing a decrease of the bystander effect with the increase of the bystander culture density.

Acknowledgements

Product decomposition for surjective 2-block NCCA

Felipe García-Ramos[1][†]

[1]*University of British Columbia, Vancouver, Canada*

In this paper we define products of one-dimensional Number Conserving Cellular Automata (NCCA) and show that surjective NCCA with 2 blocks (i.e radius 1/2) can always be represented as products of shifts and identites. In particular, this shows that surjective 2-block NCCA are injective.

Keywords: Discrete dynamical systems, cellular automata, number conserving cellular automata, conservation laws, characterization of surjective NCCA

1 Introduction

It is known that injective Cellular Automata (CA) are surjective. In general, the converse is not true, and there are many algebraic CA counterexamples. However, there are interesting subclasses where this might be true. For example, if a surjective CA has entropy 0 then it is almost injective (Moothathu (2011)) and it is not known if it is actually injective. The author believes there are some sub-classes of potential preserving CA, including Number Conserving CA (NCCA), where there are no surjective but not injective CA.

The subclass of NCCA, besides providing interesting mathematical structure, is used for discrete models in scientific disciplines where one simulates systems governed by conservation laws of mass or energy. Many papers have been published on traffic models using NCCA (for example see Maerivoet and Moor (2005)).

The Moore-Myhill theorem says that a CA is surjective iff it is injective on homoclinic classes. Actually, it is easy to see that a NCCA is surjective iff it is bijective on homoclinic classes. This suggests that there might be a closer relationship between surjective NCCA and injective NCCA. So far, it is known that surjective NCCA have dense periodic points (Formenti and Grange (2003)). If it turns out that surjective NCCA are injective we would recover this result, since for bijective CA periodic points are dense.

---

2   Definitions and classical results

Let $\mathcal{A}$ be a finite set, which will sometimes be referred to as the alphabet. We define the full $\mathcal{A}$-shift as the space of bi-sequences $\mathcal{A}^{\mathbb{Z}}$. We will endow this space with the Cantor (product) topology. If $\omega \in \mathcal{A}^{\mathbb{Z}}$, we denote $(\omega)_i$ as the ith coordinate of point x. We will use $\sigma_R : \mathcal{A}^{\mathbb{Z}} \to \mathcal{A}^{\mathbb{Z}}$ as the right shift map, i.e. the map that satisfies $(\omega)_i = (\sigma_R(\omega))_{i+1}$ for all $\omega \in \mathcal{A}^{\mathbb{Z}}$ and $i \in \mathbb{Z}$.

Definition 1 *A **cellular automaton** (CA) is a continuous map $\phi(\cdot) : \mathcal{A}^{\mathbb{Z}} \to \mathcal{A}^{\mathbb{Z}}$ that commutes with the shift.*

Theorem 2 (Curtis-Hedlund-Lyndon) *Hedlund (1969) Let $\phi(\cdot) : \mathcal{A}^{\mathbb{Z}} \to \mathcal{A}^{\mathbb{Z}}$. The map ϕ is a CA iff there exist two non-negative integers L and R (which represent the left and right radius), and a function $\phi[\cdot] : \mathcal{A}^{L+R+1} \to \mathcal{A}$, such that $(\phi(\omega))_i = \phi\left[(\omega)_{i-m}(\omega)_{i-m+1}...(\omega)_i...(\omega)_{i+a}\right]$ (note the use of ( ), and [ ] to distinguish between the two functions that are related).*

   We say $L + R + 1$ is the neighbourhood size of ϕ.

Definition 3 *In this paper a **2-block CA** ϕ (also known as CA with radius 1/2) is a map with $L = 1$ and $R = 0$.*

   For example the right shift is a 2-block CA. The reader will see that all the results are analogous for $L = 0$ and $R = 1$.
   We say two points in $\mathcal{A}^{\mathbb{Z}}$ are *equivalent*, if they differ only on finitely many coordinates. The *homoclinic class* of a point ω is the set of points equivalent to ω.

Theorem 4 (Moore-Myhill) *Moore (1963) Myhill (1963) Let ϕ be a CA. Then ϕ is surjective iff ϕ is injective when restricted to homoclinic classes iff ϕ is injective when restricted to one homoclinic class.*

Definition 5 *We say a cellular automaton $\phi : [0...A)^{\mathbb{Z}} \to [0...A)^{\mathbb{Z}}$ is **number conserving**, also denoted as **NCCA**, if for every point ω in the homoclinic class of 0^{∞} we have that $\sum_{i \in \mathbb{Z}}(\phi(\omega))_i < \infty$, and*

$$\sum_{i \in \mathbb{Z}}(\phi(\omega))_i = \sum_{i \in \mathbb{Z}}(\omega)_i.$$

   The following result is the particular case for 2-block CA of a general result by Hattori and Takesue (1991), which was used by Boccara and Fuks (2002) to characterize NCCA. We provide a proof of this weaker result for completeness.

Proposition 6 *Let $\phi : [0...a]^{\mathbb{Z}} \to [0...a]^{\mathbb{Z}}$ be a 2-block CA. Then ϕ is a NCCA iff*

$$\phi[pq] = q + \phi[p0] - \phi[q0]. \tag{1}$$

Proof: We have that $\phi(0^{\infty}p0^{\infty}) = 0^{\infty}\phi[0p]\phi[p0]0^{\infty}$. This means that

$$p = \phi[0p] + \phi[p0]. \tag{2}$$

Similarly consider the image of the point $\phi(0^\infty pq0^\infty) = 0^\infty abc0^\infty$. We have that $a = \phi\,[0p]$, $b = \phi\,[pq]$, and $c = \phi\,[q0]$. Since ϕ is a NCCA we have that

$$p + q = a + b + c = \phi\,[0p] + \phi\,[pq] + \phi\,[q0]. \tag{3}$$

Combining (3) and (2) we get (1).

Conversely suppose ϕ satisfies (1). Let ω be a point in the homoclinic class of 0^∞. This means there exist j and k, such that $(\omega)_i = 0$ for $i > k$ and $i < j$. So we get that

$$\sum_{i \in \mathbb{Z}} (\phi(\omega))_i = \phi\,[0(\omega)_j] + \sum_{i=j}^{k-1} \phi\,[(\omega)_i(\omega)_{i+1}] + \phi\,[(\omega)_k 0]$$

$$= (\omega)_j - \phi\,[(\omega)_j 0] + \sum_{i=j}^{k-1} ((\omega)_{i+1} + \phi\,[(\omega)_i 0] - \phi\,[(\omega)_{i+1} 0]) + \phi\,[(\omega)_k 0]$$

$$= \sum_{i=j}^{k} (\omega)_i.$$

$\square$

This result tells us that a 2-block NCCA is uniquely determined by the values of $\phi\,[x0]$ and if $p = q = 0$ we get $\phi\,[00] = 0$.

Example 7 *The reader can check that $\phi : [0,1,2]^{\mathbb{Z}} \to [0,1,2]^{\mathbb{Z}}$, with $\phi\,[10] = 0$ and $\phi\,[20] = 1$, is a well defined (non-surjective) NCCA but there is no 2-block NCCA $\phi : [0...3]^{\mathbb{Z}} \to [0...3]^{\mathbb{Z}}$ with $\phi\,[10] = 1$ and $\phi\,[20] = 0$, because the image of the point $0^\infty 120^\infty$ cannot have sum equal to 3.*

In general a specification $\phi\,[\cdot 0]$ is defines a 2-block NCCA $\phi : [0...a]^{\mathbb{Z}} \to [0...a]^{\mathbb{Z}}$ via (1) iff

$$0 \leq q + \phi\,[p0] - \phi\,[q0] \leq a \quad \forall p, q \in [0...a].$$

We will use the following result several times.

Theorem 8 *A NCCA is surjective iff it is bijective on the homoclinic class of 0^∞.*

Proof: Simply apply Theorem 4 and note that the image and preimage of the homoclinic class of 0^∞ under ϕ is in the homoclinic class of 0^∞. $\square$

We would like to characterize NCCA in a more concrete way, at least under some special assumptions. In this paper we will do so for surjective 2-block NCCA.

We will first show how to represent the product of 2-block NCCA's as a 2-block NCCA. We will establish some properties of the product and finally show that all surjective 2-block NCCA can be represented as the product of shift and identity maps.

3 Products of NCCA

For easier notation we define $A := a + 1$, $B := b + 1$, and $[0...n) := [0...n - 1]$. We will use A and B or a and b, depending on which one gives an easier notation.

Let $\phi : [0...A)^{\mathbb{Z}} \to [0...A)^{\mathbb{Z}}$ and $\psi : [0...B)^{\mathbb{Z}} \to [0...B)^{\mathbb{Z}}$ be two 2-block codes. Consider the function $F(p_1, p_2) = p_2 + Bp_1$, where $p_1 \in [0...A)$ and $p_2 \in [0...B)$. This function is a bijection between $[0...A) \times [0...B)$ and $[0...AB)$. Furthermore, it satisfies

$$F(p_1 + p_1', p_2 + p_2') = F(p_1, p_2) + F(p_1', p_2'),$$

for $p_1 + p_1' \in [0, a]$ and $p_2 + p_2' \in [0, b]$. Now, for all $p, q \in [0...AB)$, we can define $\phi \times \psi \, [pq] = F(\phi \, [F_1(p)F_1(q)], \psi \, [F_2(p)F_2(q)])$, where F_1 and F_2 are the coordinates of the inverse, i.e.

$$\phi \times \psi \left[\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} \right] = \begin{pmatrix} \phi \, [p_1 \, q_1] \\ \psi \, [p_2 \, q_2] \end{pmatrix},$$

where $\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = F(\alpha, \beta)$.

Now we note that if ϕ and ψ are number conserving, then the product $\chi = \phi \times \psi$ is also number conserving since

$$\chi \left[\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} \right] = \begin{pmatrix} \phi \, [p_1 \, q_1] \\ \psi \, [p_2 \, q_2] \end{pmatrix}$$

$$= \begin{pmatrix} q_1 + \phi \, [p_1 \, 0] - \phi \, [q_1 \, 0] \\ q_2 + \psi \, [p_2 \, 0] - \psi \, [q_2 \, 0] \end{pmatrix}$$

$$= \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} + \begin{pmatrix} \phi \, [p_1 \, 0] \\ \psi \, [p_2 \, 0] \end{pmatrix} - \begin{pmatrix} \phi \, [q_1 \, 0] \\ \psi \, [q_2 \, 0] \end{pmatrix}$$

$$= \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} + \chi \left[\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right] - \chi \left[\begin{pmatrix} q_1 \\ q_2 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right],$$

and therefore χ satisfies equation (1).

Since F is not symmetric in general, $\phi \times \psi$ need not be the same as $\psi \times \phi$.

Even though we can make products of any two 2-block codes, we will mainly be interested in products of shifts and identities (which will be denoted as σ_R and Id respectively).

If $\phi = Id$, and $\chi = \phi \times \psi$ then

$$\chi \left[\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right] = \begin{pmatrix} \phi \, [p_1 \, 0] \\ \psi \, [p_2 \, 0] \end{pmatrix} = \begin{pmatrix} 0 \\ \psi \, [p_2 \, 0] \end{pmatrix} = \psi(p_2 \, 0). \tag{4}$$

If $\phi = \sigma_R$, then

$$\chi \left[\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right] = \begin{pmatrix} \phi \, [p_1 \, 0] \\ \psi \, [p_2 \, 0] \end{pmatrix} = \begin{pmatrix} p_1 \\ \psi \, [p_2 \, 0] \end{pmatrix} = \psi(p_2 \, 0) + p_1(b + 1). \tag{5}$$

Example 9 *Let $\sigma_R : [0...3)^{\mathbb{Z}} \to [0...3)^{\mathbb{Z}}$, $Id : [0...4)^{\mathbb{Z}} \to [0...4)^{\mathbb{Z}}$, and $\chi = \sigma_R \times Id : [0...(3 \cdot 4))^{\mathbb{Z}} \to [0...(3 \cdot 4))^{\mathbb{Z}}$. For every $x \in [0...12)$ there exists $p_1 \in [0...3)$ and $p_2 \in [0...4)$ such that $x = p_2 + 4p_1$, where $p_1 \in [0...3)$ and $p_2 \in [0...4)$. Hence $\chi[x0] = p_1$. In a table it looks like this.*

x	0	1	2	3	4	5	6	7	8	9	10	11
$\chi[x0]$	0	0	0	0	1	1	1	1	2	2	2	2

If we take instead $Id \times \sigma_R$ we get the following table.

x	0	1	2	3	4	5	6	7	8	9	10	11
$\chi[x0]$	0	1	2	0	1	2	0	1	2	0	1	2

The following lemma describes an important property when one of the factors is a shift or an identity. The proof uses formulas (4) and (5) and is left to the reader.

Lemma 10 *Let $\phi = Id$. We have that $\chi[x0] = x$ iff $x < B$ and $\psi[x0] = x$; and $\chi[x0] = 0$ iff there exists $n \in [0...A)$ such that $x = y + nB$ and $\psi[y0] = 0$.*
Similarly let $\phi = \sigma_R$. We also have that $\chi[x0] = 0$ iff $x < B$ and $\psi[x0] = 0$; and $\chi[x0] = x$ iff there exists $n \in [0...A)$ and $y \in [0...B)$ such that $x = y + nB$ and $\psi[y\,0] = y$.

Definition 11 *Let ϕ be a 2-block cellular automata. We say ϕ is a **shift-identity product cellular automa (SIPCA)** if $\phi = \phi_n \times \cdots \times \phi_1$, where $\phi_i = Id$ for all even $i's$, and $\phi_i = \sigma_R$ for all odd (or vice versa).*

Notation 12 *We will denote $f_\phi(x) = \phi[x0]$. Notice from (1) that the function $f_\phi(x) = \phi[x0]$ completely determines ϕ.*

NCCA arise in the context of particle preserving maps. People have shown (Boccara and Fuks (2002), Pivato (2002), Moreira et al. (2004) for 1-d and recently Kari and Taati (2008) for 2-d) that it is equivalent to give a NCCA as a compatible list of *particle displacement representations*. In the case of 2-block NCCA, the particle displacement representations are given by $f_\phi(x)$, which represent how many particles move to the right when you see x particles in a certain position.

The product of two shifts is a shift, and the product of two identities is an identity. Thus all products of right-shifts and identities are SIPCA. In general if our alphabet is $[0...A)$, for every way that we can write $A = A_n \cdot A_{n-1} \cdots A_1$ (with $A_i \in \mathbb{N}$), we have two SIPCA with that alphabet. We can take $\chi = \phi_n \times \cdots \times \phi_1$ with $\phi_i : [0...A_i)^{\mathbb{Z}} \to [0...A_i)^{\mathbb{Z}}$ alternating between shifts and identities with either $\phi_1 = Id$ or $\phi_1 = \sigma_R$.

It's useful to describe the graph of f_χ. Suppose $\phi_1 : [0...A)^{\mathbb{Z}} \to [0...A)^{\mathbb{Z}}$ is any 2-block NCCA and $\phi_2 : [0...3)^{\mathbb{Z}} \to [0...3)^{\mathbb{Z}}$. Figure 1 represents the graph of f_χ when $\phi_2 = Id$, and Figure 2 when $\phi_2 = \sigma_R$. Figure 3 and 4 represent the graph of f_χ, where χ is a SIPCA and $\phi_1 = Id$ and σ_R respectively.

4 Main result

The main goal of this paper is to prove the following result.

Theorem 13 *All surjective 2-block NCCA are SIPCA.*

Fig. 1: f_ϕ when $\phi_2 = Id$.
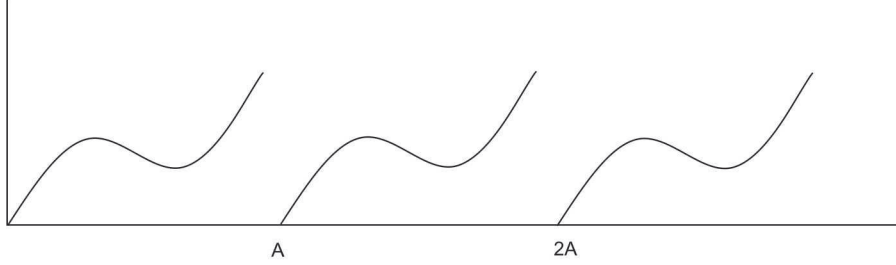
Definition 14 *Let* $\chi : [0...A)^{\mathbb{Z}} \to [0...A)^{\mathbb{Z}}$ *be a SIPCA. We say* $t \leq A$ *is a **transition point** if* $\chi \mid_{[0...t)^{\mathbb{Z}}}$ *is a SIPCA.*

Example 15 *Let* $Id : [0...A)^{\mathbb{Z}} \to [0...A)^{\mathbb{Z}}$ *and* $\sigma_R : [0...B)^{\mathbb{Z}} \to [0...B)^{\mathbb{Z}}$. *The transition points of* $Id \times \sigma_R$ *are* $[0...B) \cup \{B, 2B, ..., AB\}$.

Example 16 *Let* $Id : [0...A)^{\mathbb{Z}} \to [0...A)^{\mathbb{Z}}$ *and* $\phi : [0...B)^{\mathbb{Z}} \to [0...B)^{\mathbb{Z}}$ *be a SIPCA. The transition points of* $Id \times \phi$ *are the transition points of* ϕ *and* $\{B, 2B, ..., AB\}$.

In general if $\chi = \phi_n \times \cdots \times \phi_1$ is a SIPCA with $\phi_i : [0...A_i)^{\mathbb{Z}} \to [0...A_i)^{\mathbb{Z}}$, then t is a transition point iff $t = \left(\prod_{i=1}^{j-1} A_i \right) B_j$, where $B_j \in [1...A_j)$.

**Lemma 17** *Let* $\chi = \phi_n \times \cdots \times \phi_1 : [0...A)^{\mathbb{Z}} \to [0...A)^{\mathbb{Z}}$ *with* $\phi_i : [0...A_i)^{\mathbb{Z}} \to [0...A_i)^{\mathbb{Z}}$ *be a SIPCA and* x *a non-transition point with* $f_\chi(x) = x$ *(or 0). If* $t = \left(\prod_{i=1}^{j-1} A_i \right) B_j$ *is the previous transition point then* $f_\chi(t) = t$ *(or 0),* $f_\chi(\prod_{i=1}^{j-2} A_i) = 0$ *(or* $\prod_{i=1}^{j-2} A_i$*), and* $x - t < \prod_{i=1}^{j-2} A_i$.

Proof: Suppose x is not a transition point and $f_\chi(x) = x$. Let $t < x$ be the previous transition point, so $t = \left(\prod_{i=1}^{j-1} A_i \right) B_j$ and $\phi_j = \sigma_R$ (see Lemma 10). Let $y_1 = x - t < \prod_{i=1}^{j-1} A_i$. We have that $f_\chi(y_1) = y_1$, but since $\chi \mid_{\left[0... \prod_{i=1}^{j-1} A_i \right)^{\mathbb{Z}}} = \phi_{j-1} \times \cdots \times \phi_1$ and $\phi_{j-1} = Id$, we also know that $y_1 < \prod_{i=1}^{j-2} A_i$ (again by Lemma 10).

The other case is analogous. $\square$

We can characterize transition points as follows.

Proposition 18 *Let* $\chi = \phi_n \times \cdots \times \phi_1 : [0...A)^{\mathbb{Z}} \to [0...A)^{\mathbb{Z}}$ *with* $\phi_i : [0...A_i)^{\mathbb{Z}} \to [0...A_i)^{\mathbb{Z}}$ *be a SIPCA. Then* $x \in [0...A)^{\mathbb{Z}}$ *is a transition point iff* $\chi \mid_{[0...x)^{\mathbb{Z}}}$ *is a surjective NCCA.*

Fig. 2: f_ϕ when $\phi_2 = \sigma_R$.
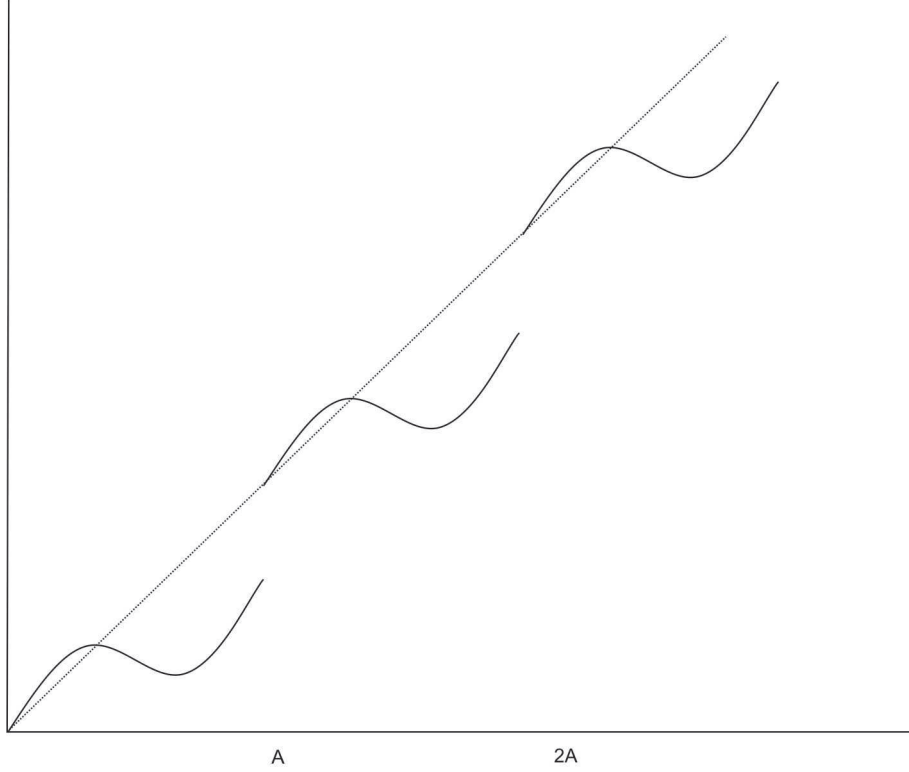
Proof: If $\chi \mid_{[0...x)^z}$ is a SIPCA then it is clearly surjective.

For the converse, first we will see that if $\chi \mid_{[0...x)^z}$ is a surjective NCCA then $f_\chi(x)$ has to be 0 or x. Suppose it's not. Since χ is bijective, there exists only one pair $p, q \leq x$ such that $\chi(0^\infty pq0^\infty) = 0^\infty x0^\infty$. Note that neither p nor q can be x or 0 because $f_\chi(x)$ is not 0 or x. This means that $p, q < x$, so the image of $\chi \mid_{[0...x)^z}$ would not be contained in $[0...x)^{\mathbb{Z}}$.

Now suppose x is not a transition point with $f_\chi(x) = x$ and consider the previous transition point

$$t = \left(\prod_{i=1}^{j-1} A_i \right) B_j < x.$$

If we define $m_2 = \prod_{i=1}^{j-2} A_i + t$, then using Lemma 17 we have that $m_2 > x$, $f_\chi(t) = t$, and

$f_\chi(\prod_{i=1}^{j-2} A_i) = 0$. Using that $\phi_j = \sigma_R$ and (5) we get $\chi(0^\infty m_2 0^\infty) = 0^\infty \prod_{i=1}^{j-2} A_i \, t \, 0^\infty \in [0...x)^{\mathbb{Z}}$. Since χ is injective, $\chi \mid_{[0...x)^z}$ cannot be surjective.

The case for $f_\chi(x) = 0$ is similar. $\qquad\square$

Fig. 3: f_ϕ, where ϕ is a SIPCA and $\phi_1 = Id$.
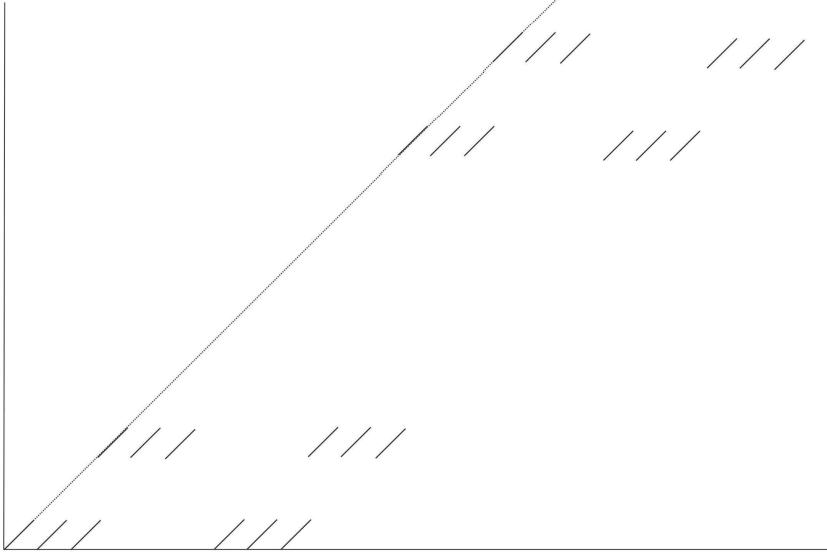
Lemma 19 *Let $\phi : [0...A)^{\mathbb{Z}} \to [0...A)^{\mathbb{Z}}$ and $\chi : [0...B)^{\mathbb{Z}} \to [0...B)^{\mathbb{Z}}$ be two 2-block surjective NCCA such that $A \leq B$ and there exists $m \in [0...A-1)$ such that $f_\phi(x) = f_\chi(x)$ for all $x \in [0...m)$. We have the following:*

·a) If $f_\chi(m) = 0$ or m, then $f_\phi(m) = 0$ or m.
·b) If $f_\chi(m) \neq 0$ or m, then $f_\phi(m) = f_\chi(m)$.

Proof: We have that $\{0^\infty de0^\infty \mid d+e = m\} \subset \{\phi(0^\infty abc0^\infty) \mid a+b+c = m \text{ and } a,b,c < m\} \cup \{\phi(0^\infty m0^\infty)\}$, and

$$\begin{aligned}
\{\chi(0^\infty m0^\infty)\} &= \{0^\infty de0^\infty \mid d+e = m\} - \{\chi(0^\infty abc0^\infty) \mid a+b+c = m \text{ and } a,b,c < m\} \\
&= \{0^\infty de0^\infty \mid d+e = m\} - \{\phi(0^\infty abc0^\infty) \mid a+b+c = m \text{ and } a,b,c < m\} \\
&= \{\phi(0^\infty m0^\infty)\}.
\end{aligned}$$

$a)$ Theorem 8 says surjective NCCA maps are bijective on the homoclinic class of 0^∞, thus

$$\{0^\infty de0^\infty \mid d+e = m\} - \{\phi(0^\infty abc0^\infty) \mid a+b+c = m \text{ and } a,b,c < m\} = \{0^\infty m0^\infty\}$$

iff $f_\phi(m) = 0$ or m.

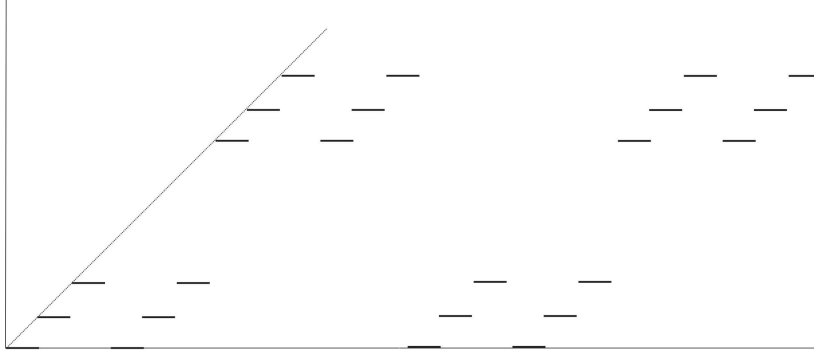$b)$ If $f_\chi(m) \neq 0$ or m, then $\{\phi(0^\infty m0^\infty)\} = \{0^\infty de0^\infty\}$, thus $f_\phi(m) = e = f_\chi(m)$.   □

To prove Theorem 13, we will need a stronger result, the inductive step shown in the following proposition.

Proposition 20 *Let* $\phi : [0...A)^{\mathbb{Z}} \to [0...A)^{\mathbb{Z}}$ *be a 2-block surjective NCCA. If there exists a SIPCA* $\chi : [0...B)^{\mathbb{Z}} \to [0...B)^{\mathbb{Z}}$ *such that* $A \leq B$ *and there exists* $m \in [0...A-1)$ *such that* $f_\phi(x) = f_\chi(x)$ *for all* $x \in [0...m)$, *then there exists a SIPCA* $\chi' : [0...C)^{\mathbb{Z}} \to [0...C)^{\mathbb{Z}}$ *such that* $A \leq C$, $f_{\chi'}(x) = f_\chi(x)$ *for* $0 \leq x < m$, *and* $f_{\chi'}(m) = f_\phi(m)$.

The proof of Proposition 20 is divided in two cases when m is a transition point and when it's not.
Case 1 (m is not a transition point)
This proof is divided into two subcases.
Case 1a) ($f_\chi(m) \neq 0$ or m)

Proof of Case 1a): It's a direct application of Lemma 19. □

Case 1b) ($f_\chi(m) = 0$ or m)

Lemma 21 *Let* $\chi = \phi_n \times \cdots \times \phi_1 : [0...A)^{\mathbb{Z}} \to [0...A)^{\mathbb{Z}}$ *with* $\phi_i : [0...A_i)^{\mathbb{Z}} \to [0...A_i)^{\mathbb{Z}}$ *be a SIPCA. If* z *is the last zero of* f_χ *(that is the largest value* z *in the domain such that* $f_\chi(z) = 0$), *then there are exactly* z *pairs* (x, y) *such that* $x < y$, $f_\chi(x) = x$, *and* $f_\chi(y) = 0$. *Analogously if* z *is the last point such that* $f_\chi(z) = z$, *then there exist exactly* z *pairs* (x, y) *such that* $x < y$, $f_\chi(x) = 0$, *and* $f_\chi(y) = y$.

Proof: Let $\chi_j = \phi_j \times \cdots \times \phi_1$, where $1 \leq j \leq n$. We denote z_j as the last zero of χ_j, and n_j as the number of pairs (x, y) such that $x < y$, $f_{\chi_j}(x) = x$, and $f_{\chi_j}(y) = 0$. It is easy to see that $z_1 = n_1$. We want to prove that $z_{j+1} - z_j = n_{j+1} - n_j$. Suppose that $\phi_1 = Id$. The number of points x such that $f_{\chi_j}(x) = 0$ is $\prod_{i \leq j \text{ odd}} A_i$ and the number of points such that $f_{\chi_j}(x) = x$ is $\prod_{i \leq j \text{ even}} A_i$.
If $\phi_{j+1} = \sigma_R$ then we won't have any new zeros of f_χ, so $n_{j+1} - n_j = 0$ (see Lemma 10).

If $\phi_{j+1} = Id$ then we have $\left(\prod_{i \leq j \text{ odd}} A_i \right) \cdot (A_{j+1} - 1)$ new zeros of f_χ. There are $\prod_{i \leq j \text{ even}} A_i$ points

where $f_\chi(x) = x$, hence we have that $n_{j+1} - n_j = \left(\prod_{i \leq j} A_i \right) \cdot (A_{j+1} - 1)$.

Now we want to calculate $z_{j+1} - z_j$. If $\phi_{j+1} = \sigma_R$ then we won't have any new zeros of f_χ so

$$z_{j+1} - z_j = 0. \text{ If } \phi_{j+1} = Id \text{ then } z_{j+1} - z_j = \left(\prod_{i \leq j} A_i \right) \cdot (A_{j+1} - 1).$$

If $\phi_1 = \sigma_R$, we simply interchange odd for even and everything works similarly.

The other part is proved analogously.   □

Lemma 22 *Let* $\chi = \phi_n \times \cdots \times \phi_1 : [0...A]^{\mathbb{Z}} \to [0...A]^{\mathbb{Z}}$ *with* $\phi_i : [0...A_i]^{\mathbb{Z}} \to [0...A_i]^{\mathbb{Z}}$ *be a SIPCA. For all non-transition points such that* $f_\chi(x) = x$*, there exist points* $p < q < r < x$ *such that* $x - r = q - p$, $f_\chi(p) = p$, $f_\chi(q) = 0$*, and* $f_\chi(r) = r$.

Similarly, for all non-transition points such that $f_\chi(x) = 0$*, there exist points* $p < q < r < x$ *such that* $x - r = q - p$, $f(p) = 0$, $f(q) = q$*, and* $f(r) = 0$.

Proof: We claim that if we have a pair (x, y) such that $x < y$, $f_\chi(x) = x$, and $f_\chi(y) = 0$, then we cannot have a different pair (x', y') such that $y - x = y' - x'$, $f_\chi(x') = x'$, and $f_\chi(y') = 0$. That is because in that case we would have $\chi(0^\infty xy'0^\infty) = 0^\infty (x + y') 0^\infty = 0^\infty (x' + y) 0^\infty = \chi(0^\infty x'y0^\infty)$, which is a contradiction since χ is injective. Thus by Lemma 21 we see that if z is a zero of f_χ then for every $w \leq z$ there is a unique pair $x, y \leq z$ with $x < y$, $y - x = w$, $f_\chi(x) = x$, and $f_\chi(y) = 0$.

Now suppose x is not a transition point and $f_\chi(x) = x$. Let $t < x$ be the previous transition point, so $t = \left(\prod_{i=1}^{j-1} A_i \right) B_j$, and let $y_1 = x - t$. By Lemma 17 $f_\chi(t) = t$, $f_\chi(\prod_{i=1}^{j-2} A_i) = 0$, and $t \leq \prod_{i=1}^{j-2} A_i$. By Lemma 21 we know we have a pair $p, q < x - t$ such that $q - p = x - t$, $f_\chi(p) = p$ and $f_\chi(q) = 0$.   □

Proof of Case 1b): Using Lemma 22 if $f_\chi(m) = m$, there exist points $p < q < r < m$ such that $m - r = q - p$, $f_\phi(p) = p$, $f_\phi(q) = 0$, and $f_\phi(r) = r$. Using Lemma 19 we know that $f_\phi(m)$ is either 0 or m. If $f_\phi(m) = 0$, then we have that $\phi(0^\infty pm0^\infty) = 0^\infty (p+m)0^\infty$ and $\phi(0^\infty rq0^\infty) = 0^\infty (r+q)0^\infty$. But since $p + m = r + q$ we have a contradiction. So, $f_\phi(m) = m = f_\chi(m)$.   □

Case 2 (m **is a transition point**).

Proof of Case 2: Since m is a transition point we have that $f_\phi(m) = 0$ or m (see Lemma 19). Let $\phi : [0...m]^{\mathbb{Z}} \to [0...m]^{\mathbb{Z}}$, $\chi_1 = \sigma_R \times \phi$, and $\chi_2 = Id \times \phi$ (for σ_R and Id on any alphabet bigger than 1). This means $\chi_1[x0] = \chi_2[x0] = \phi[x0]$ for $x < m$, but $\chi_2[m0] = 0$ and $\chi_1[m0] = m$.   □

Now we can prove Theorem 13.

Proof of Theorem 13: Let $\phi : [0...A]^{\mathbb{Z}} \to [0...A]^{\mathbb{Z}}$ be a surjective 2-block NCCA. By Proposition 20 we can use induction to see there exists a SIPCA $\chi : [0...B]^{\mathbb{Z}} \to [0...B]^{\mathbb{Z}}$ such that $A \leq B$ and $f_\phi(x) = f_\chi(x)$ for $0 \leq x < A$. If $A = B$ we are done, if $B > A$, we know that A is a transition point of χ. So by Proposition 18 we conclude that ϕ is a SIPCA.   □

Corollary 23 *If* ϕ *is a surjective 2-block NCCA on a prime alphabet then* ϕ *is a shift or the identity.*

Corollary 24 *All surjective 2-block NCCA are injective.*

5   Further questions

For bigger neighbourhoods not all surjective NCCA are SIPCA.

Example 25 *The CA ϕ on the binary alphabet defined by exchanging the blocks 10100001 and 10010001, is a well-defined bijective NCCA which is not a shift or an identity, but $\phi^2 = Id$. We can construct several similar examples where $\phi^n = \sigma_R^m$ for a certain n and m (where $\sigma_R^0 = Id$).*

We can construct several similar counter-examples where $\phi^n = \sigma_R^m$ for a certain n and m (where $\sigma_R^0 = Id$), but we can ask the following questions.

Question 26 *Are all binary surjective NCCA ψ generalized subshifts? That is, do there exist natural numbers n and m such that $\psi^m = \sigma_R^n$?*

Question 27 *If ϕ is a surjective NCCA do there exist natural numbers n and m and a SIPCA ψ such that $\phi^n = \psi^m$?*

The author is currently investigating these subjects.

NCCA are a particular class of potential conserving CA $\phi : [0...A)^{\mathbb{Z}} \to [0...A)^{\mathbb{Z}}$, when $\mu(x) = x$. The result that proves the density of periodic points for surjective NCCA (Formenti and Grange (2003)) can be easily be extended to unique ground state potentials, that is when $\mu(x) = 0$ for only one state. Theorem 8 holds also for surjective CA that conserves ground state potentials.

Question 28 *Let ϕ be a one-dimensional surjective CA that conserves the potential μ, and $\mu(x) = 0$ for only one state. Is ϕ injective?*

6   addendum

In this appendix we provide a counterexample of a surjective but non injective binary NCCA.

Example 29 *Let $A = [1000100001]$ and $B = [1001000001]$. Notice they do not overlap except at the borders, they have the same length and same weight. Now define the CA as follows.*

Everything stays where it is except where there are two of the previous blocks together, in that case the block in the right changes to A if they are the same and to B if they are different. By together we mean that they overlap in the border. We can identify strings of n blocks, and they will always remain as strings of n blocks. This map is surjective and number conserving, but it is not injective because the image of $(A'B')^\infty$ and $(B'A')^\infty$ is B'^∞. Where $A' = [000100001]$ and $B' = [001000001]$.

Hence powers of this CA are never the the identity nor the power of a shift, but on the homoclinic class of 0^∞ it is a root of the identity so it is a generalized subshift there.

Acknowledgements

References

N. Boccara and H. Fuks. Number-conserving cellular automaton rules. *Fundamenta Informaticae*, 52 (1):1 – 13, 2002. ISSN 2002-01-01. URL `http://iospress.metapress.com/content/8R32U119Q5NDJ3KB`.

E. Formenti and A. Grange. Number conserving cellular automata ii: dynamics. *Theoretical Computer Science*, 304(1-3):269 – 290, 2003. ISSN 0304-3975. doi: DOI:10.1016/S0304-3975(03)00134-8. URL `http://www.sciencedirect.com/science/article/pii/S0304397503001348`.

T. Hattori and S. Takesue. Additive conserved quantities in discrete-time lattice dynamical systems. *Physica D: Nonlinear Phenomena*, 49(3):295 – 322, 1991. URL `http://www.sciencedirect.com/science/article/pii/0167278991901508`.

G. Hedlund. Endomorphisms and automorphisms of the shift dynamical systems. *Mathematical System Theory*, 3:320 – 375, 1969.

J. Kari and S. Taati. A particle displacement representation for conservation laws in two-dimensional cellular automata. *Proceedings of JAC 2008.*, pages 65 – 73, 2008.

S. Maerivoet and B. D. Moor. Cellular automata models of road traffic. *Physics Reports*, 419(1): 1 – 64, 2005. ISSN 0370-1573. doi: 10.1016/j.physrep.2005.08.005. URL `http://www.sciencedirect.com/science/article/pii/S0370157305003315`.

E. Moore. Machine models of self-reproduction. *Proc. Symp. Appl. Math.*, 14:13 – 33, 1963.

T. Moothathu. Surjective cellular automata with zero entropy are almost one-to-one. *Chaos, Solitons & Fractals*, 44(6):415 – 417, 2011. ISSN 0960-0779. doi: 10.1016/j.chaos.2011.01.013. URL `http://www.sciencedirect.com/science/article/pii/S0960077911000440`.

A. Moreira, N. Boccara, and E. Goles. On conservative and monotone one-dimensional cellular automata and their particle representation. *Theoretical Computer Science*, 325(2):285 – 316, 2004. ISSN 0304-3975. doi: DOI:10.1016/j.tcs.2004.06.010. URL `http://www.sciencedirect.com/science/article/pii/S0304397504003950`. Theoretical Aspects of Cellular Automata.

J. Myhill. The converse of moore's garden-of-eden theorem. *Proc. Am. Math. Soc.*, 14:685 – 686, 1963.

M. Pivato. Conservation laws in cellular automata. *Nonlinearity*, 15(6), 2002. URL `http://stacks.iop.org/0951-7715/15/i=6/a=305`.

Garden-of-Eden-like theorems for amenable groups

Silvio Capobianco[1][†], Pierre Guillon[2,3][‡] and Jarkko Kari[3][§]

[1]*Institute of Cybernetics at Tallinn University of Technology, Estonia*
[2]*CNRS & IML, Marseille, France*
[3]*Mathematics Department, University of Turku, Finland*

In the light of recent results by Bartholdi, we consider several properties that, for classical cellular automata, are known to be equivalent to surjectivity. We show that the equivalence still holds for amenable groups, and give counter-examples for non-amenable ones.

Keywords: cellular automata, amenability, group theory, topological dynamics, symbolic dynamics, ergodic theory, random theory

---

1   Introduction

Retrieving global properties of cellular automata (CA) has been a main topic of research since the field was established. Indeed, the Garden-of-Eden theorem by Moore [Moo62] and its converse by Myhill [Myh62], which link surjectivity of the global map of 2D CA to *pre-injectivity* (a property that may be described as the impossibility of erasing finitely many errors in finite time) also have the distinction of being the first rigorous results of cellular automata theory. Several more properties were later proved to be equivalent to surjectivity in d-dimensional CA, such as *balancedness* of the local map [MK76] and the sending of algorithmically random configurations into algorithmically random configurations [CHJW01].

With the subsequent efforts to extend the definition of CA to more general situations than the usual Euclidean lattices, an unexpected phenomenon appeared: the Garden-of-Eden property actually depends on properties of the involved groups! In particular, counterexamples to both Moore's and Myhill's theorem are well known for CA on the free group on two generators (cf. [CSMS99]). However, from a reading of the original proofs, a key fact emerges, which is crucial for the proofs themselves: in $\mathbb{Z}^d$, the size of a hypercube is a d-th power of the side, but the number of sites on its outer surface is a polynomial of degree $d - 1$. In other words, it seems that, to get Moore's or Myhill's theorem for CA on a group G, we need that in G *the sphere grows more slowly than the ball*. What is actually sufficient is a slightly weaker property called *amenability*, which can be stated as the existence of a translation-invariant finitely additive

---

[†]Email: `silvio@cs.ioc.ee`
[‡]Email: `pierre.guillon@math.cnrs.fr`
[§]Email: `jkari@utu.fi`

probability measure on G. Bartholdi's theorem [Bar10] states then that the amenable groups are precisely those where surjective CA are pre-injective, and preserve the *product measure* on configurations.

In this paper, which illustrates work in progress, we extend the range of Bartholdi's theorem by characterizing amenable groups as those where surjective CA have additional properties. We start by considering *balancedness* [MK76], which is the combinatorial variant of measure preservation. We then include the *nonwandering* property, an important feature of dynamical systems. Finally, and for groups that have a decidable *word problem*, we prove that amenable groups are those where, in line with [CHJW01], CA preserve *descriptional complexity*.

To sum up, we get the following statement.

Theorem 1 *Let G be a finitely generated group. The following are equivalent.*

1. *G is amenable.*

2. *Every surjective CA on G is pre-injective.*

3. *Every surjective CA on G preserves the uniform product measure.*

4. *Every surjective CA on G is balanced.*

5. *Every surjective CA on G is nonwandering.*

If, in addition, G has decidable word problem, then the above are equivalent to the following:

- *Every surjective CA sends random configurations into random configurations.*

2   Preliminaries

2.1   Groups

Let G be a group. We call 1_G, or simply 1, its identity element. Given a set X, the family $\sigma = \{\sigma_g\}_{g \in G}$ of transformations of X^G, called *translations*, defined by

$$\sigma_g(c)(z) = c^g(z) = c(gz) \ \ \forall g \in G \tag{1}$$

is a *right action* of G on X^G, that is, $\sigma_{gh} = \sigma_h \circ \sigma_g$ for every $g, h \in G$. This is consistent with defining the product $\phi\psi$ of functions as the composition $\psi \circ \phi$. Other authors (cf. [CSC10]) define $\sigma_g(c)(x)$ as $c(g^{-1}x)$, so that σ becomes a *left* action. However, most of the definitions and properties we deal with do not depend on the "side" of the multiplication: we will therefore stick to (1).

A set of *generators* for G is a subset $S \subseteq G$ such that for each $g \in G$ there is a word $w = w_1 \ldots w_n$ on $S \cup S^{-1}$ such that $g = w_1 \cdots w_n$. The minimum length of such a word is called *length* of g w.r.t. S, and indicated by $\|g\|_S$, or simply $\|g\|$. G is *finitely generated* (briefly, f.g.) if S can be chosen finite. A group G is *free* on a set S if it is isomorphic to the group of reduced words on $S \cup S^{-1}$. For $r \geq 0$, $g \in G$ the *disk of radius r centered in g* is $D_r(g) = \{h \in G \mid \|g^{-1}h\| \leq r\}$. The points of $D_r(g)$ can be "reached" from the "origin" 1_G by first "walking" up to g, then making up to r steps: this is consistent with the definition of translations by (1), where to determine $c^g(z)$ we first move from 1 to g, then from g to gz. We write D_r for $D_r(1)$. We also put $U^{-r} = \{z \in G \mid D_r(z) \subseteq U\}$ and $\partial_{-r}U = U \setminus U^{-r}$. For our purposes, we will only consider f.g. groups.

A group G is *residually finite* (briefly, r.f.) if for every $g \neq 1$ there exists a homomorphism $\phi : G \to H$ such that H is finite and $\phi(g) \neq 1$. Equivalently, G is r.f. if the intersection of all its subgroups of finite index is trivial. It follows from the definitions that, if G is r.f. and $U \subseteq G$ is finite, then there exists $H \leq G$ s.t. $[G : H] \leq \infty$ and $U \cap H \subseteq \{1_G\}$.

Lemma 2 ([Fio00, Lemma 2.3.2]) *Let G be a residually finite (not necessarily f.g.) group and let F be a finite subset of G not containing 1_G. Then there exists a subgroup H_F of finite index in G, which does not intersect F, and such that the $H_F u$, $u \in F$, are pairwise disjoint.*

The *word problem* (briefly, w.p.) for a group G with a set of generators S is the set of words on $S \cup S^{-1}$ that represent the identity element of G. Although this set may depend on the choice of the presentation, its decidability does not; and although the problem is not decidable even for finitely generated groups, it is for the Euclidean groups $\mathbb{Z}^d$, the free groups, and more.

The *stabilizer* of c is the subgroup $\mathrm{st}(c) = \{g \in G \mid c^g = c\}$: be aware, that $\mathrm{st}(c)$ might not be a normal subgroup. c is *periodic* if $[G : \mathrm{st}(c)] < \infty$. If $[G : H] < \infty$ and $H \leq \mathrm{st}(c)$ we say that c is *H-periodic*. The family of periodic configurations in Q^G is indicated by $\mathrm{Per}(G,Q)$.

A group G is *amenable* if it satisfies one of the following equivalent conditions:

1. There exists a finitely additive probability measure μ on G with $\forall A \subseteq G, \forall g \in G, \mu(gA) = \mu(A)$.

2. For every finite $U \subseteq G$ and $\varepsilon > 0$ there exists a finite $K \subseteq G$ such that

$$|UK \setminus K| < \varepsilon|K| \tag{2}$$

Similar definitions want μ *right*-invariant and (2) replaced by $|KU \setminus K| < \varepsilon|K|$, or μ both left- and right-invariant and difference in (2) replaced by symmetric difference: in fact, all these definitions are equivalent. Also, if every f.g. subgroup of a given group is amenable, then the group is itself amenable.

A *bounded-propagation* $2 : 1$ *compressing map* over a group G is a map $\phi : G \to G$ such that, for some finite *propagation set* $S \subseteq G$, $\phi(g)^{-1}g \in S$ for every $g \in G$, and $|\phi^{-1}(g)| = 2$ for every $g \in G$. In particular, such a map must be surjective, and $|S| \geq 2$. By [CSC10, Theorem 4.9.2], a group has a bounded-propagation $2 : 1$ compressing map if and only if it is *not* amenable. For instance, in the case of the free group over generators a, b, one can define: $\phi(x) = y$ if $x \notin \{a^n | n \in \mathbb{N}\}$ is written in an irreducible way as yc for some $c \in \{a, b\}$; $\phi(x) = x$ otherwise. Here $S = \{1, a, b\}$ and any point y has two preimages: y and yb if y is written in an irreducible way as wa^{-1} or a^n; y and ya if y is written in an irreducible way as wb^{-1}; ya and yb otherwise.

2.2 Cellular automata

A *cellular automaton* (briefly, CA) on a group G is a triple $\mathcal{A} = \langle Q, \mathcal{N}, f \rangle$ where the *alphabet* Q is a finite set, the *neighborhood index* $\mathcal{N} \subseteq G$ is finite and nonempty, and $f : Q^{\mathcal{N}} \to Q$ is a *local function*. This, in turn, induces a *global function* on any *configuration* $c : G \to Q$, defined by

$$F_{\mathcal{A}}(c)(g) = f\left(c^g|_{\mathcal{N}}\right) = f\left(c|_{g\mathcal{N}}\right) . \tag{3}$$

Through (3) we also consider, for every finite $E \subseteq G$, a function between patterns $f : Q^{E\mathcal{N}} \to Q^E$ defined by $f(p)_j = f(p|_{j\mathcal{N}})$. *Hedlund's theorem* [CSC10, Theorem 1.8.1] states that global functions of CA are exactly those functions from Q^G to itself that commute with translations and are continuous

in the product topology. We recall that a base for this topology is given by the *cylinders* of the form $C(E, p) = \{c \in Q^G \mid c|_E = p\}$, with E a finite shape of G and $p : E \to Q$ a *pattern*: observe that, for countable groups, this base is countable. Also, the cylinders of the form $C(q, z) = \{c \mid c(z) = q\}$ form a (countable) subbase. If $p = c|_E$ we may write $C(p)$ instead of $C(c, E)$.

An *occurrence* of a pattern $p : E \to Q$ in $c \in Q^G$ is an element $g \in G$ such that $c^g|_E = p$; the pattern $p_g : gE \to Q$ defined by $p_g(gz) = p(z)$ is then a *copy* of p. For compactness reasons, a CA $\mathcal{A}$ has no *Garden-of-Eden configurations* (*i.e.*, $c \in Q^G \setminus F_{\mathcal{A}}(Q^G)$) if and only if it has no *orphan patterns*, *i.e.*, if every pattern has an occurrence in some $F_{\mathcal{A}}(c)$. Two configurations are *asymptotic* if they differ on at most finitely many points; a CA is *pre-injective* if distinct asymptotic configurations have distinct images. *Moore's Garden-of-Eden theorem* [Moo62] states that surjective CA on $\mathbb{Z}^d$ are pre-injective; *Myhill's theorem* [Myh62] states the converse implication.

A cellular automaton $\mathcal{A}$ over Q^G is *nonwandering* if for any open set $U \subset Q^G$ there exists $t \geq 1$ such that $F_{\mathcal{A}}^t(U) \cap U \neq \emptyset$; it is *transitive* if for any two open sets U and V there exists some $t \geq 1$ such that $F_{\mathcal{A}}^t(U) \cap V \neq \emptyset$. (In particular, a transitive CA is nonwandering). A state $q_0 \in Q$ is *spreading* for $\mathcal{A} = \langle Q, \mathcal{N}, f \rangle$ if for any $u \in Q^{\mathcal{N}}$ such that $u_i = q_0$ for some $i \in \mathcal{N}$ we have $f(u) = q_0$.

Remark 3 *A nonwandering non-trivial CA has no spreading state.*

By non-trivial, we mean that $|\mathcal{N}| > 1$ and $|Q| > 1$. Indeed, take a cylinder $U = C(\mathcal{N} \cup \{1_G\}, c)$ where $c_i = q_0 \neq c_{1_G}$ for some $i \in \mathcal{N} \setminus \{1_G\}$: then $F^t(U) \cap U = \emptyset$ for any $t \geq 1$.

Let $\mathcal{N} \subseteq G \leq \Gamma$ and $f : Q^{\mathcal{N}} \to Q$. The triple $\langle Q, \mathcal{N}, f \rangle$ describes both a CA $\mathcal{A}$ over G and a CA $\mathcal{A}'$ on Γ. We then say that $\mathcal{A}'$ is the CA *induced by $\mathcal{A}$ on Γ*, or that $\mathcal{A}$ is the *restriction of $\mathcal{A}'$ to G*.

2.3 Measures and randomness

Let Σ be a σ-algebra on Q^G. If $\mu : \Sigma \to [0, 1]$ is a measure on Q^G, a measurable function $F : Q^G \to Q^G$ determines a new measure $F\mu : \Sigma \to [0, 1]$ defined as $F\mu(U) = \mu(F^{-1}(U))$. We say that F *preserves* μ if $F\mu = \mu$. If Q is finite, G is countable, and Σ is the *Borel σ-algebra* generated by the open sets, by standard facts in measure theory, a measure μ is completely determined by its value on the cylinders. In particular, the measure defined by $\mu_\Pi(C(E, p)) = |Q|^{-|E|}$ is called the *uniform product measure*, because it is a product of independent uniform measures on the alphabet. *Bartholdi's theorem* [Bar10] states that the amenable groups are precisely those where surjective CA preserve μ_Π and are pre-injective.

Let μ be some probability measure over Q^G. We say that a continuous function $F : Q^G \to Q^G$ is *μ-recurrent* if for any measurable set $A \subset Q^G$ of measure $\mu(A) > 0$, there exists some time step $t \geq 1$ such that $\mu(A \cap F^t(A)) > 0$. If μ has full support, then this implies that F is nonwandering. Moreover, the *Poincaré recurrence theorem* states that any F that preserves μ is μ-recurrent.

We say that μ is *F-ergodic* (or F is *μ-ergodic*) if F preserves μ and every F-invariant set U (*i.e.*, $F^{-1}(U) = U$) has $\mu(U) \in \{0, 1\}$. In that case, the *Birkhoff ergodic theorem* gives that *μ-almost every point is μ-typical for F*, that is,

$$\mu\left(\left\{x \in Q^G \mid \forall A \in \Sigma, \lim_{n \to \infty} \frac{1}{n} \left|A \cap \{F^t(x) | 0 \leq t < n\}\right| = \mu(A)\right\}\right) = 1. \tag{4}$$

Let μ and ν be F-ergodic measures; suppose they have a typical point x in common. Then for any measurable set A, $\mu(A) = \lim_{n \to \infty} \frac{1}{n} \left|A \cap \{F^t(x) | 0 \leq t < n\}\right| = \nu(A)$: we have thus

Lemma 4 *Any two distinct F-ergodic measures have no typical point in common.*

Let $\phi : \mathbb{N} \to G$ be a total computable enumeration. It is easy to see that it induces a computable enumeration of the cylinders, which we call $B' = \{B'_i\}_{i \geq 0}$ in accordance with [CHJW01].

Given any two sequences of open sets $\mathcal{U} = \{U_i\}_{i \geq 0}$, $\mathcal{V} = \{V_j\}_{j \geq 0}$, we say that $\mathcal{U}$ is $\mathcal{V}$-*computable* if there is a *recursively enumerable* set $A \subseteq \mathbb{N}$ s.t.

$$U_i = \bigcup_{j \in \mathbb{N} : \pi(i,j) \in A} V_j \ \ \forall i \geq 0 \,, \tag{5}$$

where $\pi(i,j) = (i+j)(i+j+1)/2 + i$ is the standard primitive recursive bijection from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$. A B'-computable family $\mathcal{U}$ of open sets is a *Martin-Löf μ-test* (briefly, a M-L μ-test) if $\mu(U_n) \leq 2^n$ for every $n \geq 0$. A configuration $c \in Q^G$ *fails* a M-L μ-test $\mathcal{U}$ if $c \in \bigcap_{n \geq 0} U_n$. $c \in Q^G$ is μ-*random* (in the sense of Martin-Löf) if it does not fail any M-L μ-test. Note that, since the number of M-L μ-tests is countable, the set of μ-random configurations has full measure.

Given any pattern p, the set of configurations where p has no occurrence is an intersection of a countably infinite, computable family of cylinders U_i having equal product measure $\mu_\Pi(U_i) = m < 1$. It is then straightforward to construct a M-L μ_Π-test that every such configuration fails. If we call *rich* a configuration in which any pattern occurs (or, equivalently, whose orbit under the shift action is dense), we then have the following.

Remark 5 *Any μ_Π-random configuration is rich.*

Note that $\phi : \mathbb{N} \to G$ induces $\phi^* : Q^G \to Q^\mathbb{N}$ defined by $\phi^*(c)(n) = c(\phi(n))$. If ϕ is a computable bijection, then so is ϕ^*: in this case (cf. [GHR10, Proposition 2.5.2]) ϕ^* is continuous and preserves the product measure. In particular, c is random for the product measure on Q^G if and only if $\phi^*(c)$ is random for the product measure on $Q^\mathbb{N}$, and the set of random configurations has measure 1.

3 Results

According to Maruoka and Kimura [MK76], a d-dimensional CA with neighborhood a hypercube of radius r is n-*balanced* if each pattern on a hypercube of side n has $|Q|^{(n+2r)^d - n^d}$ pre-images. The authors then prove that a d-dimensional CA is surjective if and only if it is n-balanced for *every* n. On the other hand, the majority rule is 1-balanced but has the Garden-of-Eden pattern 01001.

The balancedness condition means that each pattern on a *given shape* has the *same number* of pre-images. (Just "patch" arbitrary shapes to "fill" a hypercube.) This works for CA over arbitrary groups.

Definition 6 *Let G be a group and let $\mathcal{A} = \langle Q, \mathcal{N}, f \rangle$ be a CA on G. $\mathcal{A}$ is* balanced *if for every finite nonempty $E \subseteq G$ and pattern $p : E \to Q$,*

$$|f^{-1}(p)| = |Q|^{|E\mathcal{N}| - |E|}. \tag{6}$$

Since the r.h.s. in (6) is always positive, no pattern is an orphan for a balanced CA. In [CSMS99], two CA on the free group on two generators are shown, one being surjective but not pre-injective, the other pre-injective but not surjective: both have an unbalanced local function. Therefore, balancedness in general groups is *strictly stronger* than surjectivity, and possibly uncorrelated with pre-injectivity.

Remark 7 *A cellular automaton is balanced if and only if it preserves the uniform product measure.*

The proof is similar to that in [CHJW01]. In fact, let $\mathcal{A} = \langle Q, \mathcal{N}, f \rangle$ and $p : E \to Q$: then $\mu_\Pi(F_\mathcal{A}^{-1}(C(E, p))) = \sum_{f(p')=p} |Q|^{-|E\mathcal{N}|}$. But balancedness means r.h.s. has $|Q|^{|E\mathcal{N}|-|E|}$ summands whatever p is, while preservation of μ_Π means l.h.s. equals $|Q|^{-|E|}$ whatever p is.

By [CSC09, Theorem 1.2] several important properties, including injectivity and surjectivity, are preserved by induction and restriction: this is also true for balancedness.

Remark 8 *Let $\mathcal{A} = \langle Q, \mathcal{N}, f \rangle$ be a CA on $G \leq \Gamma$ and $\mathcal{A}'$ the CA induced by $\mathcal{A}$ on Γ. Then $\mathcal{A}$ is balanced if and only if $\mathcal{A}'$ is balanced.*

Proof: If $\mathcal{A}'$ is balanced, then $\mathcal{A}$ clearly is. Suppose then $\mathcal{A}$ is balanced; let J be a set of representatives of the left cosets of G in Γ. Let $E \subseteq \Gamma$: put $J_E = \{j \in J \mid jG \cap E \neq \emptyset\}$. Then $E = \bigsqcup_{j \in J_E}(jG \cap E)$ and, since $\mathcal{N} \subseteq G$, $E\mathcal{N} = \bigsqcup_{j \in J_E}(jG \cap E\mathcal{N})$, with J_E finite since E is. Given $p : E \to Q$, call $p_j = p|_{jG \cap E}$ for $j \in J_E$. Then, since $\mathcal{A}'$ operates slicewise and $\mathcal{A}$ is balanced,

$$|f^{-1}(p)| = \prod_{j \in J_E} |f^{-1}(p_j)| = \prod_{j \in J_E} |Q|^{|jG \cap E\mathcal{N}| - |jG \cap E|} = |Q|^{\sum_{j \in J_E} |jG \cap E\mathcal{N}| - \sum_{j \in J_E} |jG \cap E|} ,$$

which is precisely $|Q|^{|E\mathcal{N}|-|E|}$. Since E and p are arbitrary, $\mathcal{A}'$ is balanced. $\square$

With the next statement, we strengthen [Wei00, Theorem 1.3], which states that injective CA on r.f. groups are surjective. We rely on a lemma which is immediate to prove.

Lemma 9 *If $F : Q^G \to Q^G$ commutes with translations, then $\mathrm{st}(c) \subseteq \mathrm{st}(F(c))$ for every $c \in Q^G$. In particular, if F is bijective then $\mathrm{st}(c) = \mathrm{st}(F(c))$.*

Theorem 10 *Let G be a residually finite group and $\mathcal{A} = \langle Q, \mathcal{N}, f \rangle$ an injective CA over G. Then $\mathcal{A}$ is balanced.*

Proof: Let E be a finite subset of G: it is not restrictive to suppose $1 \in E \cap \mathcal{N}$, so that $E, \mathcal{N} \subseteq E\mathcal{N}$. Suppose, for the sake of contradiction, that $p : E \to Q$ satisfies $|F_\mathcal{A}^{-1}(p)| = M > |Q|^{|E\mathcal{N}|-|E|}$. Since G is residually finite, by Lemma 2 there exists a subgroup $H \leq G$ of finite index such that $H \cap E\mathcal{N} = H \cap \mathcal{N} = \{1\}$: if J is a set of representatives of the right cosets of H such that $E\mathcal{N} \subseteq J$, then

$$|\{\pi : J \to Q \mid F_\mathcal{A}(\pi)|_E = p\}| = M \cdot |Q|^{[G:H]-|E\mathcal{N}|} > |Q|^{[G:H]-|E|} . \tag{7}$$

The r.h.s. in (7) is the number of H-periodic configurations that coincide with p on E. Since $\mathcal{A}$ is injective and G is r.f., by [Wei00, Theorem 1.3] $\mathcal{A}$ is reversible, and by Lemma 9, $F_\mathcal{A}$ sends H-periodic configurations into H-periodic configurations. But because of (7) and the pigeonhole principle, there must exist two H-periodic configurations with the same image according to $F_\mathcal{A}$: which contradicts injectivity of $\mathcal{A}$. $\square$

The proof of Moore's and Myhill's theorems for CA on amenable groups given in [CSMS99] is based on the following lemma.

Lemma 11 ([CSMS99, Step 1 in proof of Theorem 3]) *Let G be an amenable group, $q \geq 2$, and $n > r > 0$. For $L = D_n$ there exist $m > 0$ and $B \subseteq G$ such that B contains m disjoint copies of L and*

$$(q^{|L|} - 1)^m \cdot q^{|B|-m|L|} < q^{|B^{-r}|} . \tag{8}$$

We use Lemma 11 to get a combinatorial proof of the equivalence between surjectivity and balancedness, that was already essentially stated in [Bar10].

Theorem 12 *Let G be an amenable group and let $\mathcal{A}$ a CA on G. If $\mathcal{A}$ is surjective then $\mathcal{A}$ is balanced.*

Proof: Put $L = D_n$, $L' = D_{n-r}$, $q = |Q|$. Suppose, for the sake of contradiction, that $\mathcal{A}$ is not balanced. Then, for suitable n, there is a pattern $p : L' \to Q$ that has at most $q^{|L|-|L'|} - 1$ pre-images. Let m and B be as by Lemma 11. Consider the patterns on B whose image under the global rule of $\mathcal{A}$ coincides with p on each of the m copies of L' contained in those of L: their number t is at most

$$\left(q^{|L|-|L'|} - 1 \right)^m q^{|B|-m|L|} .$$

However, $\left(q^{|L|-|L'|} - 1 \right) \le q^{-|L'|} \left(q^{|L|} - 1 \right)$, so that, by Lemma 11,

$$t \le q^{-m|L'|} \left(q^{|L|} - 1 \right)^m q^{|B|-m|L|} < q^{|B^{-r}|-m|L'|} .$$

But the last term is precisely the number of patterns on B^{-r} that coincide with p on each of the given m copies of L'. There are more of these than available pre-images, so one of them must be an orphan. □

Thanks again to Lemma 11, [CHJW01, Point 1 of Theorem 4.4] generalizes to amenable groups.

Proposition 13 *Let G be an amenable group and let $\mathcal{A} = \langle Q, D_r, f \rangle$, $r > 0$, be a CA on G. If c is not rich then $F_{\mathcal{A}}(c)$ is not rich.*

Proof: Suppose there is a pattern with support $L = D_n$, $n > r$, that does not occur in c. Choose m and B according to Lemma 11. By hypothesis, the number of patterns with support B that occur in c is at most $(q^{|L|} - 1)^m q^{|B|-m|L|}$, with $q = |Q|$; therefore, the number of patterns with support $B \setminus \partial_r B$ which occur in $F_{\mathcal{A}}(c)$ cannot exceed this number too. By Lemma 11, this is strictly less than $q^{|B|-|\partial_r B|}$, which is the total number of patterns with support $B \setminus \partial_r B$: hence, some of those patterns do not occur in $F_{\mathcal{A}}(c)$. □

We now consider another property that, for CA on $\mathbb{Z}^d$, is equivalent to surjectivity: sending μ_Π-random configurations into μ_Π-random configurations. Before going ahead, we must remember that, according to [CHJW01], the definition of a random configuration on $\mathbb{Z}^d$ depends on the existence (and choice!) of a total computable bijection from $\mathbb{N}$ to $\mathbb{Z}^d$. This is still ensured for a general group G when it has a decidable word problem: we thus can first enumerate $D_0 = \{1_G\}$, then $D_1 \setminus D_0$, then $D_2 \setminus D_1$, and so on.

The proofs of the following two statements are then similar to the original ones in [CHJW01]

Lemma 14 *Let G be a group with decidable word problem, $\mathcal{U}$ a B'-computable sequence, and $\mathcal{A}$ a CA on G. Then $F_{\mathcal{A}}^{-1}(\mathcal{U})$ is a B'-computable sequence.*

Proof: Let A be a r.e. set such that $U_i = \bigcup_{\pi(i,j) \in A} B'_j$ for every $i \ge 0$, where the B'_j are cylinders. Since $\mathcal{A}$ is a CA, $F_{\mathcal{A}}^{-1}(U_i)$ is itself a union of cylinders: such union is computable because G has decidable word problem. By exploiting these facts and the primitive recursive functions $L, K : \mathbb{N} \to \mathbb{N}$ such that $\pi(L(n), K(n)) = n$ for every $n \ge 0$, we can construct a r.e. set Z such that $F_{\mathcal{A}}^{-1}(U_i) = \bigcup_{\pi(i,j) \in Z} B'_j$ for every $i \ge 0$. □

Proposition 15 *Let G be a group with decidable word problem and $\mathcal{A}$ a CA over G. If $F_{\mathcal{A}}(c)$ is μ_{Π}-random whenever c is, then $\mathcal{A}$ is surjective. If $\mathcal{A}$ preserves μ_{Π}, then $F_{\mathcal{A}}(c)$ is μ_{Π}-random when c is.*

Proof: Since μ_{Π}-random configurations form a set of measure 1 and contain occurrences of any pattern, the first part is immediate. For the second part, if $F_{\mathcal{A}}\mu_{\Pi} = \mu_{\Pi}$, then by Lemma 14 the preimage of a M-L μ_{Π}-test is still a M-L μ_{Π}-test: but if $F_{\mathcal{A}}(c)$ fails $\mathcal{U}$, then c fails $F_{\mathcal{A}}^{-1}(\mathcal{U})$. $\square$

From Proposition 15 combined with Theorem 12 follows

Corollary 16 *Let G be an amenable group with decidable word problem and $\mathcal{A}$ be a surjective CA on G. If c is μ_{Π}-random then $F_{\mathcal{A}}(c)$ is μ_{Π}-random.*

What is the role of amenability in all this? Could this happen on non-amenable groups as well? The following counterexample shows that this is not the case.

Example 17 (Surjective CA with a spreading state) *Let G be a non-amenable group; let ϕ be a bounded-propagation $2:1$ compressing map with propagation set S. Let $\preceq$ be a total ordering of S and let $Q = S \times \{0,1\} \times S \sqcup \{q_0\}$, where $q_0 \notin S \times \{0,1\} \times S$. Let $\mathcal{A} = \langle Q, S, f \rangle$ with:*

$$
f: \quad Q^S \quad \to \quad Q
$$
$$
u \quad \mapsto \quad \left|
\begin{array}{ll}
q_0 & \textit{if } \exists s \in S, u_s = q_0, \\
(p, \alpha, q) & \textit{if } \exists!(s,t) \in S \times S, s \prec t, u_s = (s, \alpha, p), u_t = (t, 1, q), \\
q_0 & \textit{otherwise.}
\end{array}
\right.
$$

Then $\mathcal{A}$ admits the spreading state q_0, and at least one other state, hence it is not nonwandering. Nevertheless, it is surjective.

Proof: Let $x \in Q^G$, $i \in G$, $j = \phi(i)$: then $i = js$ for some $s \in S$, and there exists a unique $t \in S \setminus \{s\}$ such that $\phi(jt) = j$. If $x_j = q_0$, then set $y_i = (s, 0, s)$: otherwise, we can write $x_j = (p, \alpha, q)$. If $s \prec t$, then set $y_i = (s, \alpha, p)$; otherwise set $y_i = (s, 1, q)$. This definition has the property that for any $i \in G$, $y_i \in \{\phi(i)^{-1}i\} \times \{0,1\} \times S$. Let us prove that the configuration y is a preimage of x by the global map of the CA. Let $j \in G$ and $s, t \in S$ such that $s \prec t$, $y_{js} \in \{s\} \times \{0,1\} \times S$, and $y_{jt} \in \{t\} \times \{0,1\} \times S$. Then $s = \phi(js)^{-1}js$ and $t = \phi(jt)^{-1}jt$, and $\phi(js) = \phi(jt) = j$: hence, there exists *exactly one* such pair (s,t). If $x_j = q_0$, then the definition of y gives $y_{jt} = (t, 0, t)$, and f will apply its third subrule. If x_j is written (p, α, q), then $y_{js} = (s, \alpha, p)$ and $y_{jt} = (t, 1, q)$, and f will apply its second subrule. $\square$

Now, let G be a non-amenable group with decidable w.p., $\mathcal{A}$ the CA from Example 17, and c a μ_{Π}-random configuration. By Remark 5, there are some points $g \in G$ where $c(g) = q_0$: since $|S| \geq 2$, $F_{\mathcal{A}}(c)$ cannot have isolated q_0's, and by the same Remark 5, it cannot be μ_{Π}-random. On the other hand, as a consequence of the Poincaré recurrence theorem, a CA that preserves μ_{Π} is nonwandering: we have thus yet another characterization of amenable groups as those where surjective CA are nonwandering.

A general scheme of the implications is provided by Figure 1. By joining Bartholdi's theorem, Remark 7, Corollary 16, Example 17, and the observations above we get Theorem 1.

We conclude this section with some results involving general measures for the configuration space.

Proposition 18 *Let $\mathcal{A} = \langle Q, \mathcal{N}, f \rangle$ be a CA over group G, and μ a σ_k-ergodic Borel probability measure on Q^G for some $k \in G$. Then for $t \geq 1$, $F_{\mathcal{A}}^t \mu$ is also σ_k-ergodic. Moreover, $F_{\mathcal{A}}$ is μ-recurrent if and only if $F_{\mathcal{A}}^t$ preserves μ for some $t \geq 1$.*
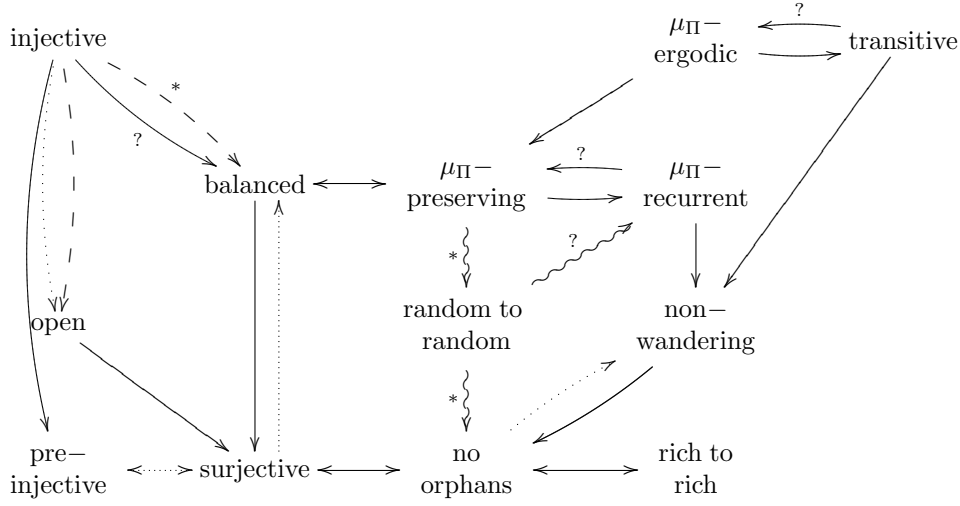
Figure 1: A diagram of implications between cellular automata properties. Full lines hold for every group; dotted lines hold for amenable groups; dashed lines hold for residually finite groups; wavy lines hold for countable groups with decidable word problem. Starred implications are proved in the present paper. Implications with a question mark are conjectured.

Proof: Since σ_k and $F_\mathcal{A}$ commute, if $\sigma_k^{-1}(U) = U$ then $\sigma_k^{-1}(F_\mathcal{A}^{-t}(U)) = F_\mathcal{A}^{-t}(U)$ as well, hence $F_\mathcal{A}^t \mu(U) \in \{0,1\}$; also, for any Borel set U, $F_\mathcal{A}^t \mu(\sigma_k^{-1}(U)) = \mu(\sigma_k^{-1}(F^{-t}(U))) = F_\mathcal{A}^t \mu(U)$.

By the Poincaré recurrence theorem, if $F_\mathcal{A}^t$ preserves μ then it is μ-recurrent, and this trivially implies that F also is. For the converse implication, let U be the set of μ-typical configurations for σ_k: then $\mu(U) = 1$, so $t \geq 1$ exists such that $\mu(U \cap F_\mathcal{A}^t(U)) > 0$. But since σ_k and $F_\mathcal{A}$ commute, if x is μ-typical for σ_k, then $F_\mathcal{A}^t(x)$ is $(F_\mathcal{A}^t \mu)$-typical for σ_k: thus, μ and $F_\mathcal{A}^t \mu$ are two σ_k-ergodic measures having a common typical point for σ_k, so they are equal by Lemma 4. □

If F is a μ-recurrent system where μ is σ_k-ergodic for some $k \in G$, then for suitable $t \geq 1$ the *mean* of $F^i \mu$ for $0 \leq i < t$ is F-invariant. Note that this does *not* imply that F is μ-invariant: a simple counter-example is a CA performing a simple state permutation, over a non-uniform Bernoulli measure.

Example 19 *Let $Q = \{0,1\}$ and let μ be a product of independent identical measures $\mu(0) = 1/3$, $\mu(1) = 2/3$; let $\mathcal{A} = \langle Q, \{1_G\}, f \rangle$ with $f(z) = 1 - z$. Then $F_\mathcal{A}^2 \mu = \mu$ but $F_\mathcal{A} \mu \neq \mu$. However, if $\bar{\mu}_2 = (\mu + F_\mathcal{A} \mu)/2$, then $F_\mathcal{A} \bar{\mu}_2 = \bar{\mu}_2$.*

4 Conclusions

We have shown that several characterizations of surjective CA which are known to hold on Euclidean groups also hold in the more general case of amenable groups.

This is a work in progress, and many more questions arise. Among those:

1. Does Myhill's theorem only hold on amenable groups?

2. What is the actual role of the word problem in Lemma 14 and Proposition 15? Can we find some amenable groups with undecidable word problem but where surjective CA still send μ_Π-random to μ_Π-random?

3. For the uniform product measure, is every recurrent CA invariant?

Acknowledgements

References

[Bar10] L. Bartholdi. Gardens of eden and amenability on cellular automata. *J. Eur. Math. Soc.*, 12:141–148, 2010.

[BGH+11] L. Bienvenu, P. Gács, M. Hoyrup, C. Rojas, and A. Shen. Algorithmic tests of randomness with respect to a class of measures. arXiv:1103.1529v2[math.LO], 2011.

[CHJW01] C. Calude, P. Hertling, H. Jürgensen, and K. Weihrauch. Randomness on full shift spaces. *Chaos, Solitons & Fractals*, 12:491–503, 2001.

[CSC09] T. Ceccherini-Silberstein and M. Coornaert. Induction and restriction of cellular automata. *Ergod. Th. & Dynam. Sys.*, 29:371–380, 2009.

[CSC10] T. Ceccherini-Silberstein and M. Coornaert. *Cellular Automata and Groups*. Springer Verlag, 2010.

[CSMS99] T. Ceccherini-Silberstein, A. Machì, and F. Scarabotti. Amenable groups and cellular automata. *Annales de l'Institut Fourier*, 49:673–685, 1999.

[Fio00] F. Fiorenzi. *Cellular automata and finitely generated groups*. PhD thesis, Sapienza Università di Roma, 2000.

[GHR10] S. Galatolo, M. Hoyrup, and C. Rojas. Effective symbolic dynamics, random points, statistical behavior, complexity and entropy. *Inform. & Comput.*, 208:23–41, 2010.

[MK76] A. Maruoka and M. Kimura. Condition for injectivity of global maps for tessellation automata. *Inform. Control*, 32:158–162, 1976.

[Moo62] E.F. Moore. Machine models of self-reproduction. In *Proc. Symp. Appl. Math.*, volume 14, pages 17–33, 1962.

[Myh62] J. Myhill. The converse of moore's garden-of-eden theorem. In *Proc. Amer. Mat. Soc.*, volume 14, pages 685–686, 1962.

[Pet83] K. Petersen. *Ergodic theory*. Cambridge studies in advanced mathematics 2. Cambridge University Press, 1983.

[Wei00] B. Weiss. Sofic groups and dynamical systems. *Sankhyā: Indian J. Stat.*, 62:350–359, 2000.

CA-based Diffusion Layer for an SPN-type Block Cipher

Jaydeb Bhaumik[1][†] and Dipanwita Roy Chowdhury [2][‡]

[1] *Dept.of ECE, HIT Haldia, India*
[2] *Dept.of CSE, IIT Kharagpur, India*

In this paper, a new method to design a diffusion layer based on a maximum distance separable code for a Substitution Permutation Networks(SPN)-type block cipher is proposed. The proposed diffusion layer has been designed employing Cellular Automata(CA). It is first time, the maximum-length group CA has been used to design diffusion layer of lengths 16-bit, 32-bit, 64-bit and 128-bit for an SPN-type block cipher. As a case study, a 128-bit diffusion layer is discussed in detailed. The purpose of using a single 128-bit diffusion layer for a block cipher of block length 128-bit is to provide complete diffusion in a single round. Also, superiority of the proposed diffusion over AES-like diffusion has been discussed here.

Keywords: Diffusion, Block cipher, Cellular Automata

---

1   Introduction

Diffusion is an important cryptographic properties for the design of a secure block cipher. Each round function of an SPN-type block cipher consists of three layers : substitution layer, permutation layer and round key mixing layer. The permutation layer dissipates the statistics of the plaintext in the statistics of the ciphertext, it is often referred to as the diffusion layer.Only a substitution layer which is strong against differential cryptanalysis (DC) and linear cryptanalysis (LC) does not guarantee a secure SPN structure against DC and LC if a diffusion layer does not provide an avalanche effect. Hence the role of the diffusion layer is very important for the design of a secure block cipher.

In Advanced Encryption Standard (AES)[DR02], diffusion is accomplished by combination of Mix-Columns and ShiftRows. MixColumn operates on a 32-bit data at a time and it is based on a Maximum Distance Separable (MDS) code. The distance between any two distinct codewords (called branch number [Dae95]) is five in case of AES. So all plaintext bits diffuse completely after two rounds. Therefore, diffusion in AES is relatively slow. Junod and Vaudenay have presented perfect diffusion primitives for block ciphers by considering software implementations on various platforms [JV04]. Authors in [JV04] have constructed efficient (4×4) and (8×8) matrices over $GF(2^8)$ for block cipher. Hence, for a 128-bit

---

block cipher, multiple parallel modules are required and complete diffusion is not possible in a single round. Recently, a new (16×16) involutory MDS matrix for AES is proposed in [NJA09]. In scheme [NJA09], complete diffusion is possible after a single round but the drawback of the proposed construction is the performance penalty. SHARK [RDP$^+$96] is a 64-bit block cipher which uses a Reed-Solomon (RS) code to construct its diffusion layer. It has branch number 9. Two other block ciphers Khazad [BRb] and Anubis [BRa] have been designed by Barreto and Rijmen. Khazad is a 64-bit, 8-round block cipher and it employs an MDS diffusion layer which has branch number 9. It provides complete diffusion after one round. Anubis is a 128-bit, $12 - 18$ rounds block cipher but it has a slower, Rijndael-like 32-bit diffusion layer [Bir03]. A diffusion layer with large branch number increases the security of cipher. A common feature exploited by several existing attacks on reduced-round AES is the slow diffusion via the combination of ShiftRows and MixColumns [NJA09].

In this paper, first time a maximum length group CA [CRCNC97] has been employed to design diffusion layer for an SPN-type block cipher. The proposed diffusion layer is based on MDS codes. Diffusion layers of lengths 16-bit, 32-bit, 64-bit and 128-bit are designed using CA for an SPN-type block cipher. As a case study, design of an 128-bit diffusion layer and its superiority have been discussed in detailed.

The rest of this paper is organized as follows. For the sake of completeness, a brief description of CA is given in next section. The proposed diffusion layer is described in section 3. In section 4, implementation of diffusion layer employing CA and combinational circuits are discussed. The superiority of the proposed diffusion over AES-like diffusion is explained in section 5 and finally the paper is concluded in section 6.

2   Cellular Automata

It consist of a number of cells arranged in a regular manner, where the state transitions of each cell depends on the state of its neighbors. Each cell consists of a D flip-flop and a combinational logic implementing the next-state function. An r-cell CA can be characterized by an $(r \times r)$ binary characteristic matrix T. The state S_{t+1} can be computed by multiplying S_t with T, where S_t and S_{t+1} represents the states of the CA at t-th and $(t + 1)$-th instant respectively. If the next-state function of a cell is expressed in the form of a truth table, the decimal equivalent of the output is conventionally called the rule number[CRCNC97] for the CA. If the state transition graph of an r-cell CA consists of a single cycle containing all $L = 2^r - 1$ non zero states, then the CA is called as maximum length CA. For example, a four-cell hybrid one dimensional $(1D)$ maximum length CA having rule vector $(90, 150, 90, 150)$ and the corresponding characteristic matrix is as follows

$$T_4 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

The characteristic polynomial associated with T_4 is $m(x) = x^4 + x + 1$, which is a primitive polynomial in $GF(2^4)$. A 4-cell maximum length CA is presented in Fig. 1. During hardware implementation, the i-th row of the matrix T describes the neighborhood relation of the i-th cell. If an element T_{ij} (at row i and column j of matrix T) is 1, then the ith cell in the array has neighborhood dependence on the jth cell. As for example, second row of T_4 is 1110. Therefore, the second cell (from left) in Fig. 1 is connected with right and left neighbors as well as its own output.
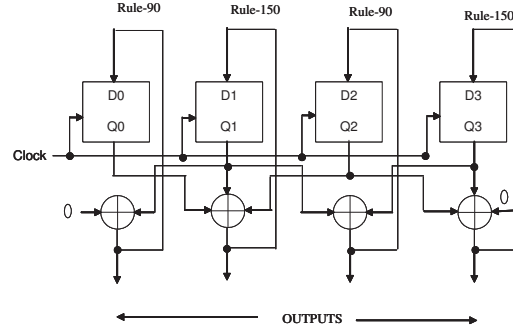
Fig. 1: A four cell null boundary hybrid linear one dimensional CA.

3 Diffusion Layer Using CA-based MDS code

A diffusion layer plays an important role in the design of a secure block cipher. It does not allow to preserve some characteristics that result from a substitution layer. Several SPN-type block ciphers use MDS code for the construction of diffusion layer. The main aim in the design of MDS code based diffusion layer is to reduce the computational cost by selecting an appropriate MDS matrix. Here, one such diffusion layer based on Cellular Automata (CA) is introduced. One rule vector for an 8-cell maximum length CA is $< 150\ 150\ 90\ 150\ 90\ 150\ 90\ 150 >$. The corresponding characteristic matrix (T) of an 8-cell maximum length CA is as follows.

$$T = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

The characteristic polynomial is defined as determinant of $(T + x[I])$. The polynomial associated with T is $p(x) = x^8 + x^7 + x^6 + x^5 + x^4 + x^2 + 1$, which is one of the primitive polynomials of $GF(2^8)$. In the rest of this paper, T will indicate characteristic matrix of the 8-cell maximum length CA, which is mentioned above. Rule-90 and Rule- 150 are considered here because they are well suited for VLSI implementation.

3.1 CA-based MDS code:

A $[n, m, d]$ code that meets the Singleton bound, namely $d = n - m + 1$, is called an MDS code, where m is the number of data symbols, n is the number of symbols in a codeword and d is the minimum distance of separation between two distinct codewords. For an MDS code, the minimum number of non-zero symbols in any codeword is d. The generator matrix $G = [I|M]$ of a $[n, m, d]$ MDS code over $GF(2^8)$ is a $(m \times n)$ matrix, where elements of G are in $GF(2^8)$, I is a $(m \times m)$ identity matrix and M is a $(m \times n - m)$ matrix. Sometimes, the matrix M is designed using Vandermonde's construction.

In this case, each element of M is power of a primitive element of $GF(2^8)$. In case of maximum length CA, a characteristic matrix T is equivalent to a primitive element α. Therefore, the matrix $M_{16 \times 16}$ of a $[32, 16, 17]$ code can be constructed from characteristic matrix T, where each element of M is a power of T and it is as follows.

$$M_{16 \times 16} = \begin{bmatrix} T & T^2 & T^3 & . & . & . & T^{15} & T^{16} \\ T^2 & T^4 & T^6 & . & . & . & T^{30} & T^{32} \\ T^3 & T^6 & T^9 & . & . & . & T^{45} & T^{48} \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ T^{15} & T^{30} & T^{45} & . & . & . & T^{225} & T^{240} \\ T^{16} & T^{32} & T^{48} & . & . & . & T^{240} & T \end{bmatrix}$$

The linear code generated by the generator matrix $G = [I|M]$ is an MDS code, where $I_{16 \times 16}$ is an identity matrix and each element of I is an (8×8) binary matrix. The linear code has dimension 16, length 32 and the minimum distance of separation between two distinct codewords is 17. The matrix M is sometimes called MDS matrix.

Similarly, the MDS matrix for a $[16, 8, 9]$ code is as follows.

$$M_{8 \times 8} = \begin{bmatrix} T & T^2 & T^3 & . & . & . & T^7 & T^8 \\ T^2 & T^4 & T^6 & . & . & . & T^{14} & T^{16} \\ T^3 & T^6 & T^9 & . & . & . & T^{21} & T^{24} \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ T^7 & T^{14} & T^{21} & . & . & . & T^{49} & T^{56} \\ T^8 & T^{16} & T^{24} & . & . & . & T^{56} & T^{64} \end{bmatrix}$$

In case of a $[8, 4, 5]$ code, the MDS matrix $M_{4 \times 4}$ is as follows.

$$M_{4 \times 4} = \begin{bmatrix} T & T^2 & T^3 & T^4 \\ T^2 & T^4 & T^6 & T^8 \\ T^3 & T^6 & T^9 & T^{12} \\ T^4 & T^8 & T^{12} & T^{16} \end{bmatrix}$$

Similarly, a $[4, 2, 3]$ code can be designed using the generator matrix $G_{4 \times 4} = [I|M_{2 \times 2}]$, where

$$M_{2 \times 2} = \begin{bmatrix} T & T^2 \\ T^2 & T^4 \end{bmatrix}$$

The code generated by $G_{4 \times 4}$ has dimension 2, length 4 and the minimum distance between two distinct codewords is 3.

3.2 Design of diffusion layer

MDS code thus generated can be employed to design a 128-bit, 64-bit, 32-bit and a 16-bit diffusion layers. For a 128-bit diffusion layer, the output $Y = [y_1 \ y_2 \ y_3 \ ... \ y_{16}]$ is computed from the input $X = [x_{16} \ ... \ x_3 \ x_2 \ x_1]$ of the diffusion layer by using the relationship $[Y] = [X][M_{16 \times 16}]$, where each x_i and y_i is an 8-bit vector. In case of a 64-bit diffusion layer, the relationship is $[Y] = [X][M_{8 \times 8}]$, where X and Y are input and output of size 8 bytes each. The output Y of a 32-bit diffusion layer is obtained by multiplying $[X]$ with $M_{4 \times 4}$, where X and Y are input and output of size 4 bytes each. Similarly, for a 16-bit diffusion layer, the output $Y = [y_1 \ y_2]$ is computed from the input $X = [x_2 \ x_1]$ using the relationship $[Y] = [X][M_{2 \times 2}]$.

For a 128-bit block cipher, a 128-bit diffusion layer can be designed either employing a single 128-bit diffusion layer or 2/4/8 parallel 64-bit/32-bit/16-bit diffusion layers respectively. For a 128-bit block cipher, a single 128-bit diffusion layer can be used in all rounds and it is advantageous for a single round iterative architecture. So, it is good for folded implementation. Complete diffusion is achieved after single round. There are minimum 34 active S-boxes in a 128-bit four rounds cipher. But in case of 4-round AES, minimum number of active S-boxes is 25. In the rest of this paper, a single 128-bit diffusion layer is chosen for a 128-bit block cipher.

4 Implementation of CA-based diffusion layer

In this section, two implementation techniques of proposed diffusion layer are explained. In one implementation 8-bit maximum length group CAs have been employed and in other implementation output bits are expressed interms modulo 2 addition of input bits.

4.1 CA-based implementation
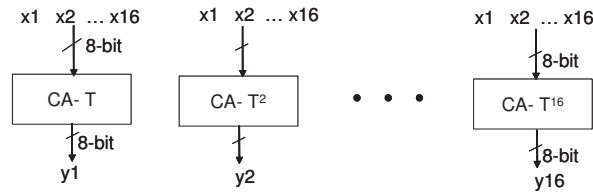
Fig. 2: Block Diagram of 128-bit diffusion Layer

Figure 2 shows a 128-bit diffusion layer using an 8-bit maximum length cellular automata. In Fig. 2, sixteen output bytes $y_1, y_2, ...y_{16}$ are computed by running CA-T, CA-T^2, ... CA-T^{16} respectively for 16 times, while sequentially feeding 16 input bytes (starting from x_1 up to x_{16}). Following algorithm explains method for computing output byte y_i

Comp-out-byte: Output byte y_i computation algorithm

s denotes the state of the 8 bits CA

begin

$s := 0$;   **for** $k = 1 \ to \ 16$ **do**

    begin

$s := s \oplus x_k;$
     Run the CA for one cycle; (CA with characteristic matrix T^i)
   end;
 $y_i := s;$
end;
 Figure 3 shows the internal architecture of CA-T, which represents an 8-bit CA having characteristic
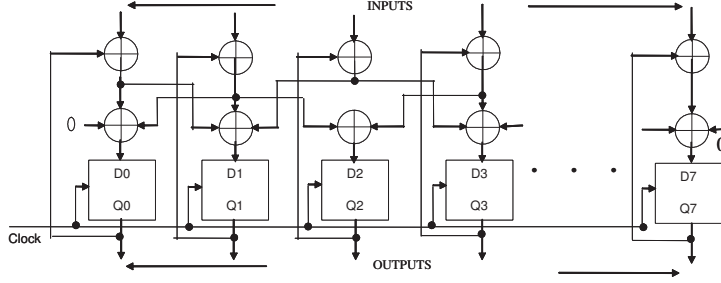
Fig. 3: Internal Architecture of CA-T

matrix T.

4.2   *Combinational logic implementation*

The MDS matrix M has dimension (16×16), where each element is an (8×8) binary matrix. A (128×128) binary matrix is realized by substituting the values of all elements of M which are power of T. The value of T is given in Section 2 and the other powers of T are obtained by matrix operation in $GF(2)$. As a result, each output bit of the diffusion layer can be expressed as bitwise XOR of input bits.

5   Superiority of the proposed diffusion layer

In this section, a single 128-bit diffusion layer is used to construct an SPN-type block cipher. The specification of the cipher is given first. Then the superiority of proposed diffusion layer over AES-like diffusion against linear and differential cryptanalysis are analyzed. It has been shown that minimum number of active S-boxes for a 4-round cipher attains the optimum value which enhances the cipher security.The block diagram of the cipher is shown in Fig.4. It is a 128-bit SPN type block cipher and the number of rounds is eight. Each round consists of three layers: linear (XOR) round key mixing layer, substitution layer having 16 AES S-boxes and diffusion layer based on MDS code. In Fig. 4, a single 128-bit diffusion layer is used in all rounds.

5.1   *Differential probability value for characteristic:*

Differential cryptanalysis seeks to exploit a scenario where a particular output difference ΔY occurs given a particular input difference ΔX with a very high probability. The probability of the differential
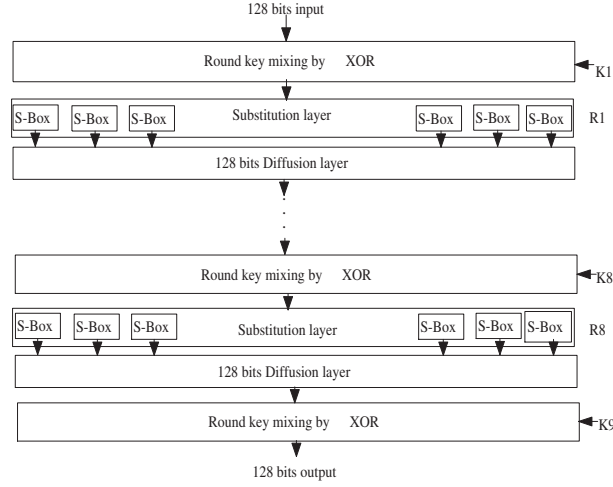
Fig. 4: Block Diagram of an SPN-type Block Cipher

is a more accurate measure for the success rate of a differential attack. But in general, the probability of differential over multiple rounds of an SPN-type block cipher is difficult to compute. Therefore, in this paper the upper bound of expected differential probability (EDP) for characteristic is computed. The differential probability $DP_f(a, b)$ of a differential (a, b) with respect to $f(x)$ is defined in [DLP$^+$09] and the expression is as follows.

$$DP_f(a, b) = 2^{-n} \#\{x \in F_2^n | f(x + a) = f(x) + b\} \tag{1}$$

If f is a function of parameter k, then expected differential probability (EDP) of a differential (a, b) is defined as the mean value of parameterized differential probability $DP[k](a, b)$ and expressed as

$$EDP(a, b) = 2^{-|\kappa|} \Sigma_{k \in \kappa} DP[k](a, b) \tag{2}$$

here k is assumed to be a uniformly distributed random variable taking values in κ, set of all keys of size $|\kappa|$ bits. AES S-boxes have been used in the proposed construction of cipher. For AES S-box, maximum differential probability [DLP$^+$09] $max_y DP(x, y) = 2^{-6}$. The 128-bit diffusion layer has branch number 17. There are at least 34 active S-boxes in the 4-rounds cipher. It assumed that the round keys which are XORed with the input data at each round are independent and random. Therefore, the best EDP value for characteristics of the 128-bit 2-round cipher is bounded by $(2^{-6})^{17} = 2^{-102}$. For a 4-round cipher the value is $(2^{-102})^2 = 2^{-204}$. Therefore, classical differential attack is not possible after four rounds. A comparison of the EDP value for the characteristic of four rounds of existing related block ciphers are given in Table 1. It is observed that EDP value for characteristic of the proposed 4-round cipher is less compared to same length (128-bit) block cipher AES and Anubis.

5.2 *Maximum probability for linear characteristic:*

The basic idea in linear cryptanalysis is to approximate the operation of a portion of the cipher with an expression that is linear (XOR) and has a suitably large enough linear probability bias. According to Hong

Tab. 1: Comparison of probability of differential for characteristic

Name of the Cipher	Block length	Branch no.	Prob. of diff. Characteristic
Proposed one	128	17	2^{-204}
SHARK [RDP$^+$96]	64	9	2^{-108}
AES [DR02]	128	5	2^{-150}
KHAZAD [BRb]	64	9	2^{-90}
ANUBIS [BRa]	128	5	2^{-125}

et al., the linear probability [HLL$^+$01] of an S-box S_i is defined as follows.

$$LP^{S_i}(\Gamma x \to \Gamma y) = (\frac{\#\{x \in Z_2^m | \Gamma x.x = \Gamma y.S_i(x)\}}{2^{m-1}} - 1)^2$$
$$LP_{max}^{S_i} = max_{\Gamma x, \Gamma y \neq 0} LP^{S_i}(\Gamma x \to \Gamma y) \tag{3}$$

where Γx and Γy are input and output mask respectively and $1 \leq i \leq n$. It has been shown in [HLL$^+$01] that the probability for each linear characteristic of Substitution, Diffusion and Substitution (SDS) function is bounded by q^n, where $q = LP_{max}^{S_i}$ is the maximum linear probability of S-boxes in the substitution layer and $n+1$ is a lower bound for the number of active S-boxes in two consecutive rounds of a linear approximation. In the proposed cipher, all sixteen S-boxes in the substitution layer are same and the AES S-box is used. For AES S-box, the value of $LP_{max}^{S_i} = LP^S = (\frac{144}{128} - 1)^2 = 2^{-6}$. Therefore, the probability for linear characteristic of SDS function is bounded by $(2^{-6})^{16} = 2^{-96}$. So the maximum probability for linear characteristic of the four rounds cipher is $(2^{-96})^2 = 2^{-192}$. Hence four rounds of proposed construction is sufficient to resist classical linear attack. A comparison of probability of best linear approximation of related block ciphers is shown in Table 2. Therefore, classical differential and

Tab. 2: Comparison of probability for linear characteristic

Name of the Cipher	Block length	Branch no.	Prob. for linear Characteristic
Proposed one	128	17	2^{-192}
SHARK [RDP$^+$96]	64	9	2^{-108}
AES [DR02]	128	5	2^{-150}
KHAZAD [BRb]	64	9	2^{-72}
ANUBIS [BRa]	128	5	2^{-115}

linear attacks are not possible to succeed after four rounds.

6 Conclusions

In this paper, a new technique to design a diffusion layer for an SPN-type block cipher based on an MDS code has been introduced. Scheme to design diffusion layer of lengths 16-bit, 32-bit, 64-bit and 128-bit for an SPN-type block cipher has been proposed. As a case study, 128-bit diffusion layer has been discussed in detailed. The superiority of the proposed diffusion scheme over AES-like diffusion is shown in this paper.

References

[Bir03]   Alex Biryukov. Analysis of involutional ciphers: Khazad and anubis. In *Fast Software Encryption*, volume 2887 of *Lecture Notes in Computer Science*, pages 45–53. Springer, 2003.

[BRa]   P. Barreto and V. Rijmen. The anubis block cipher. Submission to the NESSIE Project.

[BRb]   P. Barreto and V. Rijmen. The khazad legacy-level block cipher. Submission to the NESSIE Project.

[CRCNC97] P. P. Chaudhuri, D. Roy Chowdhury, S. Nandi, and S. Chattopadhyay. *Additive Cellular Automata: Theory and Applications*. IEEE Computer Socity press, 1997.

[Dae95]   J. Daemen. *Cipher and hash function design strategies based on linear and differential cryptanalysis*. PhD thesis, K. U. Leuven, March 1995.

[DLP$^+$09] J. Daemen, M. Lamberger, N. Pramstaller, V. Rijmen, and F. Vercauteren. Computatioal aspects of the expected differential probability of a 4-round aes and aes-like ciphers. *Journal of Computing*, 85:85–104, 2009.

[DR02]   J. Daemen and V. Rijmen. *The Design of Rijndael-AES, The Advanced Encryption Standard*. Springer-Verlag, 2002.

[HLL$^+$01] S. Hong, S. Lee, J. Lim, J. Sung, Cheon D., and I. Cho. Provable security against differential and linear cryptanalysis for the spn structure. In *Fast Software Encryption*, volume 1978 of *Lecture Notes in Computer Science*, pages 273–283, 2001.

[JV04]   Pascal Junod and Serge Vaudenay. Perfect diffusion primitives for block ciphers - building efficient MDS matrices. In *Selected Areas in Cryptography, 11th International Workshop, SAC 2004*, volume 3357 of *Lecture Notes in Computer Science*, pages 84–99, 2004.

[NJA09]   J. Nakahara Jr and E. Abrahao. A new involutory mds matrix for the aes. *Int. Journal of Network Security*, 9:109–116, 2009.

[RDP$^+$96] V. Rijmen, J. Daemen, B. Preneel, A. Bosselaers, and E. De Win. The cipher shark. In *Fast Software Encryption 1996*, volume 1039 of *Lecture Notes in Computer Science*, pages 99–111, 1996.

Chaos in Fuzzy Cellular Automata in Conjunctive Normal Form

David Forrester[1][†] and Paola Flocchini[1][‡]

[1] *School of Electrical Engineering and Computer Science*
University of Ottawa

Fuzzy cellular automata (FCA) are continuous extensions of Boolean cellular automata (CA). Given a Boolean cellular automata rule we can define a corresponding Fuzzy cellular automata rule by allowing real states in $[0, 1]$ and by "fuzzifying" a Boolean form describing the transition function of the CA.

To date, FCA have only been studied in disjunctive normal form (DNF) and their study has revealed interesting properties and links with classical Boolean Cellular Automata.

In this paper, we start the study FCA in conjunctive normal form (CNF). Our main objectives are to see whether the fuzzification of CNF and DNF have similar behaviors and whether, being different representation of the same Boolean truth table, they capture different aspects of their Boolean counterparts. We start the investigation by focussing on FCA from homogeneous configurations, and we classify them analytically. In striking contrast to the periodic behaviours of DNF FCA, we prove that a large class of FCA exhibit chaos in CNF.

Keywords: Fuzzy Cellular Automata, Chaos, Cantor set

1 Introduction

Fuzzy cellular automata (FCA) were introduced in [CFM⁺97] as a particular type of Coupled map lattices [Kan84], that is, a continuous-valued version of elementary discrete cellular automata. FCA employs the concept of *fuzzy logic* [Zad65]. This "many-valued logic" extended the notion of *true* and *false* to include in-between values. Rather than having cells which could assume only binary values, FCA cells could assume any value between zero and one, inclusively. Boolean operators are also "fuzzified" by replacing them with standard algebraic operators. FCA use this process to "fuzzify" the Boolean logic of a corresponding Boolean CA that is expressed in some normal form.

Disjunctive Normal Form (DNF) is one of the standard canonical normal forms for a Boolean expression, consisting of a disjunction of terms, each of which is the conjunction of variables or their negations. FCA in DNF are derived by first representing elementary Boolean CA transition functions in DNF, and then "fuzzifying" the Boolean logic to derive the real-valued function. While the choice of normal form does not change the Boolean function (all normal forms are logically equivalent), different normal forms will produce different fuzzy logic equivalents. Generally, FCA have been studied in DNF. Certain equivalences were shown between Boolean elementary CA their elementary FCA counterparts in DNF [BF11b]. It was also shown that none of the elementary FCA with circular or null boundary conditions exhibited chaos, and all had a periodic asymptotic behaviour [BF11a, BF11b, Min06a, Min06b].

In the study of FCA in DNF, the logic operators $\land$, $\lor$ and *not* are substituted by "fuzzy logic" operators where *not* corresponds to $1 - x$, $\land$ to $x \cdot y$, and $\lor$ to $min\{1, x + y\}$. It turns out that such a choice is special in the sense that it is the only fuzzification that is affine in all its variables. Moreover, this transformation

gives rise to continuous cellular automata with interesting properties. It has been shown in [BF11b] that some conservation properties are preserved through fuzzification; for example, a Boolean cellular automata is number conserving if and only if its corresponding DNF-fuzzy cellular automata is sum-conserving. Another interesting link between the two systems concerns additivity. In fact, it has been shown that a Boolean cellular automata is additive if and only if its corresponding DNF-fuzzy cellular automata is self-oscillating (a particular property of its behavior at infinity). Circular elementary DNF-fuzzy cellular automata have been studied quite extensively; in particular, their asymptotic behavior has been deeply analyzed to see whether there are some rules that exhibit chaotic behavior. It was shown that this was not the case. It turns out that only four behaviors are possible, all of them periodic. Any elementary circular DNF-fuzzy CA asymptotically converges to a periodic behavior of length 1,2,4, or n, where n is the size of the smallest repeating window.

What has not been studied, however, is FCA in the other standard canonical form: *conjunctive normal form* (CNF), where the formula is a conjunction of terms, each of which is the disjunction of variables or their negations. In this paper, we start the investigation of elementary CNF-fuzzy cellular automata. The main goal of this investigation is twofold. First of all, it would be interesting to determine whether chaotic behavior can be observed, a question that also motivated the study of DNF-FCA. Secondly, it would be useful to determine whether the fuzzification of CNF and DNF, being different representation of the same Boolean truth table, could provide insights of different nature on the properties and characteristics of Boolean CAs.

As there is no single and universally accepted definition of chaos, for our purposes, we will use the classical definition proposed by Devaney [BBC+92] in which chaos is identified by three components. Given a continuous map $f : S \to S$ on some metric space S, f is chaotic over S if it is transitive, its periodic points are dense in S, and f depends sensitively on initial conditions.

In our study, the DNF and CNF fuzzifications prove to be quite different. In fact, we show that a large class of CNF-fuzzy rules, to our surprise and in contrast with DNF-fuzzy rules, exhibit chaos even from from homogeneous initial configurations. The observation of the evolution of this class of rules always display a quick convergence to zero. In other words, chaos is not visible to the eyes. However, in the attempt to analytically prove that indeed they always have convergent behavior, we discover that they do converge to zero from an infinite number of initial configurations, but they also have chaotic dynamics on another infinite set. We first show that such rules have both fixed points and periodic configurations. We then observe that certain points exhibit a periodic behaviour of period three, which means that points of any period exist in the dynamics of these rules. We then show that the recursive definition of all points that do not converge to zero produces points that will always lie on the open Cantor set. Finally, we conclude that these rules are chaotic over the open Cantor set and convergent to zero otherwise. The presence of chaos is unexpected, since it has been proven that no such chaos exists in FCA in DNF.

2   Definitions and Notations

2.1   Boolean Cellular Automata

A *cellular automaton* (CA) is a dynamical system which is composed of a regular lattice of *cells* that change their state with time. The concept of Cellular Automaton has been introduced by Von Neumann [von66] and CA have been studied extensively since then (for a recent survey see [Kar05]).

In one dimensional Boolean CA, cells are arranged in a linear array and each of them has a state in $\{0, 1\}$. The system evolves synchronously in discrete time steps by simultaneously updating each cell's state employing a local function that takes into account the state of the cell itself and the states of its neighbourhing cells up to a certain distance (its its *neighborhood*).

A one dimensional bi-infinite cellular automaton with a neighborhood consisting of only itself and its left and right neighbors is known as an *elementary cellular automata*. We will restrict our study to elementary cellular automata.

Elementary CA have only 8 possible local configurations. We can express the local transition rule as a map from $\{0, 1\}^3 \to \{0, 1\}$:

$$(111, 110, 101, 100, 011, 010, 001, 000) \to (r_7, r_6, r_5, r_4, r_3, r_2, r_1, r_0)$$

The set of binary triplets on the left represent all possible inputs to the local transition function. The set of binary numbers $(r_7, \ldots, r_0)$ represents the resultant values after applying the local transition function. These digits concatenated together are known as as the *binary representation* of the CA.

The binary representation of the rule can converted to a decimal number in the standard way: $\sum_{i=0}^{7} 2^i r_i$. This decimal representation is known as the rule's *name* or *number*. Since there are only 8 possible inputs to the local transition function of an elementary CA, and the binary representation of an elementary CA only has 8 digits, the highest possible rule number is $2^8 - 1$, or 255.

A CA can be also represented as a boolean function, and, in particular, as a *normal* logical form (Disjunctive Normal Form or Conjunctive Normal Form). Let us denote by b the binary representation of the rule. Let us denote by b_i the ith digit from the right of b (counting from zero). Let us denote by d_i the tuple mapping to b_i. Finally, let us denote by $d_{i,j}$ the jth digit of d_i from the right (counting from one).

The DNF of a boolean elementary CA is then expressed canonically as:

$$f(x_1, x_2, x_3) = \bigvee_{i|b_i=1} \bigwedge_{j=1:3} x_j^{d_{ij}}$$

where x^0 represents $\neg x$, and x^1 represents x. The CNF of a boolean elementary CA is then expressed canonically as:

$$f(x_1, x_2, x_3) = \bigwedge_{i|b_i=0} \bigvee_{j=1:3} x_j^{1-d_{ij}}$$

where x^0 represents $\neg x$, and x^1 represents x.

> *For example, to find the CNF expression for rule 233, take the binary representation $b = 11101001$, and find $i|b_i = 0$. $b_i = 0$ when $i = 1, 2, 4$. Since there are three 0s in b, the final expression will be a conjunction of three clauses. To find the first clause, take d_1, the tuple mapping to b_1, which is $(0, 0, 1)$. Then apply $\bigvee_{j=1:3} x_j^{1-d_{ij}}$ which gives the clause $(x_1^1 \vee x_2^1 \vee x_3^0)$, which is then written as $(x_1 \vee x_2 \vee \neg x_3)$. Apply the same procedure to find the other two clauses. $d_2 = (0, 1, 0)$, which generates the clause $(x_1 \vee \neg x_2 \vee x_3)$. $d_4 = (1, 0, 0)$, which generates the clause $(\neg x_1 \vee x_2 \vee x_3)$. Finally, conjunct the three clauses together as a function:*
>
> $$f_{233}(x_1, x_2, x_3) = (x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

In the following we will restrict ourselves to homogeneous initial configurations, i.e., bi-infinite configurations of the form: $X^0 = (\ldots, x^0, x^0, x^0, \ldots)$.

2.2 Fuzzy Cellular Automata

A *fuzzy cellular automaton* (FCA) is a cellular automaton whose cells may have real number values in the range of $[0, 1]$, rather than being restricted to values from the binary set $\{0, 1\}$. In a FCA, the state set is $S = [0, 1]$.

Since we are now using real values instead of Boolean values, we can no longer use Boolean operators in our equations. We must "fuzzify" the Boolean operators, which consists of replacing them with standard algebraic operators. The method by which we replace Boolean operators is known as *a logic*. There are many different fuzzy logics available, and each attempts to preserve certain properties of their Boolean counterparts. One property that is generally maintained by all logics is that a fuzzy logic should produce the same result as standard Boolean logic when operating on Boolean values. Some of the more common fuzzy logics used in CA are presented in Table 1.

To convert a Boolean CA into a fuzzy CA, we first convert it into one of the normal forms from the previous sections, then we apply convert the Boolean operators to standard operators by choosing one of the logics from the above table.

> *For example, to find the formula for elementary fuzzy CA rule 200 in disjunctive normal form using CFMS logic, we first convert Boolean rule 200 into DNF:*
>
> $$f_{200}(x_1, x_2, x_3) = (\neg x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge \neg x_3) \vee (x_1 \wedge x_2 \wedge x_3)$$

Logic	$\neg x$	$x \wedge y$	$x \vee y$
CFMS	$1 - x$	$x \cdot y$	$min\{1, x + y\}$
Probabilistic	$1 - x$	$x \cdot y$	$x + y - x \cdot y$
Lukasiewicz	$1 - x$	$max\{0, x + y - 1\}$	$min\{1, x + y\}$
Gödel	if $x = 0$ then 1, else 0	$min\{x, y\}$	$max\{x, y\}$
Zadeh	$1 - x$	$min\{x, y\}$	$max\{x, y\}$
Product	if $x = 0$ then 1, else 0	$x \cdot y$	$x + y - x \cdot y$

Table 1: Common Fuzzy Logics.

Then we apply the logic from table 1 to the above formula, which produces:

$$f_{200}(x_1, x_2, x_3) = min\{1, ((1 - x_1) \cdot x_2 \cdot x_3) + (x_1 \cdot x_2 \cdot (1 - x_3)) + (x_1 \cdot x_2 \cdot x_3)\}$$

This can then be simplified using standard algebra to

$$f_{200}(x_1, x_2, x_3) = min\{1, (x_1 \cdot x_2 + x_2 \cdot x_3 - x_1 \cdot x_2 \cdot x_3)\}$$

Note that we restrict the result to ≤ 1 by using the min function. It was shown in [CFM$^+$97] that this was not necessary with DNF fuzzification, as no terms ever exceeded 1. This assumption is not valid with CNF, so the min function must be used.

3   CNF-fuzzy cellular automata

3.1   *The four classes of behaviors*

The following table may be used in the construction of the local transition function of an elementary CNF FCA. The presence of a 0 in a particular column of the rule's binary representation indicates the presence of the given factor in its transition function.

Column	**Transition**	**Factor**
1	$(0, 0, 0) \rightarrow 0$	$min\{1, (x_1 + x_2 + x_3)\}$
2	$(0, 0, 1) \rightarrow 0$	$min\{1, (x_1 + x_2 - x_3 + 1)\}$
4	$(0, 1, 0) \rightarrow 0$	$min\{1, (x_1 - x_2 + x_3 + 1)\}$
8	$(0, 1, 1) \rightarrow 0$	$min\{1, (x_1 - x_2 - x_3 + 2)$
16	$(1, 0, 0) \rightarrow 0$	$min\{1, (-x_1 + x_2 + x_3 + 1)\}$
32	$(1, 0, 1) \rightarrow 0$	$min\{1, (-x_1 + x_2 - x_3 + 2)\}$
64	$(1, 1, 0) \rightarrow 0$	$min\{1, (-x_1 - x_2 + x_3 + 2)\}$
128	$(1, 1, 1) \rightarrow 0$	$min\{1, (-x_1 - x_2 - x_3 + 3)\}$

Table 2: FCA CNF Factors

Recall from table 2 that the local transition function $f(x_1, x_2, x_3)$ of a FCA in CNF is a product of factors whose presence (or absence) is determined by the presence (or absence) of transitions that go to 0. In the case of a homogeneous configuration, $x_1 = x_2 = x_3 = x$, so we can simplify the products from table 2. We can further simply the factors by applying the knowledge that the value of any cell in the CA is in the range $[0, 1]$. (We impose this restriction on the initial configuration, and we ensure this property is preserved by carefully choosing the transition function, as discussed in [FC08].) Table 3 below simplifies the factors from table 2 in the homogeneous case:

The presence (or absence) of factors of 1 do not affect the resultant value of a function of products. We can therefore safely ignore factors that go to 1. Therefore, we can say that in the homogeneous case, the local transition rule of FCA in CNF can be determined solely by the presence (or absence) of the following two

Transition	CNF Factors	Factors When $x_1 = x_2 = x_3 = x$	Factors When $0 \le x \le 1$
$(0,0,0) \to 0$	$min\{1, (x_1 + x_2 + x_3)\}$	$min\{1, (3x)\}$	$min\{1, (3x)\}$
$(0,0,1) \to 0$	$min\{1, (x_1 + x_2 - x_3 + 1)\}$	$min\{1, (1+x)\}$	1
$(0,1,0) \to 0$	$min\{1, (x_1 - x_2 + x_3 + 1)\}$	$min\{1, (1+x)\}$	1
$(0,1,1) \to 0$	$min\{1, (x_1 - x_2 - x_3 + 2)\}$	$min\{1, (2-x)\}$	1
$(1,0,0) \to 0$	$min\{1, (-x_1 + x_2 + x_3 + 1)\}$	$min\{1, (1+x)\}$	1
$(1,0,1) \to 0$	$min\{1, (-x_1 + x_2 - x_3 + 2)\}$	$min\{1, (2-x)\}$	1
$(1,1,0) \to 0$	$min\{1, (-x_1 - x_2 + x_3 + 2)\}$	$min\{1, (2-x)\}$	1
$(1,1,1) \to 0$	$min\{1, (-x_1 - x_2 - x_3 + 3)\}$	$min\{1, (3-3x)\}$	$min\{1, (3-3x)\}$

Table 3: Homogeneous FCA CNF Factors

transitions: $(1,1,1) \to 0$, and $(0,0,0) \to 0$. Furthermore, since there are only four combinations of the above two transitions being present or not, we can divide all homogeneous rules canonically into four classes.

In table 4, we define the following four classes of homogeneous CNF FCA, along with their local transition function (note: The binary representation of a class of rules includes a third digit, $*$, which represents indifferently 1 or 0):

Class	Binary	Rules r_z	$f(x, x, x)$
A	$1******1$	$z \in [128, 256), z \bmod 2 \neq 0$	1
B	$1******0$	$z \in [128, 256), z \bmod 2 = 0$	$min\{1, (3x)\}$
C	$0******1$	$z \in [0, 128), z \bmod 2 \neq 0$	$min\{1, (3-3x)\}$
D	$0******0$	$z \in [0, 128), z \bmod 2 = 0$	$min\{1, (3-3x)\} \cdot min\{1, (3x)\}$

Table 4: Homogeneous FCA CNF Classes

The first three classes exhibit simplistic dynamics: The first class, A, contains all rules whose decimal representation is odd and greater than 128: all these rules converge directly to one. The second class, B, contains all even rules greater than 127; such rules have two fixed points, and converge either to zero or to one. The third class, C, has a fixed point at $\frac{3}{4}$, but all other values converge to a temporally periodic configuration that alternates between zero and one; this class contains all odd rules smaller than 128. Finally, in our simulations the rules of Class D always display a quick convergence to zero; however, the analytical analysis, which is more complicated than the one employed for the other classes, shows otherwise.

3.2 Chaotic Behavior of Class D

We first remind the classical definition of chaos by Devaney [BBC+92]. Given a continuous map $f : S \to S$ on some metric space S, f is chaotic over S if it is transitive, its periodic points are dense in S, and f depends sensitively on initial conditions. Transitivity implies that for any non-empty open subsets U and V of S there exists a t such that $f^t(U) \cap V$ is not empty; in other words, there exists a time when an orbit from a point in U reaches V, for any U and V. Density of the periodic points is an element of regularity and means that any point is arbitrarily close to a periodic point. Finally, transitivity to initial conditions means that starting from two arbitrarily close points in the state space, the orbits created by iterations of f become arbitrarily far from each other.

Class D rules include all rules with the transitions $(0,0,0) \to 0$ and $(1,1,1) \to 0$, that is, all even rules smaller than 128. Class D rules have the local transition function of $f(x,x,x) = min\{1, (3-3x)\} \cdot min\{1, (3x)\}$. It is a product of Class B and Class C.

Let $X = (\ldots, x, x, x, \ldots)$ be a bi-infinite homogeneous configuration, and let $f(x,x,x)$ be the local function $f : [0,1]^3 \to [0,1]$. Let F be the corresponding global function $F : [0,1]^\infty \to [0,1]^\infty$. Since we are restricting our study to homogeneous initial configurations, the global function F will always produce homogeneous configurations; the evolution of the global function is then equivalent to the evolution of the local function

$f(x, x, x)$, which is always acting on three identical values. We will then denote for simplicity by $f(x)$, the local function $f(x, x, x)$ on homogeneous values noticing that the dynamics of $f : [0, 1] \to [0, 1]$ is also describing the dynamics of $F : [0, 1]^\infty \to [0, 1]^\infty$. We have:

$$f(x) \begin{cases} 3x, & 0 \le x \le \frac{1}{3} \\ 1, & {}^1/_3 \le x \le \frac{2}{3} \\ 3 - 3x, & \frac{2}{3} \le x \le 1 \end{cases}$$

In this Section we prove that f is Chaotic on a subset of $[0, 1]$. Instead of proving it directly using the definition of chaos, we prove it by showing that it is conjugate to the typical chaotic map: the shift map.

To study the state space of this function, we start by determining its fixed points and by calculating (if they exist) the points of small periods (2 or 3).

Fixed points and periodic points.   By solving $f(x, x, x) = x$, it is easy to determine the fixed points of function f: $x = 0$ and $x = {}^3/_4$. Analogously one could determine all points of period 2 and the ones of period 3 (see Tables 5 and 6).

$\boldsymbol{x}$	$\boldsymbol{f(x, x, x)}$	$\boldsymbol{f[f(x, x, x)]}$	$\boldsymbol{f[f(x, x, x)] = x}$
$0 \le x \le {}^1/_9$	$3x$	$9x$	$x = 0$
${}^2/_9 \le x \le {}^3/_9$	$3x$	$9 - 9x$	$x = {}^9/_{10}$
${}^6/_9 \le x \le {}^7/_9$	$3 - 3x$	$9x - 6$	$x = {}^3/_4$
${}^8/_9 \le x \le 1$	$3 - 3x$	$3 - 9x$	$x = {}^3/_{10}$

Table 5: Class D Period 2

$\boldsymbol{x}$	$\boldsymbol{f(x, x, x)}$	$\boldsymbol{f[f(x, x, x)]}$	$\boldsymbol{f\{f[f(x, x, x)]\}}$	$\boldsymbol{f\{f[f(x, x, x)]\} = x}$
$0 \le x \le {}^1/_{27}$	$3x$	$9x$	$27x$	$x = 0$
${}^2/_{27} \le x \le {}^3/_{27}$	$3x$	$9x$	$3 - 27x$	$x = {}^3/_{28}$
${}^6/_{27} \le x \le {}^7/_{27}$	$3x$	$3 - 9x$	$27x - 6$	$x = {}^3/_{13}$
${}^8/_{27} \le x \le {}^9/_{27}$	$3x$	$3 - 9x$	$9 - 27x$	$x = {}^9/_{28}$
${}^{18}/_{27} \le x \le {}^{19}/_{27}$	$3 - 3x$	$9x - 6$	$27x - 18$	$x = {}^9/_{13}$
${}^{20}/_{27} \le x \le {}^{21}/_{27}$	$3 - 3x$	$9x - 6$	$21 - 27x$	$x = {}^3/_4$
${}^{24}/_{27} \le x \le {}^{25}/_{27}$	$3 - 3x$	$9 - 9x$	$27x - 24$	$x = {}^{12}/_{13}$
${}^{26}/_{27} \le x \le 1$	$3 - 3x$	$9 - 9x$	$27 - 27x$	$x = {}^{27}/_{28}$

Table 6: Class D Period 3

By eliminating fixed points we can see that there is only one set of values of period 2: $f((0.3)) = (0.9)$ and $f((0.9)) = (0.3)$, and two sets of period 3:

- $f(({}^3/_{28})) = ({}^9/_{28})$, $f(({}^9/_{28})) = ({}^{27}/_{28})$, and $f(({}^{27}/_{28})) = ({}^3/_{28})$

- $f(({}^6/_{26})) = ({}^{18}/_{26})$, $f(({}^{18}/_{26})) = ({}^{24}/_{26})$, and $f(({}^{24}/_{26})) = ({}^6/_{26})$

Note that, by the Sharkovskii's theorem citeintroDynamics, the presence of configurations of period 3 guarantees the presence of configurations of all periods.

We can also deduce some values that will eventually converge to these periodic points. For example, it is easy to show that any value of the form $x = \frac{0.3}{3^k}$, where $k \ge 0$ will be transformed into 0.3 after undergoing the function $f^k(x, x, x) = 3x$. While it is easy to identify some points eventually converging to one of these periodic orbits, it does not seem easy to proceed in this way to fully understand the structure of the state space.

Points not converging to zero. In the following we wish to start by determining the set of values that *do not converge to zero*. To do so we will proceed recursively by elimination. We will begin with the open interval $(0, 1)$, and subtract the set of all values that converge to the fixed point 0 in two steps; by definition, these are the values of $^1/_3 \leq x \leq {}^2/_3$ (for such values $f(x) = 1$ and $f^2(x) = 0$).

We now compute the range of values that converge to zero in three steps: x such that $^1/_3 \leq f(x, x, x) \leq {}^2/_3$; that is $^1/_9 \leq x \leq {}^2/_9$ and $^7/_9 \leq x \leq {}^8/_9$. Now we can compute the range of values x such that $^1/_3 \leq f^2(x, x, x) \leq {}^2/_3$. When we delete all these values from $(0, 1)$, we have the space depicted in Figure 1.
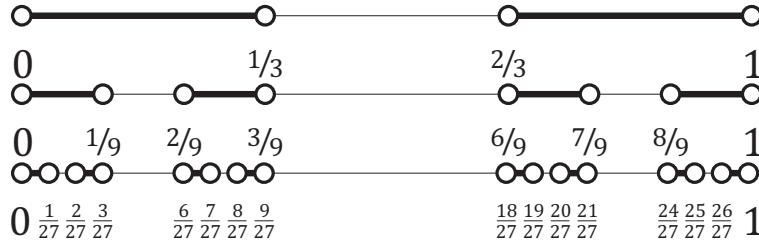
Figure 1: Structure of the space of values not converging to zero.

Continuing recursively in this way, we can easily see that the set of values that *do not converge to $x = 0$* corresponds to the *open Cantor set*.

Notice that this is the classical definition of Cantor Set except that we do not include the extremes of the set and we obtain in this way an open set. Before discussing the properties of the open Cantor set, we recall the notion of *ternary expansion*.

Ternary Expansions. A sequence of integers $0.a_1 a_2 a_3 \ldots$ where each a_i is either 0, 1, or 2 is called the ternary expansion of x if

$$x = \sum_{i=1}^{\infty} \frac{a_i}{3^i}$$

A real number could have different ternary expansions (for example, $\frac{1}{3}$ has expansion $0.10000\ldots$, but also $0.022222\ldots$). In fact, one can see that all rational numbers of the form $\frac{p}{3^k}$, for some integer $0 \leq p < 3^k$, have more than one ternary expansion, while all other numbers have a unique one.

Note that if x has ternary expansion $0.a_1 a_2 \ldots$, the digit a_1 determines to which third of the interval $[0, 1]$, x lies. If $a_1 = 0$ then $x \in [0, \frac{1}{3}]$, if $a_1 = 1$, then $x \in [\frac{1}{3}, \frac{2}{3}]$, if $a_1 = 2$, then $x \in [\frac{2}{3}, 1]$. Once determined in which third of $[0, 1]$ x belongs, the second digit a_2 indicates recursively in which third of that subinterval x lies. So, all numbers with ambiguous representation are the extremes of the intervals removed at each step during the construction of the Cantor set.

Chaos over the Cantor set. The open Cantor set (here denoted by K) is still uncountable and most properties of the classical "closed" Cantor set are preserved. Useful properties that are easy to verify are the following [Dev03, Dev92]:

Property 1 *The open Cantor set contains all real numbers in $(0, 1)$ for which any ternary expansion contains no 1s.*

Property 2 *A ternary expansion that does not contain any 1s has an equivalent ternary expansion containing some 1s if and only if it is of the form:* $0.a_1 \ldots a_n 0222 \ldots$ *or* $0.a_1 \ldots a_n 200 \ldots$

So, the Cantor set K contains all real numbers in $(0, 1)$ whose ternary expansion contains no 1s and does not terminate with infinite 0s nor infinite 2s.

Based on the above properties, we now show that if we apply function f to a real number in K, the result is still in K.

Lemma 1 *Let $x \in K$. Then we have:* $f(x) \in K$.

Proof: Let $0.a_1a_2a_3\ldots$ be a ternary expansion of $x \in K$. Applying f we either obtain $3x$ (if $0 < x < \frac{1}{3}$) or $3(1-x)$ (if $\frac{2}{3} < x < 1$). By definition of ternary expansion, if $0 < x < \frac{1}{3}$ we have that $a_1 = 0$ and the ternary expansion of $3x$ is $0.a_2a_3a_4\ldots$, which is still in K. Moreover, if $\frac{2}{3} < x < 1$ we have that $a_1 = 2$, which means that $\bar{a}_2 = 0$ and the ternary expansion of $3(1-x)$ is $0.\bar{a}_2\bar{a}_3\bar{a}_4\ldots$, which is also still in K because $\bar{a}_i$ cannot be 1 (it is either 2 or 0). So, in both cases $f(x) \in K$.   □

Let us now introduce the well known concept of symbolic dynamics and of shift map. Let $\Sigma = \{(s_0s_1s_2\ldots) : s_j = 0 \text{ or } 1\}$ be the sequence space on two symbols, each sequence in Σ being an infinite sequence composed of 0s and 1s. Let $\sigma : \Sigma \to \Sigma$ denote the *shift map* defined as follows:

$$\sigma(s_0s_1s_2\ldots) = (s_1s_2s_3\ldots)$$

It is well known that the shift map is a chaotic dynamical system on Σ; in fact its periodic points are dense, it is transitive, and it depends sensitively on initial conditions.

Theorem 1 *The shift map σ on Σ is conjugate to f on K.*

Proof: We know by Lemma 1 that $f : K \to K$. We have now to show that there exists an homeomorphism $S : K \to \Sigma$ such that $S \circ f(x) = \sigma \circ S(x)$ for any $x \in K$; that is, that the following diagram commute:

$$
\begin{array}{ccc}
 & f & \\
K & \longrightarrow & K \\
S \downarrow & & \downarrow \; S \\
 & \sigma & \\
\Sigma & \longrightarrow & \Sigma
\end{array}
$$

In the following, we will indicate the real numbers belonging to K in their ternary expansion. Let $S : K \to \Sigma$ be a function defined as follows:

$$S(0.a_1a_2a_3\ldots) = \begin{cases} a_1'a_2'a_3'\ldots & \text{if } a_1 = 0 \\ \bar{a}_1{'}\,\bar{a}_2{'}\,\bar{a}_3{'}\ldots & \text{if } a_1 = 2 \end{cases} \tag{1}$$

where

$$a_i' = \begin{cases} 0 & \text{if } a_i = 0 \\ 1 & \text{if } a_i = 2 \end{cases}$$

and

$$\bar{a}_i{'} = \begin{cases} 1 & \text{if } a_i = 0 \\ 0 & \text{if } a_i = 2 \end{cases}$$

For example: $S(0.2022202020202020\ldots) = 010001010101010\ldots$, while $S(0.0022002200220\ldots) = 0011001100110011\ldots$

It is easy to see that S is a well defined homeomorphism.

Consider $x \in K$. By the properties of K, we can write x as a ternary expansion not containing any 1 and not terminating with infinite 0s nor infinite 2s. Let $x = 0.a_1a_2a_3\ldots$ with $a_i = 0$ or 2. Consider $f(x)$. By definition of f, and by definition of ternary expansion, if $a_1 = 0$ then $x \in (0, \frac{1}{3})$ and thus $f(x) = 3x$ and its ternary expansion can be written as $0.a_2a_3\ldots$. On the other hand, if $a_1 = 2$ then $x \in (\frac{2}{3}, 1)$, thus $f(x) = 3(1-x)$ and its ternary expansion can be written as $0.\bar{a}_2\bar{a}_3\ldots$. in other words, we have:

$$f(x) = \begin{cases} 0.a_2a_3\ldots & \text{if } a_1 = 0 \\ 0.\bar{a}_2\bar{a}_3\ldots & \text{if } a_1 = 2 \end{cases} \tag{2}$$

On the other hand, by definition of S we have that $S \circ f(x)$ is the sequence of digits of $f(x)$ after the 0 where, if $a_1 = 0$ every 2 is replaced by 1, if $a_1 = 2$, every 2 is replaced by 0 and every 0 by 1. Such a binary infinite sequence is precisely $\sigma \circ S(x)$.   □

Since it is well known that σ is chaotic on Σ, we can conclude that:

Theorem 2 *Rule f is chaotic on the open Cantor set.*

So, we can conclude that all elementary fuzzy cellular automata in conjunctive normal form with the transitions $(0, 0, 0) \rightarrow 0$ and $(1, 1, 1) \rightarrow 0$ in the rule table are chaotic on all homogenous configurations $(x)^n$ with $x \in K$, and converge to zero otherwise.

4   Conclusion

We began our investigation with the realization that FCA CNF "fuzzification", being a conjunction of disjunctions, produce formulae with too many terms to be easily analyzed. In order to facilitate analysis, we began by restricting our investigation to the special case of a homogeneous initial configuration.

The case of homogeneous configurations is much easier to analyse that the general case of heterogeneous initial configurations, because the global rule of the CA can be studied as a simple function from $[0, 1]$ to $[01]$. We discovered that all 256 such FCA fall into one of four classes. We were able to solve the asymptotic behaviour of each of these classes.

The first three classes exhibited very simple dynamics: all rules converge directly to one. The second class has two fixed points, and all rules converge either to zero or to one. The third class has a fixed point at $\frac{3}{4}$, but all other values converge to a temporally periodic configuration that alternates between zero and one.

The final, most interesting class exhibits chaos. In fact we showed that all CNF-fuzzy rules in this class display chaotic behavior on a subset of $[0, 1]$ even with homogeneous initial configurations. This is in striking contrast with the periodic behaviors of all DNF-fuzzy rules.

What is not known, is whether the theorems we prove in the homogeneous case are applicable in the general case. In particular, does chaos exist in CNF the non-homogeneous case ? It is also unclear to us why certain classes in the homogeneous and heterogeneous cases seem to be identical, while other classes have no apparent correlation. We leave these as open questions.

Further research is under way also regarding the usefulness of the CNF-fuzzification for better understanding Boolean CAs. The current investigation is concerned only with CNF fuzzy rules with homogeneous initial configurations; such a setting is very restrictive and the use of CNF does not allow to differentiate enough among the rules. This makes it impossible to infer meaningful information regarding the dynamics of their Boolean counterpart. The more general study of heterogeneous configurations might disclose interesting links.

Acknowledgements

References

[BBC+92]   J. Banks, J. Brooks, G. Cairns, G. Davis, and P. Stacey. The american mathematical monthly. *On Devaney's definition of chaos*, 99(4):332–334, 1992.

[BF11a]   H. Betel and P. Flocchini. On the asymptotic behavior of fuzzy cellular automata. *Journal of Cellular Automata*, 6:25–52, 2011.

[BF11b]   H. Betel and P. Flocchini. On the relationship between boolean and fuzzy cellular automata. *Theoretical Computer Science*, 412(8-10):703–713, 2011.

[CFM+97]   G. Cattaneo, P. Flocchini, G. Mauri, C. Quaranta Vogliotti, and N. Santoro. Cellular automata in fuzzy backgrounds. *Physica D: Nonlinear Phenomena*, 105(1-3):105–120, 1997.

[Dev92]   R. Devaney. *A First Course in Chaotic Dynamical Systems: Theory and Experiment*. Addison-Wesley, 1992.

[Dev03]   R. Devaney. *An Introduction to Chaotic Dynamical Systems, 2nd Edition*. Westview Press, 2003.

[FC08]   P. Flocchini and V. Cezar. Radial view of continuous cellular automata. *Fundamenta Informaticae*, 87(2):165–183, 2008.

[Kan84]   K. Kaneko. Quasiperiodicity in antiferro-like structures and spatial intermittency in coupled logistic lattice. *Progress of Theoretical Physics*, 72, 1984.

[Kar05]   J. Kari. Theory of cellular automata: A survey. *Theoretical Computer Science*, 334(1-3):3–33, 2005.

[Min06a]   A. Mingarelli. The global evolution of general fuzzy automata. *Journal of Cellular Automata*, 1:141–164, 2006.

[Min06b]   A. B. Mingarelli. A study of fuzzy and many-valued logics in cellular automata. *Journal of Cellular Automata*, 1(3):233–252, 2006.

[von66]   J. von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, 1966.

[Zad65]   L. A. Zadeh. Fuzzy sets. *Information Control*, 8:338–353, 1965.

Cellular automata-based model with synchronous updating for Task Static Scheduling

Murillo G. Carneiro and Gina M. B. de Oliveira

Universidade Federal de Uberlandia
Pós-Graduação em Ciência da Computação
Avenida João Naves de Ávila, 2121, Santa Mônica
38400-902 Uberlândia, MG - Brazil

Task Static Scheduling Problem (TSSP) in multiprocessors is an NP-Complete problem. Approaches proposed to solve it typically use heuristics or meta-heuristics. Previous works have shown the promising use of Cellular Automata (CA) for extraction and reuse of knowledge in TSSP. However, they have not exploited the massive parallelism inherent to CA because good results were obtained only using asynchronous updating of cells. This paper presents a new model called Synchronous CA-based Scheduler (SCAS) that uses parallel updating of cells. Aiming to compare and analyze SCAS, related works were reproduced. Program graphs found in the literature and randomly generated ones were used in experiments. Experiments showed that SCAS improved previous works in terms of quality of extracted knowledge.

Keywords: SCAS, CA-based scheduler, synchronous updating, Task Static Scheduling

1 Introduction

Scheduling is a decision-making process that involves resources and tasks in the search for optimize one or more objectives Pinedo (2008). Resources can be machines in a workshop or processing units in a computing environment, while tasks can be operations in a production process, executions of computer programs, and so on. There are several applications such as production scheduling, employees scheduling and computational tasks scheduling.

  Task scheduling aims to allocate a set of computational tasks that compose a parallel application in the nodes of a multiprocessor architecture. Considering Task Static Scheduling Problem (TSSP), all information about the tasks is known a priori. An optimal solution for an instance of TSSP is such that the precedence constraints are satisfied and the runtime - or *makespan* - is minimized. The problem is NP-Complete, even limited to the simplest case: a parallel system with only two processors Garey and Johnson (1979). Furthermore, it is a challenge for many researchers. The proposed approaches to solve it typically employ heuristics or meta-heuristics. Some of the most known heuristics to TSSP are: HLFET (Hightest Level First with Estimated Time), ISH (Insertion Scheduling Heuristic) and MCP (Modified

Critical Path), Kwok and Ahmad (1999) and Jin et al. (2008). However, such heuristics do not have the ability to extract knowledge of the scheduling process of a parallel application and reuse it in other instances.

Cellular Automata (CA) are discrete dynamical systems which have the potential to exhibit a complex global behavior from simple interactions between local units. The most investigated update mode of cells is synchronous - or parallel - because it explores the inherent parallelism in CA. Previous works pointed to the promising use of CA-based approaches to TSSP Seredynski and Zomaya (2002), Swiecicka et al. (2006) and Vidica and Oliveira (2006). Such models combine the use of CA and Genetic Algorithms (GA) Goldberg (1989) due to the employment of transition rule spaces with high cardinality Mitchell et al. (1996). However, they have not exploited the massive parallelism inherent to CA because good results were obtained only using asynchronous updating of cells.

The main objective of this work is to present and evaluate a new model with synchronous updating of cells called Synchronous CA-based Scheduler (SCAS). An important feature of SCAS is that it is suitable to implement in parallel hardware. In addition, the new scheduler should be able to perform the optimal (or at least sub-optimal) scheduling of tasks. Other important feature investigated in SCAS is its ability to extract knowledge during the process of scheduling of an application and to reuse it while solving other instances of TSSP. Furthermore, the results obtained in SCAS are compared to results obtained in reproductions of previous works.

The remainder of this paper is organized as follows: Sections 2 presents a background about multiprocessor scheduling. Section 3 and Section 4 contain concepts about CA-based scheduling and descriptions of the proposed system, respectively. Section 5 contains experimental results concerning CA applied to scheduling in two processor systems. The last section contains conclusions and future works.

2   Multiprocessor Scheduling

A multiprocessor system can be represented by an undirected and not weighted graph $G_s = (V_s, E_s)$, called system graph. V_s is the set of P processors of the system graph and E_s is the set of bi-directional channels between processors that define the topology of the multiprocessor system. In this model it is also assumed that all processors have the same computational power and the communications between the channels do not consume any extra time of the processor beyond the communication time between tasks as specified in graph.

A parallel application can be represented by a directed acyclic graph (DAG) defined by $G = (V, E, W, C)$, where $V = \{ t_1, \ldots, t_N \}$ denotes the set of N graph tasks; $E = \{ e_{i,j} \mid t_i, t_j \in V \}$ represents the set of communication edges, also called precedence constraints; $W = \{ w_1, \ldots, w_N \}$ represents the set of run times of the tasks, in others words, for each task $t \in V$ a computational weight $w(t) \in W$ is assigned relative to its computational cost; and $C = \{ c_{i,j} \mid e_{i,j} \in E \}$ denotes the set of communication times of the edges, in others words, for each edge $e_{i,j} \in E$ is assigned a communication cost $c_{i,j} \in C$ related to the cost of data transfer between tasks t_i and t_j when they are carried out on different processors. The set of edges E defines the precedence relations between tasks. So, a task cannot be executed unless all its predecessors complete their executions and all relevant data are available. Tasks are represented by nodes. Tasks without predecessors will be called starting tasks and tasks without successors will be called exit tasks. Preemption of tasks and redundant executions are not allowed. G is called the precedence graph of tasks, or simply program graph. Figure 1 shows an example of program graph called *gauss18* that represents a set of 18 tasks.
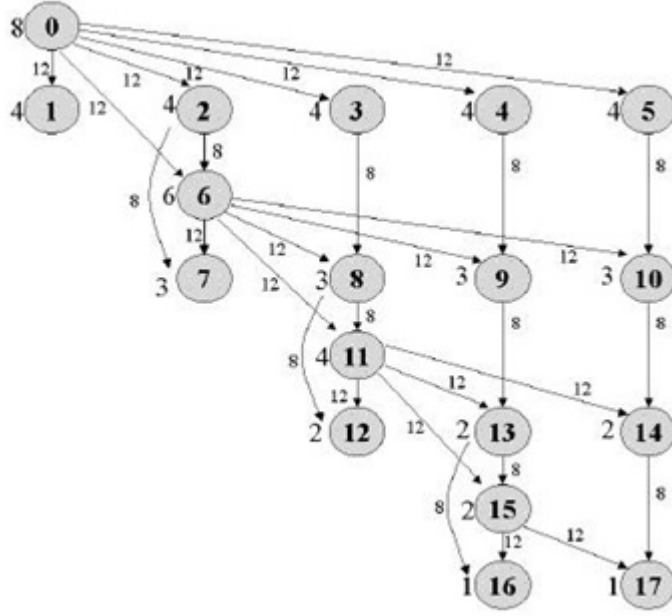
Fig. 1: Example of program graph (*gauss18*).

A scheduling policy defines tasks execution order in each processor. Note that while the scheduler distributes tasks among processors the scheduling policy orders these tasks within each processor. A scheduling policy was used for all tests: the task with the highest dynamic b-level first. The level (b-level) of a task in program graph is the highest cost between it and a exit task of graph, thus the level of a task i can be recursively calculated by:

$$bl_i = \left\{ \begin{array}{l} w_i, \text{ if } i \text{ is a exit task;} \\ \max_{j \in successors(i)} (bl_j + c_{i,j}) + w_i, \text{ otherwise.} \end{array} \right\}$$

The level of a task is dynamic when it is calculated considering the allocation of the tasks in processors and the communication cost is just considered when tasks are distributed on different processors.

3 Previous CA-based Schedulers

Considering the CA-based scheduler model proposed in Seredynski and Zomaya (2002), it is assumed that each cell of the lattice is associated with a task of the program graph. Thus, if a set of tasks has cardinality x, the CA lattice must have x cells. Furthermore, given an architecture consisting of P processors, the CA will have P possible states. Assuming a system with two processors ($P0$ and $P1$), each cell in the lattice can be in state 0, indicating that the corresponding task is allocated on processor $P0$, or state 1, indicating that the task is allocated on processor $P1$. For example, a program graph composed of four tasks should be represented by a lattice of 4 cells and considering a configuration where the tasks 0 and 3 are allocated

in $P0$ and tasks 1 and 2 in $P1$, the lattice will be 0110. To calculate the scheduling time T for the lattice it is necessary to use a scheduling policy that defines tasks execution order in each processor.

In Seredynski and Zomaya (2002) was presented a CA-based scheduler that uses nonlinear neighborhoods and operates in two modes: learning and normal operating. In the learning mode, the scheduler uses a Genetic Algorithm (GA) to discover rules of CA that can find optimal solutions (or sub-optimal) for random instances (initial configurations) of a program graph. The initial population of the GA is composed by randomly generated transition rules. The fitness function of each transition rule is calculated by: (i) randomly sorting a set of lattices S_{Latt} that represent initial allocations of tasks in processors; (ii) updating lattice applying the transition rule for t time steps starting for each initial configuration of S_{Latt}; (iii) final lattices are scheduled with support of a predefined scheduling policy and the average of scheduling cost for each rule is obtained. The best rule presents the smallest average. Figure 2 shows the major steps of GA that uses elitist strategy, where the set E of best rules are maintained for the next generation and only them are considered in parent selection for crossover.

"Learning Mode"
Step 0 Randomly sorting a population of P rules;
Step 1 While (!condition_finish) // do Step 2-5
Step 2 Compute the fitness of the P rules // do Step (i)-(iii)
Step 3 Selecting pairs of rules in P using elitism strategy E;
Step 4 Applying the single-point crossover operator to pairs selected generating P-E rules;
Step 5 Submit the P-E rules to mutation;
Step 6 P is stored in rules repository.

Fig. 2: GA used in learning mode.

In normal operation mode it is expected that, for any initial allocation of tasks, the rules of CA are able to minimize the *makespam*. It is also expected that the rules obtained in the learning phase can be used in the scheduling of other graphs.

In Swiecicka et al. (2006) was presented a CA-based scheduler that uses linear neighborhood and operates in three modes: learning, normal operation and reuse. The first two modes are similar to those proposed by Seredynski and Zomaya (2002). In reuse mode, the previously discovered rules are reused with the help of an artificial immune system (AIS) to solve new problem instances.

4 Synchronous CA-based Scheduler (SCAS)

Locality of cellular interactions, simplicity of basic components (cells) and possibility of implementation on parallel hardware are among the most notable features of cellular automata Sipper (1997). Experiments performed using previous models on the literature, Seredynski and Zomaya (2002), Swiecicka et al. (2006) e Vidica and Oliveira (2006), have found that CAs with asynchronous updating (only one cell can update its state at a time) performed much better than synchronous mode. However, the large capacity

of parallelism inherent to CA is lost using asynchronous updating of cells Seredynski and Zomaya (2002). Thus, a new model of CA-based scheduler able to explore efficiently the parallelism in CA is proposed here. It was called Synchronous CA-based Scheduler (SCAS).

SCAS employs linear neighborhood for three reasons: simplicity, low computational cost and arbitrary number of processors. In previous works Seredynski and Zomaya (2002) observed that nonlinear neighborhoods presented best results than linear ones, however they are limited to multiprocessor architecture with two nodes and they have a complex structure. For example, the nonlinear neighborhood investigated in Seredynski and Zomaya (2002) defines rules with size equal to 250 bits, while using linear neighborhood with radius 3, the size of rule is 128 bits. Besides, the nonlinear models investigated use only two processors in system graph and a generalization of these models to a higher number of processors is very complex and it would lead to rules with larger size. On the other hand, using linear neighborhood it is possible to increase the number of processors in system graph simply using more states by cell.

Another important feature concerns the null boundary condition used in previous models. In SCAS, the cells to the left of first cell are considered in state 0 while the cells to the right of last cell are considered in state 1, different from other models that use state 0 in both sides. These values were defined by an analysis on the influence of boundary condition in which it was concluded that such condition (0,1) offers a more balanced boundary: it uses the output bits of the transition rule in a more distributed way along the temporal updating of lattice.

In addition, the genetic algorithm in SCAS employs a different approach from the previous models: it does not use elitist strategy. In SCAS, the selection is made by simple tournament ($Tour = 2$) and reinsertion is based on fitness of rules. In other words, the total population (parents and children) is ordered and the best rules are selected. The purpose of these changes is to stimulate the competition between individuals of the population to allow a broad search in the space of possible solutions and consequently generating a more efficient set of rules in the final population, which is very difficult to obtain using an elitist strategy.
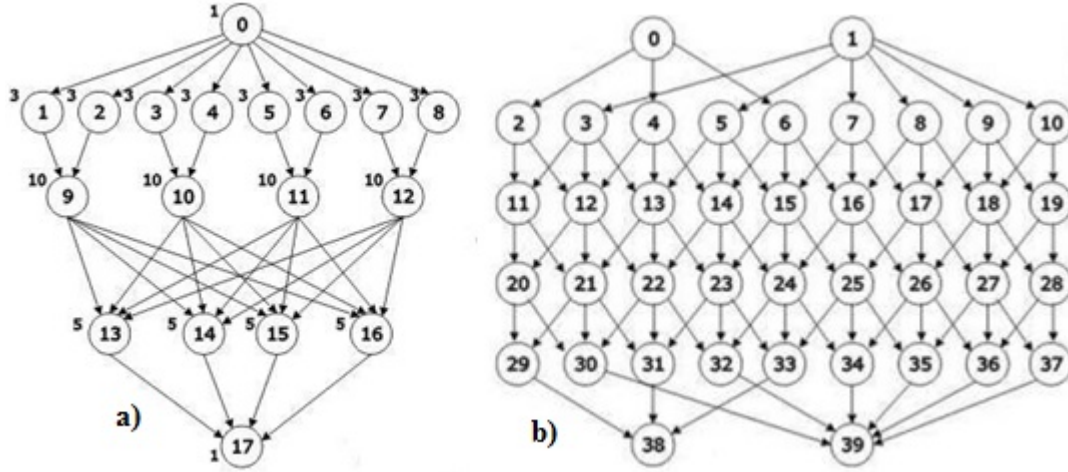
In Figure 3 is presented a framework of SCAS. SCAS receive as input a program graph and a system graph. In learning mode, as well as in Seredynski and Zomaya (2002), a GA is used to search for CA rules able to find optimal scheduling for the program graph. In execution mode, program graph is loaded in IC (randomly initial configurations) and CA is equipped with a rule of RDB. Then, CA synchronously update the lattice by t time steps obtaining the final allocation of tasks. This allocation is submitted to scheduling policy. Finally, scheduling time is calculated for these allocations.

5   Experiments

Experiments to evaluate SCAS performance are presented in this section. The goal is to compare the results obtained with SCAS with a reproduction of the model proposed in a related work Swiecicka et al. (2006) in which both synchronous and asynchronous updating were investigated.

Program graphs available in Swiecicka et al. (2006) and randomly generated graphs were considered in the experiments. Figure 1 shows the program graph *gauss18* presenting computational and communication costs. Figure 4(a) shows program graph *g18*, in which computational costs are presented and communication costs are omitted because they are equal to 1 for all edges. Figure 4(b) shows program graph *g40* with computational and communication costs equal to 4 and 1, respectively. Program graphs *random30*, *random40* and *random50* were generated with 30, 40 and 50 tasks, respectively, being that computational and communication costs were randomly generated.

Fig. 3: SCAS scheme.

The number of processors V_s used in all experiments is 2. Linear neighborhoods with radius 2 and 3 were used. The number of time steps during CA temporal evolution is equal to 50. The parameters used in GA were: size of population $T_{pop} = 200$, simple tournament with $Tour = 2$, crossover rate $P_{cross} = 100\%$, mutation rate $P_{mut} = 3\%$ and number of generations $G = 200$ except for program graph *gauss18* where $G = 1000$ as in Swiecicka et al. (2006). For each experiment, 20 runs were performed and the scheduling policy adopted is "the task with the highest dynamic level first".

Table 1 shows the results found in learning mode for SCAS and those obtained with two reproductions of the model proposed in Swiecicka et al. (2006): one with synchronous updating mode (Swie-Par) and the other with asynchronous updating mode (Swie-Seq). In Table 1,"LM" means the fitness of best rule for learning mode (out of 20 runs) and "AVG" shows the average of the best rules considering 20 runs.

Graphs	SCAS		Swie-Par		Swie-Seq	
	LM	AVG	LM	AVG	LM	AVG
g18	46,00	46,00	46,00	46,00	46,00	46,00
g40	80,00	80,81	80,00	80,76	80,00	80,89
gauss18	44,00	47,86	47,00	49,31	44,00	47,60
random30	1225,84	1267,50	1250,76	1275,81	1239,00	1247,71
random40	996,52	1024,19	1008,32	1027,58	1006,00	1020,47
random50	661,04	673,98	669,68	676,80	659,04	667,78

Tab. 1: Learning mode in CA-based scheduler models.

The first experiment was conducted with program graph *g18* and *g40*. The optimal solutions for *g18*

**Fig. 4:** Program graphs found in literature: (a) *g18*; (b) *g40*.

.

and *g40* with $V_s = 2$ are respectively 46 and 80. In Swiecicka et al. (2006) it was possible to find the optimal solutions for these program graphs with synchronous and asynchronous updating mode of cells. Table 1 shows that SCAS and reproductions of Swiecicka et al. (2006) were also able to find optimal solution for *g18* and *g40*. Furthermore, the quality of rules obtained was also examined. Figure 5 presents the average of fitness obtained in execution mode for each rule stored in RDB, for the best run (out of 20). Although finding the optimal solution, Figure 5 shows that the reproductions Swie-Par and Swie-Seq created a set of rules with a very large variation in scheduling performance. On the other hand, SCAS was able to find all rules with optimal performance.
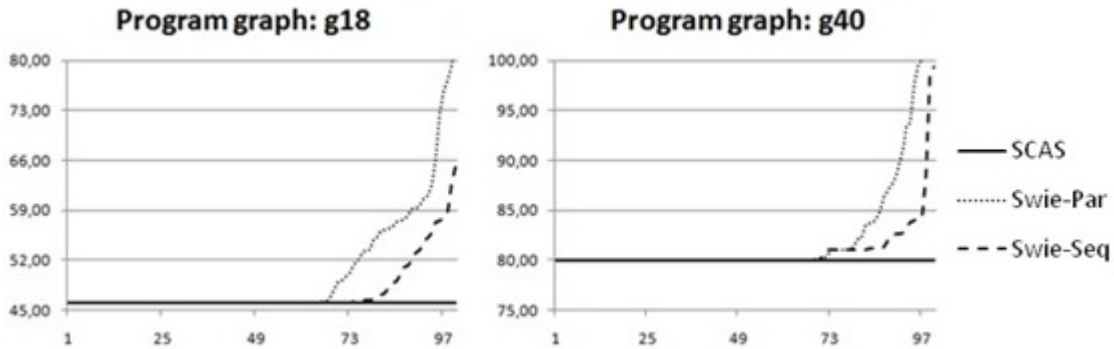
Fig. 5: Execution mode for *g18* and *g40*.

Gauss18 was used in the second experiment. Considering $V_s = 2$, the optimal solution for this program

graph is 44. Swiecicka et al. (2006) found the optimal solution for *gauss18* only using asynchronous update mode of cells. Table 1 shows that reproduction Swie-Par that uses synchronous updating mode of cells was not able to find optimal solution, unlike SCAS and reproduction Swie-Seq. The quality of rules obtained in learning mode for *gauss18* were also examined. Figure 6(a) presents the average of fitness obtained in execution mode by each rule of final population. Reproductions Swie-Par and Swie-Seq generated a final set of rules with a very large variation in performance: more than 50% of the rules were not able to find optimal scheduling for all initial configurations. SCAS on the contrary was able to find optimal scheduling for almost all rules. The worst rules obtained with SCAS, Swie-Par and Swie-Seq schedules returned average *44.1*, *79.2* and *94.0*, respectively (considering 1000 initial configurations).
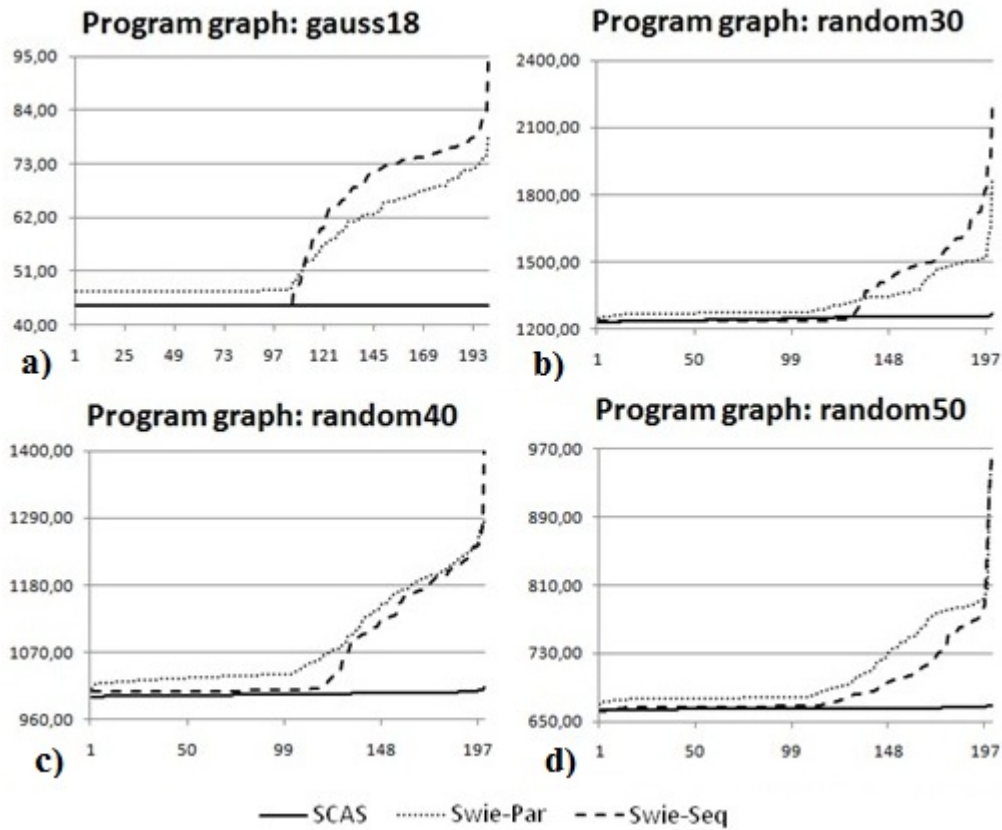
Fig. 6: Execution mode: (a)*gauss18*; (b)*random30*; (c)*random40*; (d)*random50*.

The third experiment was conducted with the randomly generated program graphs: *random30*, *random40* and *random50*. It is possible to observe in Table 1 that SCAS returned the best results for learning mode using *random30* and *random40*. For *random50*, the best result was found by Swie-Seq, although the best rule of SCAS returned a closest performance. Figure 6 shows the performance of rules in the

execution mode: figures 6(b), 6(c) and 6(d) for *random30*, *random40* and *random50*, respectively. One can see a large variation in scheduling performance to all random program graphs, except for SCAS rules. Considering *random30*, the worst rules obtained by SCAS, Swie-Par and Swie-Seq returned an average *1265.42*, *1875.97* and *2186.54*, respectively. Considering *random40*, the worst rules obtained with SCAS, Swie-Par and Swie-Seq returned an average of *1012.54*, *1286.56* and *1414.88*, respectively. Finally, the worst rules considering *random50* obtained with SCAS, Swie-Par and Swie-Seq returned an average of *668.24*, *959.70* and *966.15*, respectively.

Table 2 shows the results obtained in execution mode using SCAS, Swie-Par and Swie-Seq. It is expected that, for any initial allocation of tasks, the discovered rules being able to minimize *makespam*. Each value in Table 2 presents the performance of the best result in execution mode when each rule is used to evolve 1000 randomly generated initial configurations. All results for SCAS were equal (*g18*, *g40*) or better (*gauss18*, *random30*, *random40* and *random50*) than Swie-Par. In addition, SCAS showed better results than Swie-Seq for *random30* and *random40*, while results for *random50* were close.

Graphs	SCAS	Swie-Par	Swie-Seq
g18	46,00	46,00	46,00
g40	80,00	80,00	80,00
gauss18	44,00	47,00	44,00
random30	1226,95	1251,36	1239,00
random40	997,27	1010,40	1006,00
random50	662,90	669,48	661,18

Tab. 2: Execution mode in CA-based scheduler models.

6 Conclusions

This paper presents a new model of CA-based scheduler for Task Static Scheduling Problem (TSSP) in multiprocessors. It was called *Synchronous CA-based Scheduler* (SCAS). Previous related works pointed the promising use of cellular automata in TSSP. However, the synchronous mode was discarded because it returned worse results than the asynchronous update. Thus, the main objective of this work was to present a model that in addition to extract and reuse the knowledge it is also able to exploit the inherent parallelism in CA with good performance.

SCAS was analyzed in a comparative scheme with a previous model and it presented important advantages in relation to the synchronous updating mode on such model. In addition, obtaining a rule set where almost all elements are able to find good scheduling performance represents a major advance for the learning mode in SCAS, when compared to the previous model even considering the asynchronous mode.

Despite the good results obtained in experiments is also necessary to improve the behavior of our model on other aspects such as the increase in number of processors.

Acknowledgements

References

M. R. Garey and D. S. Johnson. *Computers and Interactability. A Guide to the Theory of NPCompleteness.* Freemann And Company, 1979.

D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, 1989.

S. Jin, G. Schiavone, and D. Turgut. A performance study of multiprocessor task scheduling algorithms. *The Journal of Supercomputing*, 2008.

Y. K. Kwok and I. Ahmad. Benchmarking and comparison of the task graph scheduling algorithms. *Journal of Parallel and Distributed Computing*, 59(3):381–422, 1999.

M. Mitchell, J. P. Crutchfield, and R. Das. Evolving cellular automata with genetic algorithms: A review of recent work. *In Proceedings of the First International Conference on Evolutionary Computation and Its Applications (EvCA'96)*, 1996.

M. L. Pinedo. *Scheduling: Theory, Algorithms, and Systems.* Springer Science, third edition, 2008.

F. Seredynski and A. Y. Zomaya. Sequential and parallel cellular automata-based scheduling algorithms. *IEEE Transactions on Parallel and Distributed Systems*, 13(10):1009–1022, 2002.

M. Sipper. *Evolution of Parallel Cellular Machines, The Cellular Programming Approach.* Springer, 1997.

A. Swiecicka, F. Seredynski, and A. Y. Zomaya. Multiprocessor scheduling and rescheduling with use of cellular automata and artificial immune system support. *IEEE Transactions on Parallel and Distributed Systems*, 17(3):253–262, 2006.

P. M. Vidica and G. M. B. Oliveira. Cellular automata-based scheduling: A new approach to improve generalization ability of evolved rules. *Brazilian Symposium on Artificial Neural Networks (SBRN'06)*, 2006.

A simple cellular multi-agent model of bacterial biofilm sustainability

Tiago Guglielmeti Correale[2†] and Pedro P.B. de Oliveira[1,2‡]

Universidade Presbiteriana Mackenzie
[1]*Faculdade de Computação e Informática &* [2]*Pós-Graduação em Engenharia Elétrica*
São Paulo, SP - Brazil

A cellular multi-agent system is used to implement a simple and abstract model of bacterial biofilm. Biofilms are social organisations of bacteria that allow them much more adaptive and functional roles than when they are found individually; in fact, contrarily to commonsense knowledge, this is the most common form of bacteria organisation in nature. A series of experiments are reported with the model, addressing the issue of biofilm sustainability, once it has been created. The model is based upon two kinds of agents, representing bacteria and food sources, the former presenting two different roles, according to their ability to sustain the biofilm production. The investigation is focused on the influence of different proportions of bacterial agents with these roles in the system. Some quantitative characterisation to the experiments is given, according to the initial world configuration, its population life span and the energy levels of the system, which allow for explanations of some qualitative observations. The latter clarify the view that biofilm sustainability depends on a balance between the apparently conflicting roles of the bacterial agents involved.

Keywords: Discrete dynamical system; multi-agent system; cellular world; DRIMA; BacDRIMA; biofilm; quorum-sensing; artificial life.

1　Introduction and motivation

Multi-agent systems have been used in a wide range of applications and as conceptual tools (Wooldridge, 2009; Jennings et al., 1998). In particular, in biology several efforts have also been made (Khan et al., 2003; Amigoni and Schiaffonati, 2007), supplementing the more traditional modelling techniques (Endy and Brent, 2001). Among those, cellular multi-agent approaches find its niche in terms of the simplicity and abstraction they naturally support, as well as with prospects to bridging the modelling efforts with the wealth of available knowledge in cellular automata theory and applications (Spicher et al., 2009; Ediger and Hoffmann, 2009).

　　The problem at issue herein is biofilm formation by bacteria, a very important subject in microbiology. Biofilms are defined as matrix-enclosed bacterial populations adherent to each other and/or to surfaces

---

or interfaces (Costerton et al., 1995). Contrarily to commonsense knowledge, the presence of bacteria in biofilms are much more common in nature than in their individualised (or, planktonic) form (Costerton, 2007). Bacteria produce and release molecules known as auto-inducers, whose concentration may be regarded as information about the density of bacteria in some region of the space. When population density of bacteria increases, the auto-inducer concentration also increases, eventually reaching a point where certain changes in the bacteria phenotype are triggered. This is the moment where bacteria can start producing a certain type of enzyme that allows the effective construction of the biofilm.

In order to explore this theme, we rely upon the current status of the BacDRIMA model, which is aimed at the possibility of addressing a number of issues in the dynamics of formation and sustainability of bacterial biofilm, from the perspective of a cellular, multi-agent system. This model is built upon the simple multi-agent system DRIMA (de Oliveira, 2010), which is totally based on local and simple rules governing the action of agents on a cellular world, much alike cellular automata. Due to space limitation, many aspects of the present conception and implementation of DRIMA are being omitted here.

The model is based upon two kinds of agents, representing bacteria and food sources. Two kinds of bacterial agents are defined, modelling two functional roles of the same kind of bacteria. The first are standard bacteria type organisms, which are the ones directly involved in biofilm formation, and are referred to herein as the *normal* bacteria, for the sake or simplicity. The second functional role defines the so-called *cheaters*, which benefit from the work of normal bacteria without directly contributing to biofilm formation.

BacDRIMA is an abstraction of all these processes. It tries to capture some essential aspects of biofilm development, without specific details related to specific bacteria, therefore aiming at an understanding of the generic dynamics of biofilm development. In order to go about it, the model was built with the following characteristics:

- Multi-agent based;

- Each agent has some kind of energy, that simulates their strength;

- The system has energy (or food) sources;

- Agents can cooperate, by jointly releasing enzymes to maximize energy production, but they can cheat on the work of others;

- Agents can have different properties; and

- The system must support the existence of cheaters(in the sense that they may exist without producing enzymes).

BacDRIMA was developed in *Mathematica*, just like the DRIMA system it was built upon (de Oliveira, 2010). The results reported are preliminary, and refer to the dynamics of cooperation versus cheating. Specifically, although one might imagine that cheaters are always deleterious for the whole system, in terms of always making biofilm development more difficult, the experiments to be reported indicate that this is not always the case. In fact, depending on certain conditions, cheaters help the system as a whole.

The remainder of the paper is organised as follows. After very briefly describing DRIMA in the next section, the basic concepts behind BacDRIMA are presented in the sequence. Then, artificial experimental results are presented, drawn from various experiments, with different initial conditions. Finally, a conclusions section discusses the results obtained, the model itself, and perspectives for the subsequent developments of the work.

2 DRIMA

DRIMA, an acronym for Dynamics of Randomly Interacting Moving Agents, is a discrete dynamical system, created around the idea of a set of reactive agents that interact locally, by changing the way they move (de Oliveira, 2010). It is composed of a regular lattice of cells, with periodic boundary conditions, a set of agents placed on the cells, and parameters that defined the dynamics of the agent's interactions. As a model and a computational system, DRIMA is a tool that may be used in a spectrum of experiments, as long as the problem at issue would rely on the local interaction among the agents resulting in their movement patterns being affected. Agents move on the lattice in a non-deterministic and local way, according to the probabilities defined by their so-called movement pattern; in particular, there is also a probability of their not moving. Each agent has an interaction radius associated with them, and they can interact with all agents within that radius. Interactions will change their movement pattern, by altering the probabilities associated with each direction of movement. The agents follow an interact-first-then-move cycle.

Although various aspects of DRIMA have similarities with cellular automata, a key difference to be noticed is that DRIMA's grid is only a lattice on top of which the agents can roam about.

Agent movement in DRIMA is in general non-deterministic and the lattice is presently either two- or one-dimensional. In the one-dimensional version, each agent can move to the left or right, or simply stay at its current position. In the two-dimensional case, which is our concern herein, each agent has 9 possibilities: East (E), Northeast (NE), North (N), Northwest (NW), West (W), Southwest (SW), South (S), Southeast (SE), and Stop (X). Each agent have a probability associated with each movement possibility. In the case of deterministic agents, only one possibility has probability one, and all others zero; their movement pattern does not change in time, and they are not affected by interactions with other agents. In the case of random agents, each direction has an associated probability (jointly total ling 1), represented as a vector that describes the agent's movement pattern.

The interactions between agents are represented by changes in their movement pattern. Each agent has a radius of influence, within which the interactions occur, including the possibility of an agent interacting with various agents at once (an n-ary interaction). At each interaction, only one agent changes their movement pattern, referred to as the reference agent.

In the case of n-ary interaction, initially the reference agent is identified, together with the neighbouring agents it will interact with.

Since the reference agent will in fact interact with the resulting vector obtained from the movement patterns of the agents in its neighbourhood, this resulting vector is first obtained and normalised (to ensure that its total movement probability remains equal to 1). Only then the actual interaction can occur.

As for the interaction itself, the idea is that the reference agent is 'attracted', so to speak, in the direction of the resulting movement vector of its neighbouring agents. This attraction is implemented in terms of a rotation of the vector representing the reference agent towards the resulting neighbouring vector, and the amount of rotation depends on the angle between the agents.

The actual interaction is governed by Shannon's entropy, associated to each interacting vector, defined as $H = -\sum_{i=1}^{N} p_i log(p_i)$ (Borda, 2011). In our case, $N = 9$ (nine possible movements), p_i is the probability associated with the direction i. The idea here is that entropy indicates the degree of randomness in the movement pattern of an agent.

In order to calculate the approximation angle the reference agent has to undergo as a result of an interaction, a function was defined that gives the angle of approximation, according to the entropies of the

agents involved in the interaction. The details are being omitted here, but the function definition followed the three general conditions below:

1. Movement limits: When an agent with maximum entropy interacts with another with minimum entropy, the agent with minimum entropy should not undergo any changes, while the other should undergo the maximum possible change.

2. Different random agents with minimum entropy should have a minimum (though not null) change in their movement pattern.

3. Agents with the same movement pattern should not change as the result of an interaction.

The 'vectorial' interaction scheme introduced above differs from the one described in (de Oliveira, 2010), and will be presented in detail elsewhere.

3   BacDRIMA and Biofilms

BacDRIMA is a biologically inspired multi-agent model, that relies on DRIMA for its basic dynamics. The reader should be aware that BacDRIMA should be regarded as an abstract model, since it neglects several details of its biological counterpart.

In BacDRIMA, two kind of agents are defined: the bacterial agents, and the energy or food source. Both of them are placed on a two-dimensional grid, presently with periodic boundary. The bacterial agents can move about, in a completely random way, while the food source is randomly placed on the grid, fixed. Each bacterial agent has an active metabolism that consumes energy at each iteration. If the internal energy goes down to 0 the agent dies, so that it must roam around the grid trying to find energy sources to fulfil its energy necessities.

Food sources do not release energy immediately; rather, they require a certain level of enzyme to be released on them, after being produced and secreted by normal bacteria. The enzymes degrade with time, i.e. after some number of iterations they will be destroyed. In order to release more energy, more enzyme must be produced. Each food source has a finite amount of energy to be released, and will eventually cease after some point.

In order to make enzymes, normal bacteria spend energy. Therefore, an energy balance has to be achieved in the system: in order to get energy, an agent must have some energy to produce enzymes, which in turn will be used to liberate energy from the food sources. Since the agents cannot control how much energy will be released by the food source, they must find a good strategy to survive, before their initial energy level becomes too low.

Both normal bacteria and cheaters can secrete another substance, the auto-inducer, that regulates enzyme production. Normal bacteria will produce enzymes only after the auto-inducer concentration becomes greater than a certain value. This is a simple model based on bacteria behaviour. In BacDRIMA, each agent measures its own auto-inducer concentration, as well as those of its neighbours. According to the total auto-inducer production in the neighbourhood of an agent, it may start the enzyme synthesis.

For the sake of energy release, a food source considers the global production of all agents in its neighbourhood, so that the actual release starts when a certain level of enzyme is locally present. But notice that this benefits all agents in its neighbourhood, not only those responsible for the enzyme production that triggered the energy liberation. This is a very important aspect of the model, because it makes it possible the definition of cheaters, i.e., agents that receive the released energy, without having contributed to the enzyme production.

Notice that the energy released by the food source depends on the enzyme production and on the number of agents in its neighbourhood. With higher enzyme levels, more energy is released, but more agents in the neighbourhood ends up sharing it. So, for some enzyme level, more neighbours entail less energy to be accumulated in the food source for each agent. Hence, a trade-off becomes apparent in the model. Since the total amount of energy that each food source can release is set at the start of an experiment, the number of agents around the food source is irrelevant. In fact, what changes is the rate of energy consumption, and not the total amount of energy of the system. So, the rate of energy consumption can be regarded as an efficiency measure of the system. Accordingly, the experiments that we run have shown various degrees of overall efficiency in the system.

Another key point of the model is the way the agents move, which is a slightly modified version from DRIMA, since a simple chemotaxis mechanism has been added. Accordingly, at each iteration, each agent measures its own energy level and, if the energy level keeping decreasing through a certain number of iterations in sequence, the agent starts searching for food in a completely random fashion (this is done by changing the agent's movement pattern to fully random). Nevertheless, if the internal energy of an agent decreases to 0, it eventually dies, being removed from the world.

Now, how does BacDRIMA's characterisation above relate to biofilm production? The way that agents interact in DRIMA entails the deterministic agents to have a huge influence in the system's dynamical stability. In fact, it is typically the case that if all deterministic agents have the same movement pattern, the system eventually converges to this movement pattern. Since all energy sources in BacDRIMA are modelled as deterministic agents (with probability 1 of staying at the same position) all other agents will tend to stop in the neighbourhood of the energy sources. And this tendency can be regarded as a metaphor to biofilm formation.

But since the chemotaxis mechanism may push away from this trend, a question arises about the stability degree of the existing biofilm, in terms of some of the variables involved. This question is the focus of the following section.

4 Experiments

Initially, a series of runs were performed (whose results are being omitted here), just to check whether the model would lead to basic coherent observations. After this successful stage, another set of experiments were run, with the following characteristics:

- Various grid sizes, with the same number of food sources (so as to test the influence of the density of energy sources in the system).
- Fixed population size but varied composition, in terms of different number of normal bacteria and cheaters (so as to test the influence of cheating in the system).
- Each combination of grid size and population composition, is run 10 times, from randomly generated initial conditions, i.e. random initial positions of food sources and bacteria.

Across all experiments, the following parameters are kept the same:
- Each bacteria initial energy: 1000 energy units.
- Each food source's initial energy: 10 times the bacterial value, i.e., 10000 energy units.
- Number of food sources: 5, randomly placed on grid at each execution.
- Enzyme production threshold (i.e., amount of auto-inducer units that must be made in order to activate enzyme production by each bacteria): 10.
- Energy cost to make each enzyme: 10 energy units.

- Amount of energy released by the food source, when enzyme production threshold is attained: 100 energy units.
- Enzyme production rate per bacteria: 1 molecule per iteration.
- Each execution has 2000 iterations.
- Auto-inducer release rate per agent: 1 molecule per iteration for normal bacteria, and 4 for cheaters.
- Waiting period of a bacteria until the chemotaxis mechanism is activated: 10 iterations.
- Energetic need of bacteria for their metabolism: 10 energy units.
- Dying-out criteria for all agents: when their energy level decrease to 0. However, while bacterial agents are removed from the grid, the food sources are not.
- Each enzyme has 20% probability to be degraded at each iteration.

Naturally, cheaters do not produce enzymes, but they produce auto-inducers. We could use the same value of auto-inducer production for both kind of agents, but it will be necessary to use a greater number of agents, making the simulation slow. Although it change the numeric value of simulations, it does not change the qualitative behaviour of the system. In order to test the effect of cheaters, experiments were made with varied percentage of cheaters in the population. All runs have 10 agents simulating bacteria, each one with a different proportion in the number of cheaters and normal bacteria (exception made for the situation with 10 cheaters, because without any bacteria producing enzymes, the system does not obtain energy).

4.1 Results

A key indicator for the successfulness of an organism is its life span. In the present case, in order to have an estimate of the life span of a population of agents, the individual life spans are accumulated, over the entire set of 10 experiments, and the average taken. This measure can be regarded as the total life span of the biofilm; hence, the higher the latter, the larger the sustainability degree of the biofilm.

By varying the proportion of cheaters in the population and the grid size, while preserving the same amount of food source, some aspects become apparent. The smaller the grid size, the larger the energy density available in the world; this is clearly the case for the 4×4 grid, as shown in Figure 1, that refers to normal bacteria alone. Notice that they live less, as the proportion of cheaters grows. Also, they can live relatively longer for small grid sizes than for larger ones. On a very large grid (100×100), their life span becomes the same as in the situation with no food sources, indicating they would be living just with the energy they started with.

Depending on the type of bacterial agent at issue (normal bacteria, cheaters, or both of them together), different kinds of observations can be made, as the grid size grows. So, for cheaters, the situation is the opposite to that of the normal bacteria, as shown in Figure 2. Cheaters cannot produce enzymes, so that, in order to survive for longer, they must find normal bacteria that secrete enzymes. Since in the experiments each agent has 1000 energy points at the beginning, but needs 10 points at each iteration to fulfil its own internal metabolism, a cheater can survive for 100 iterations, without any additional energy source (analogously, n cheaters have a total life span of $100n$). In Figure 2 this corresponds to the two lower curves (grid sizes larger than 25). For these sizes, cheaters simply do not take any advantage. However, for small grid sizes (and, consequently, larger energy densities) cheaters benefit much more than normal bacteria. But the benefits depend on the proportion of cheaters in the population. Notice in Figure 2 that the cheaters' total life span can grow until the proportion of 0.7; after this point, their total life span starts diminishing, as there are just too many cheaters for few normal bacteria, and the system collapses.
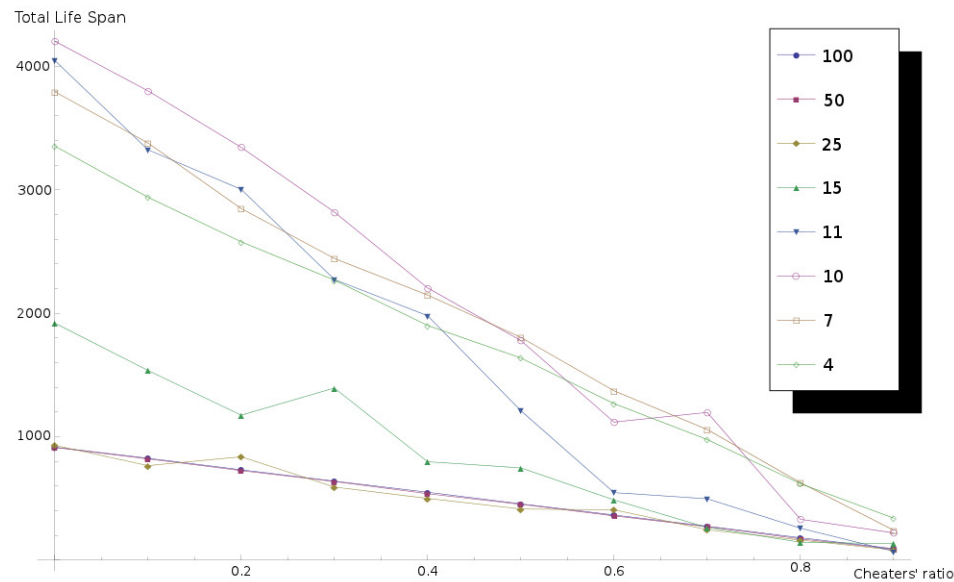
Fig. 1: Total life span for normal bacteria, with different grid sizes.
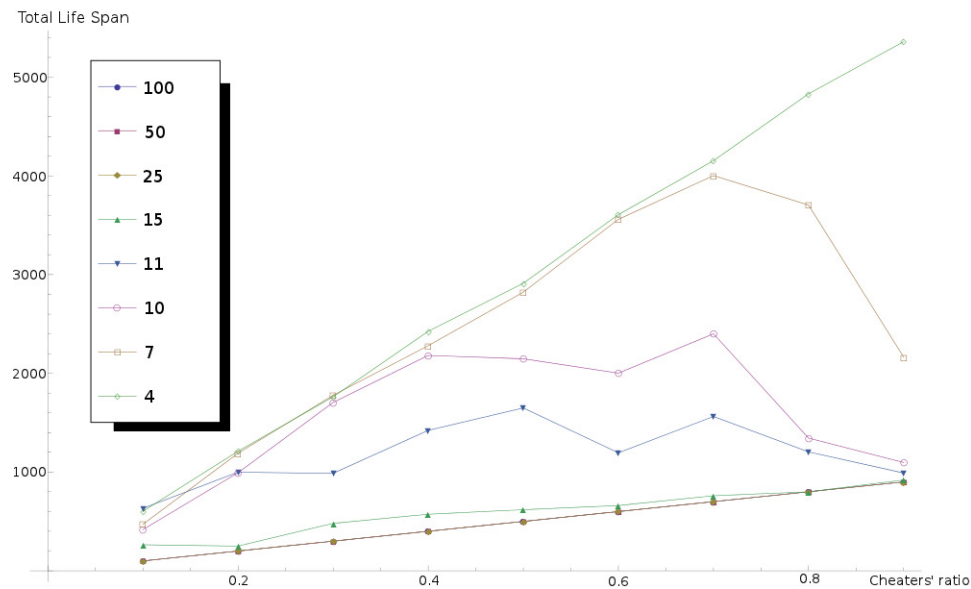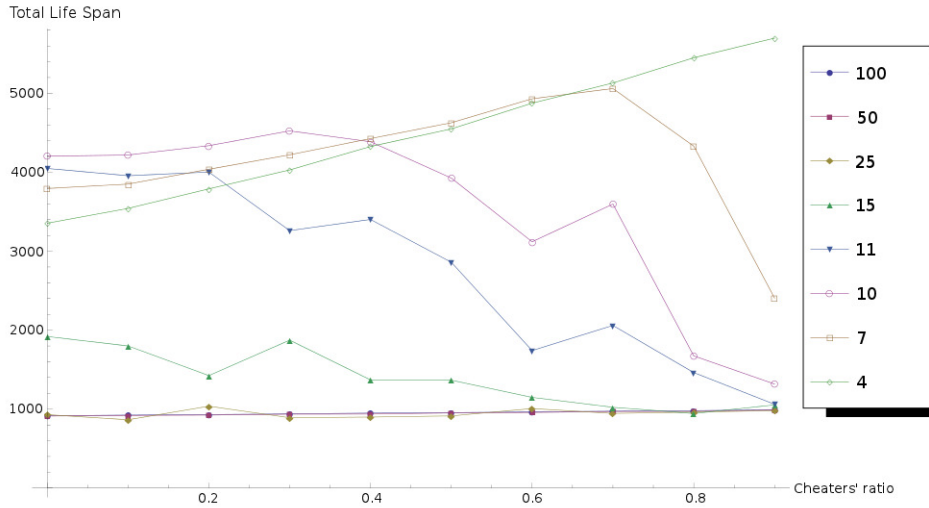

Fig. 2: Total life span for cheaters, with different grid sizes.

Figure 3 shows the situation for all bacterial agents, normal and cheaters, that is, a characterisation of the biofilm as a whole. For grid sizes smaller than 10×10, the system has always a tendency to grow their total life span, until the proportion of cheaters becomes 0.7. So, as a whole, cheaters can make the total life span grow, up to a critical point. For grid sizes larger than 25×25, the overall total life span becomes the same, so that the good influence of the cheaters to the biofilm sustainability can no longer be observed. In other words, the interesting good effect of the cheaters on the biofilm happens for high energy densities of the world.
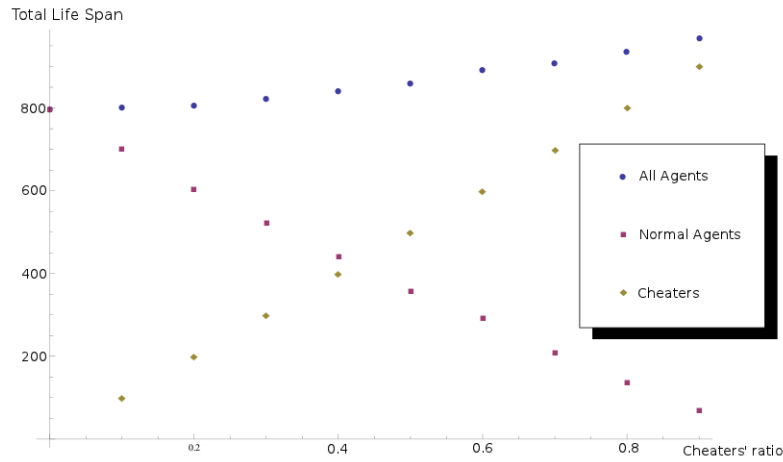

Fig. 3: Total life span for all agents, with different grid sizes.

Figure 4 refers to 10 simulations, with a 7×7 grid, without any food source. The overall system (i.e., the biofilm as a whole) has a small increase in its total life span, as the proportion of cheaters increases. The increase is a consequence of the fact that cheaters do not spend energy at enzyme production, but normal agents do, even though with unfruitful effects. But notice that the maximum life span in all three cases depicted in the figure falls below 1000 iterations, which is the maximum life span of the agents in the situation where they would rely only upon their initial energy levels, without any energy consumption (due to enzyme production).

Figure 5 depicts two plots of the total internal energy of all agents of the system on a 7×7 grid. Each time an agent obtains energy from a food source, its internal energy increases. But, at each iteration, each agent spends energy with its own metabolism, and on enzyme production. The total energy of the system is the sum of the internal energy of all agents. The peak in the solid line plot of Figure 5 corresponds to the moment at which most agents are getting the largest amount of energy from all sources.

Notice that in the solid line of Figure 5, a peak occurs for 125 iterations. And a similar peak was also observed for an ensemble of 60 executions (with the very tight standard deviation of less than 10), with varying proportions of cheaters, up to 60%. For progressively larger proportions, the peak gets displaced more and more to the left-hand side, eventually reaching the situation depicted in the dashed

**Fig. 4:** Total life span for a system without food, grid size 7×7, with different agent types.

line, that corresponds to 90% of cheaters. The intensity of the peak becomes smaller as the concentration of cheaters increases; so, in the case displayed in the dashed line, the maximum value is the very first point, indicates that the energy of the system only gets smaller at each iteration. This dynamics is a consequence of the energy balance of the system. So, with very few (or no) cheaters, normal agents can get more energy from the food sources, therefore living longer. Consequently, they are able to produce more enzymes which, in turn, entail they can get more energy, and the cycle restarts. When the proportion of cheaters becomes higher than 60%, the normal agents tend to live less, thus producing less enzymes, therefore leading the peaks to happen earlier and, since the system has produced less energy, the peaks are progressively smaller.

5   Concluding remarks

A cellular multi-agent model is used to address the problem of bacterial biofilm sustainability. In spite of its biological motivation, this work can also be clearly regarded as aligned with artificial life type efforts (Langton, 1997). The study is a direct application of the current status of the BacDRIMA model, which is aimed at the possibility of addressing a number of issues in the dynamics of formation and sustainability of bacterial biofilm, from the perspective of a cellular, multi-agent system. Key in the present study is the premise that the biofilm formation is a direct consequence of the inherent dynamics of DRIMA, on which the model is implemented. Aspects of the present conception and implementation of DRIMA are being omitted here.

Some quantitative characterisation to the experiments was given, according to the initial world configuration, its population life span and the energy levels of the system, which allowed for explanations of some qualitative observations. The latter clarified the view that biofilm sustainability depends on a balance between the apparently conflicting roles of normal bacterial agents and cheaters.

Indeed, on a certain proportion of those agents in the system there is a degree of cooperation among them, so as to support the biofilm sustainability as a whole. Depending on the grid size and the proportions
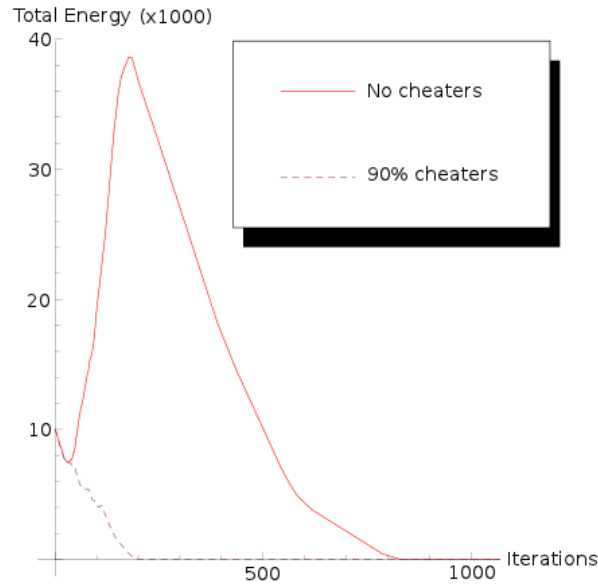
Fig. 5: Total internal energy of all agents in the system, on a 7×7 grid. Solid line: energy level at each iteration, with no cheaters. Dashed line: energy level for a single execution, with 90% of cheaters.

of bacterial agents, cheaters are not bad for the biofilm sustainability, as one might preconceive. This is a direct consequence of the fact that auto-inducers are important for the regulation of enzyme production of the normal bacterial agents. But if the proportion of cheaters becomes too high, the system collapses. This is in agreement with results in (Sandoz et al., 2007), (Travisano and Velicer, 2004) and (Allison, 2005). Consequently, in spite of its simplicity and high level of abstraction, BacDRIMA displays consistent outcomes.

Viweing the present results under the light of (West et al., 2006), the normal bacteria are "actors" (using that paper's notion) that must secrete enzymes to get energy, and they benefit from their own enzyme release; but cheaters (therein referred to as "recipients") benefit too. So, both types of organisms benefit from enzyme production, in clearly mutual benefit. The problem is that, as the proportion of cheaters grows, the same amount of energy must be shared by the entire population, thus entailing that the benefit of enzyme production becomes lower. At some point, normal bacteria become unable to get enough energy to survive, and eventually die out; but since cheaters cannot produce enzymes, they cannot get energy by they own, and eventually die out as well. At this point, the phenomenon dubbed in (West et al., 2006) as "tragedy of the commons" comes about: if all bacteria would cooperate and produce enzymes, all the population would benefit, but, with too many individuals not cooperating, the system breaks, and everyone dies. So, depending on the proportion of cheaters, three types of system can be observed: with low proportion, mutual benefit; with the selfish behaviour of cheaters, the benefits of enzyme production for normal bacteria declines, down to the point where the latter's behaviour becomes altruistic (as they spent more energy to produce enzymes than to get energy from food); finally, after the death of all normal bacteria, the tragedy of the commons becomes apparent, leading to the extinction of the population, in a

clearly spiteful behaviour.

Since BacDRIMA is yet an ongoing development, forthcoming improvements in the work include:

- Reproduction: The main focus of the current effort to expand the BacDRIMA model is the introduction of an asexual reproduction scheme for the agents, based upon their genome, which encodes the agent's enzyme production and auto-inducer production.
- Diffusion of chemical elements: Since in the present model the chemical elements (enzymes and auto-inducers) do not diffuse on the grid, the addition of some kind of diffusion would make the overall behaviour more natural.
- Addition of energy cost to auto-inducer production: This is meant to allow the study of the relation between the metabolism energy cost, enzyme production and auto-inducer production.

Acknowledgements

References

S. Allison. Cheaters, diffusion and nutrients constrain decomposition by microbial enzymes in spatially structured environments. *Ecology Letters*, 8(6):626–635, 2005.

F. Amigoni and V. Schiaffonati. Multi-agent-based simulation in biology. *Model-Based Reasoning in Science, Technology, and Medicine*, pages 179–191, 2007.

M. Borda. *Fundamentals in Information Theory and Coding*. Springer, 2011.

J. Costerton. *The Biofilm Primer*. Springer Verlag, 2007.

J. Costerton, Z. Lewandowski, D. Caldwell, D. Korber, and H. Lappin-Scott. Microbial biofilms. *Annual Reviews in Microbiology*, 49(1):711–745, 1995.

P. P. B. de Oliveira. DRIMA: A Minimal System for Probing the Dynamics of Change in a Reactive Multi-agent Setting. *The Mathematica Journal*, 12(1):1–18, 2010.

P. Ediger and R. Hoffmann. CA models for target searching agents. *Electr. Notes Theor. Comput. Sci.*, 252:41–54, 2009.

D. Endy and R. Brent. Modelling cellular behaviour. *Nature*, 409(6818):391–396, 2001.

N. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-agent Systems*, 1(1):7–38, 1998.

S. Khan, R. Makkena, F. Mc Geary, K. Decker, W. Gillis, and C. Schmidt. A multi-agent system for the quantitative simulation of biological networks. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 385–392. ACM, 2003.

C. Langton. *Artificial life: An Overview*. Complex Adaptive Systems. MIT Press, 1997.

K. Sandoz, S. Mitzimberg, and M. Schuster. Social cheating in pseudomonas aeruginosa quorum sensing. *Proceedings of the National Academy of Sciences*, 104(40):15876, 2007.

A. Spicher, N. Fatès, and O. Simonin. From reactive multi-agents models to cellular automata - illustration on a diffusion-limited aggregation model. In J. Filipe, A. L. N. Fred, and B. Sharp, editors, *ICAART*, pages 422–429. INSTICC Press, 2009.

M. Travisano and G. Velicer. Strategies of microbial cheater control. *Trends in Microbiology*, 12(2): 72–78, 2004.

S. West, A. Griffin, A. Gardner, and S. Diggle. Social evolution theory for microorganisms. *Nature Reviews Microbiology*, 4(8):597–607, 2006.

S. West, A. Griffin, and A. Gardner. Social semantics: Altruism, cooperation, mutualism, strong reciprocity and group selection. *Journal of Evolutionary Biology*, 20(2):415–432, 2007.

M. Wooldridge. *An Introduction to Multi-Agent Systems*. John Wiley & Sons, 2009.

A simple block representation of reversible cellular automata with time-symmetry

Pablo Arrighi[1][†] and Vincent Nesme[2][‡]

[1] *Université de Grenoble, LIG, 220 rue de la chimie, 38400 Saint-Martin-d'Hères, France*
and École Normale Supérieure de Lyon, LIP, 46 Allée d'Italie, 69364 Lyon, France
[2] *QMIO, Freie Universität Berlin, Arnimallee 14, 14195 Berlin, Germany*

Reversible Cellular Automata (RCA) are a physics-like model of computation consisting of an array of identical cells, evolving in discrete time steps by iterating a global evolution G. Further, G is required to be shift-invariant (it acts the same everywhere), causal (information cannot be transmitted faster than some fixed number of cells per time step), and reversible (it has an inverse which verifies the same requirements). An important, though only recently studied special case is that of Time-symmetric Cellular Automata (TSCA), for which G and its inverse are related via a local operation. In this note we revisit the question of the Block representation of RCA, i.e. we provide a very simple proof of the existence of a reversible circuit description implementing G. This operational, bottom-up description of G turns out to be time-symmetric, suggesting interesting connections with TSCA. Indeed we prove, using a similar technique, that a wide class of them admit an Exact block representation (EBR), i.e. one which does not increase the state space.

Keywords: Reversible Cellular Automata, Time-symmetric Cellular Automata

Introduction

RCA, Block representation. In [Kar96], Kari showed that any one-dimensional or two-dimensional reversible cellular automaton (RCA) can be expressed as a composition of finite reversible gates (or 'block permutations') and partial shifts. In two dimensions the proof is quite involved, the representation requires three layers of blocks, and it has been proved that this cannot be brought down to a two-layered block representation [Kar99]; The problem is still open in higher dimensions.

However we may not need an exact representation, and be willing to encode our original cells into some larger ones (or equivalently to interleave some ancillary cells), as proposed in [DL01]. Then the construction of [Kar99] shows that even n-dimensional RCA admit a two-layered block representation. In some sense what we are doing then is simulating the original RCA in a way which preserves the spatial layout of cells, with another, simpler RCA that we know admits a two-layered block representation. In

---

this sense the intrinsically universal RCA [DL95] also accomplishes this task.

Our Section 1 revisits this issue in a minimalistic manner: In our construction each block can be interpreted a reversible version of the local update rule of the CA, moreover its size turns out to be exactly that of the Block Neighborhood introduced in [AN10].

TSCA, EBRs. Recently another line of investigation has emerged which refines the now well-studied concept of RCA to admit a further requirement: That of time symmetry. In simple terms, a CA G is time-symmetric if G is its own inverse up to a simple recoding H of the cells. More formally, $G^{-1} = HGH$ with H a self-inverse CA. Credit must be given to [MG10] for emphasizing time-symmetry as a property of CA, which has barely been studied for its own sake thus far. It is clear nevertheless that many instances of time-symmetric CA (TSCA) can be encountered in the literature, as discussed in [MG10] (for instance the Margolus lattice gas model). In the above-discussed non-exact Block representation of RCA [Kar99] just like in ours, the author first encodes a RCA F into a TSCA G_F, and then provides an EBR of G_F. As a consequence, one may wonder whether these issues, block representations of RCA and TSCA are only accidentally related, or whether exhibiting a reversible local implementation mechanism for G amounts to unravelling the time-symmetry of G.

Our Section 2 begins to explore this issue by showing the existence of an EBR for squares of locally time-symmetric CA.

1   A simple block representation

In the classical picture a CA G is usually defined by a local update rule δ, namely a function from $\Sigma^{\mathcal{N}}$ to Σ, giving the new state of a cell as a function of the old state of its neighbours; It can be thought as a 'local mechanism' for implementing G. In other words, δ can be viewed as a local gate, and G a circuit made by infinitely repeating δ across space as in Fig. 1.
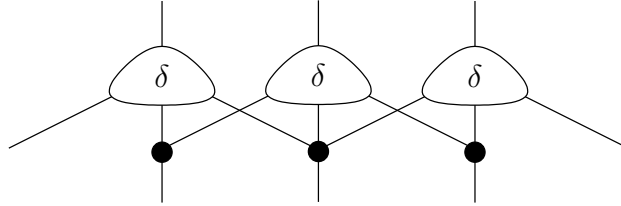
Fig. 1: The trivial circuit representation of a classical CA from its local update rule.

Using a local update rule to define RCA is of course possible, but for a circuit representation of G one may wish to use a local mechanism that is itself reversible — for instance in the context of quantum mechanical devices or due to Landauer's principle. And indeed it is the case that every RCA G admits a reversible circuit implementation. Proving the existence of such reversible circuits is the business of the aforementioned block representation theorems for RCA. It could be regretted, however, that in these theorems the reversible local gates (a.k.a blocks) constitutive of the reversible circuits (a.k.a block representations) end up looking quite different from δ. I.e. they are hard to interpret as reversible versions of the local update rule.

The following proof of the block representation theorem for RCA is hopefully simpler to understand. It starts off by defining a reversible update operator K_0, which can be interpreted as a reversible version

of the local update rule δ. We will define K_0 globally, in a way that does not make it obvious that it is actually a block permutation — but we will then proceed to show that it is the case. Notice that it is impossible to implement CA of non-trivial Welch index[i] without shifts or auxiliary space: In our case, we use auxiliary space, which results in the collateral damage of implementing, in parallel to G, its inverse on the auxiliary strip.

Repeatedly we will define a bijection f from a set of words written on some fixed set of cells X, and then wonder whether f could be defined on a smaller subset. We will say that f is localized upon $Y \subseteq X$ if we can write $f = f_Y \times \mathrm{id}_{Y \setminus X}$, i.e. if $Y \setminus X$ is superfluous in the definition of f. For instance, a bijection of $\Sigma^{\mathbb{Z}}$ that applies a permutation of the alphabet on cell 0 and leaves the other cells untouched is localized upon $Y \subseteq \mathbb{Z}$ if Y contains 0; The identity is localized on the empty set.

From the definition, it is obvious that if f is localized upon Y and $Y \subseteq Z \subseteq X$, then f is also localized upon Z. Slightly less trivial is the property that, whenever f is localized upon Y and Z, then it is also localized upon their intersection $Y \cap Z$. From there follows the existence of the smallest Y upon which f is localized, which is called the *localization* of f, and denoted $\mathrm{Loc}(f)$. So, back to our elementary example where f is a permutation π of Σ applied solely on cell 0, $\mathrm{Loc}(f) = \begin{cases} \emptyset & \text{if } \pi = \mathrm{id} \\ \{0\} & \text{otherwise} \end{cases}$.

In general, K_0 is not localized upon the neighborhood of G. We will show however that its localization is $\mathcal{BN}$, the Block neighborhood defined in [AN10] whose definition we will recall. Hence it can thus be viewed as a block permutation of size $|\mathcal{BN}|$. The last step of the proof is just to show that G a circuit made by infinitely repeating K across space.

Reversible updates K_i...

In the classical picture, the local update rule δ looks at a neighborhood $\cdots c_{-1} c_0 c_1 \cdots$ and computes $G(c)_0$, but it leaves all the other cells uncomputed. Can we, in a similar fashion, define a reversible update K_0 which focuses on computing $G(c)_0$? Moreover can we, in an again a similar fashion, define it solely in terms of G? A naive, operational approach would be to: 1. Apply G. 2. Swap $G(c)_0$ out of the system. 3. Apply G^{-1}. This will turn out to work. Technically, we will extend the alphabet to Σ^2. For i running over all cells, we denote by S_i the swap acting only on position i according to $\begin{pmatrix} \Sigma^2 & \to & \Sigma^2 \\ (a,b) & \mapsto & (b,a) \end{pmatrix}$.

Definition 1 (reversible update) *The reversible update K_i is the function from $\mathcal{C}_{\Sigma^2} \simeq \mathcal{C}_\Sigma^2$ to itself given by the following composition*

$$K_i = (G^{-1} \times \mathrm{id}) S_i (G \times \mathrm{id})$$

where $\mathcal{C}_\Sigma$ denotes the space of configurations of cells having alphabet Σ.

We can right now formulate the important remark that the K_i-s commute. We will later prove with Proposition 1 that each K_i, despite being defined globally, is actually a local permutation, acting in some neighborhood of cell i; Let us admit this fact within this paragraph. With these informations in mind, it makes sense to define the infinite product $\prod_i K_i$. Indeed, for any given cell, the number of K_i-s acting on this cell is finite; Therefore the composition of all the K_i-s can be written as a circuit of finite depth and

---

[i] For a definition, cf. section 3 of [Kar96]

is thus perfectly well-defined. Moreover, it is equal to $(G^{-1} \times \mathrm{id})S(G \times \mathrm{id})$, where $S = \prod_i S_i$. Therefore we have $S \prod_i K_i = G \times G^{-1}$.

Let us take a closer at K_0. Start with a configuration $\ldots (c_i, d_i) \ldots$. Applying $G \times \mathrm{id}$ takes it to $\ldots (G(c)_i, d_i) \ldots$. Then S_0 turns it into

$$\ldots (G(c)_{-2}, d_{-2}), (G(c)_{-1}, d_{-1}), (d_0, G(c)_0), (G(c)_1, d_1), (G(c)_2, d_2) \ldots$$

So K_0 leaves the second component unchanged, except in position 0. In fact, the rest of the second component could be left out in the definition of K_0, since it plays no role. Specifically, one can write K_0 as a product of the identity on these cells and of some bijection of $\mathcal{C}_\Sigma \times \Sigma$. The left component, after applying K_0, finds itself in the state $G^{-1}(\ldots G(c)_{-2} G(c)_{-1} d_0 G(c)_1 G(c)_2 \cdots)$. Of course, outside of some neighborhood of 0, this is the identity; But that triviality alone is not enough to conclude that K_0 is localized upon a finite number of cells. We are going to check that it is indeed the case, and moreover that its localization is a rather remarkable set.

. . . are localized within the Block Neighborhood $\mathcal{BN}$ *. . .*
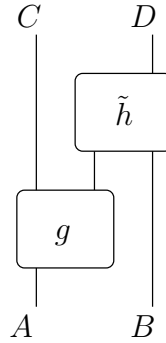
Fig. 2: Semilocalizability.

In [AN10], the authors introduced the block neighborhood $\mathcal{BN}$ of a RCA, using the concept of semilocalizability that appeared in [ESW02] in the context of quantum information theory. Given a bijection $F : X \to Y$ and a decomposition of X and Y in respectively $A \times B$ and $C \times D$, F is said to be semilocalizable (with respect to this decomposition) when it can be written in the form of Figure 2, where g and $\tilde{h}$ are themselves bijections. The quantum neighborhood of a RCA F is then the smallest subset $\mathcal{BN}$ such that, as a function from $\Sigma^{\mathcal{BN}} \times \Sigma^{\overline{\mathcal{BN}}}$ to $\Sigma^{\{0\}} \times \Sigma^{\overline{\{0\}}}$, F is semilocalizable — see Figure 3 for an illustration.

The definition of the block neighborhood was motivated by the fact that it is both the (quantum) neighborhood of the quantum CA obtained by linearization from a RCA, and obviously related to the decomposition of a QCA into a product of local permutations, a link that we make more precise in this article. More details on $\mathcal{BN}$ are to be found in [AN10], where it is the object of definition 1.9, and where explicit bounds on $\mathcal{BN}$ are given in function of the neighborhoods of G and of its inverse. We will not need these bounds, except for the fact that they do prove that $\mathcal{BN}$ is finite:
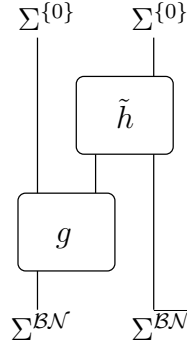
Fig. 3: The block neighborhood.

- $\mathcal{BN}$ is included in $(\mathcal{N} - \mathcal{N} + \widetilde{\mathcal{N}}) \cap (\widetilde{\mathcal{N}} - \widetilde{\mathcal{N}} + \mathcal{N})$, with $\widetilde{\mathcal{N}}$ the transpose of the inverse neighborhood $\mathcal{N}^{-1}$. There are examples saturating this bound;

- $\mathcal{BN}(G^k)/k$ tends towards $\mathcal{N}(G^k) \cup \widetilde{\mathcal{N}}(G^k)$ in the limit where k goes to infinity, with $\mathcal{BN}(G^k)$ the Block Neighborhood of G^k etc.

In the definition of K_0, cells are divided into two subcells, so that these subcells are naturally indexed by $\{0,1\} \times \mathbb{Z}$. We now prove that the localization of K_0 is essentially the block neighborhood $\mathcal{BN}$; As $\mathcal{BN}$ is also the *quantum* neighborhood, i.e. the neighborhood when inputs are not just words but can be linear combination on words (cf. [AN10]), this gives a nice way to characterize the quantum dynamics in a purely classical setting.

Proposition 1 *Consider a RCA G, and let K_0 be its reversible update. Then* $\mathrm{Loc}(K_0) = \{0\} \times \mathcal{BN} \cup \{(1,0)\}$.

Proof: $[\subseteq]$. Consider a $\tilde{h}g$-decomposition of G in the manner of Figure 3. Then g is localized upon $\mathcal{BN}$, $\tilde{h}$ outside of cell 0, and

$$
\begin{aligned}
K_0 &= (G^{-1} \times \mathrm{id})S_0(G \times \mathrm{id}) \\
&= ((\tilde{h}g)^{-1} \times \mathrm{id})S_0((\tilde{h}g) \times \mathrm{id}) \\
&= (g^{-1} \times \mathrm{id})(\tilde{h}^{-1} \times \mathrm{id})S_0(\tilde{h} \times \mathrm{id})(g \times \mathrm{id}) \\
K_0 &= (g^{-1} \times \mathrm{id})S_0(g \times \mathrm{id})
\end{aligned}
$$

where the last line follows from the fact that $\mathrm{Loc}(\tilde{h})$ does not contain $\{0\}$, whereas S_0 is localized upon cell 0. From this last line we can read $\mathrm{Loc}(K_0) \subseteq \{0\} \times \mathcal{BN} \cup \{(1,0)\}$.

$[\supseteq]$. *Note that this second inclusion is no needed for the proof of the Block representation; It is provided here just for completeness.* As we have already mentioned, $\mathrm{Loc}(K_0)$ is of the form $\mathrm{Loc}(K_0)_0 \cup \{(1,0)\}$. So $\mathrm{Loc} \prod_{n \neq 0} K_n$ does not contain $(1,0)$. But $K_0 \prod_{n \neq 0} K_n = (G^{-1} \times \mathrm{id})S(G \times \mathrm{id})$. For $a \in \Sigma$, let X_a be the subset of words on $\mathrm{Loc}(K_0)$ that are equal to a on $(1,0)$. The image of X_a by $S_0(G \times \mathrm{id})$ is of the form $Y_a \times \Sigma$, where Y_a is the set of words on $\mathrm{Loc}(K_0)_0 \cup \{(0,0)\}$ that are equal to a in $(0,0)$, and Σ is

localized on $(1,0)$. Therefore the image of X_a by K_0 is also of the form $Z_a \times \Sigma$ for some subset Z_a of the words on $\text{Loc}(K_0)_0$.

Furthermore, we know that there exists a bijection finishing the job after the isolation of $G(c)_0$ by K_0, namely $\prod_{n \neq 0} K_n$. We must thus have a semilocalization of G with respect to $\text{Loc}(K_0)_0$: In figure 3, K_0 plays the role of g, $\mathcal{BN}$ is $\text{Loc}(K_0)_0$, and $\tilde{h}$ is $\prod_{n \neq 0} K_n$. Since $\mathcal{BN}$ is the smallest set fulfilling this property, it must then be included in $\text{Loc}(K_0)_0$.   □

...and thus implement G.

Combining the above results we obtain the following:

Corollary 1 ($G \times G^{-1} = S(\prod K)$) *Consider a RCA G, and let K be its reversible update. Consider the function $G \times G^{-1}$ from $\mathcal{C}_\Sigma^2$ to $\mathcal{C}_\Sigma^2$. We have that*

$$G \times G^{-1} = S \prod_i K_i \quad with \quad \text{Loc}(K_0) = \{0\} \times \mathcal{BN} \cup \{(1,0)\}\,.$$

Hence we have here a proof that all RCA admit a block representation, the third of its genre [Kar96, DL01], but hopefully also the most straightforward, as it simply takes the form a product of reversible updates. There is one bad and one good news about this proof. The bad news is that it provides only a non-exact Block representation of RCA, leaving it open whether $n > 2$-dimensional RCA admit an EBR or not. The good news is that it provides an EBR for those TSCA which are of the form $G \times G^{-1}$. This suggests that we should look at the relation between EBRs and time-symmetry of CA.

2 EBRs and time-symmetry

The core of the argument that we developed in the previous section for the existence of an EBR for $G \times G^{-1}$ could be restated as follows: Say F and H are RCA such that H admits an EBR, then so does FHF^{-1}! Indeed, if $H = \prod_i B_i$, then $FHF^{-1} = \prod_i FB_iF^{-1}$. Moreover following Proposition 1.[⊆], the blocks FB_iF^{-1} are localized, at most, on the localization of B_i extended by $\mathcal{BN}(F)$ the block neighborhood of F; Hence each of them is finitely localized, i.e. is itself a block permutation.

In Section 1 we applied this argument with $F = G^{-1} \times \text{id}$ and $H = S$, which admits a trivial block representation $S = \prod_{n \in \mathbb{Z}} S_n$. This gave an EBR of $(G^{-1} \times \text{id})S(G \times \text{id})$, which is only a swap away from $G \times G^{-1}$. In fewer words, $G \times G^{-1}$ admits an EBR because the set of RCA having this property

- contains the permutations of Σ, and

- is a normal subgroup of the group of RCA.

Having generalized this procedure, let us now have a look at what it tells us in the context of TSCA.

Definition 2 (Locally Time-Symmetric CA) *A RCA G is a* locally time-symmetric CA *(LTSCA) if there exists an involution h of Σ such that $G^{-1} = HGH$, with $H = \prod_i h$.*

Our definition of LTSCA is identical to that of TSCA given in [MG10] except for one extra condition: We further demand that the RCA H be of radius zero. On this question of the locality of H, let us quote the authors of this first paper introducing TSCA [MG10]: "*Requiring H to be a CA is somewhat arbitrary, [. . .] the reason for this restriction is that we expect reversibility (including the particular case of time-symmetry) to be a local property.*". Moreover, whilst the theoretical results they prove are valid for H an involution RCA of arbitrary radius, it also true that in all of the examples provided, H is of radius zero. In fact, one may wonder whether there LTSCA and TSCA are not equivalent up to a simple encoding. Anyhow, if H has radius zero, then in particular it admits an EBR, and so does $GHG^{-1}H = G^2$. Therefore, the squares of LTSCA have EBRs:

Corollary 2 (EBR of LTSCA^2) *Let G be an LTSCA with respect to an involution h . We have $G^2 = H \prod_i L_i$, where $L_i = G^{-1}h_i G$, furthermore* $\mathrm{Loc}(B_0) \subseteq \mathcal{BN}$.

Some remarks are in order:

- h_0 plays the role that S_0 had in section 1. Likewise, in the standard examples of TSCA [MG10], H can be interpreted as a swap. This is certainly the case in particular for the standard time-symmetrizations $G \times G^{-1}$ of any RCA G, as in Prop. 5.3. of [MG10].

- This time the block representation is an exact one, hence it is remarkable that LTSCA have this property given the difficulty of finding the EBRs of $n > 2$-dimensional RCA. Nevertheless, the representation applies to G^2 and not G itself. Simply proving that any involutive RCA admits an EBR is probably difficult, as it gets dangerously close to solving the aforementioned open problem.

Conclusion

Generalizations. As in [AN10], the block representation defined in Section 1, and the proof that it is of minimal size, rely only on notions on neighborhood, while others characteristics of CA, such as finiteness of the alphabet and translation invariance, are simply irrelevant. Moreover, whilst the arguments we have provided in this paper are purely classical, they have their counterparts in the field of quantum CA [SW04], some of which were of direct inspirations to this paper [ANW]. Part of our motivation was to make these techniques available to classical CS.

Questions, answers and more questions. Why is time-symmetry such a key step Block representations of RCA? In this paper gave a simple proof of the block representation of RCA, which partly explains this role. Could it be that TSCA admit an EBR? In this paper we gave a simple proof of the EBR of squares of LTSCA. These are all but partial answers, suggesting that many questions remain on the topic of understanding differences in structure between RCA and TSCA, TSCA and LTSCA. There might lie a path towards EBRs of RCA in arbitrary dimensions.

Acknowledgements

References

[AN10]   Pablo Arrighi and Vincent Nesme. The Block Neighborhood. In TUCS, editor, *Proceedings of JAC 2010*, pages 43–53, Turku, Finlande, December 2010.

[ANW]   Pablo Arrighi, Vincent Nesme, and Reinhard F. Werner. Unitarity plus causality implies localizability. To appear in *Journal of Computer and System Sciences*. `arXiv:0711.3975v3`.

[DL95]   Jérôme Durand-Lose. Reversible cellular automaton able to simulate any other reversible one using partitioning automata. In *Proceedings of the Second Latin American Symposium on Theoretical Informatics*, LATIN '95, pages 230–244, London, UK, 1995. Springer-Verlag.

[DL01]   Jérôme Durand-Lose. Representing reversible cellular automata with reversible block cellular automata. In Robert Cori, Jacques Mazoyer, Michel Morvan, and Rémy Mosseri, editors, *Discrete Models: Combinatorics, Computation, and Geometry, DM-CCG '01*, volume AA of *Discrete Mathematics and Theoretical Computer Science Proceedings*, pages 145–154, 2001.

[ESW02] T. Eggeling, Dirk Schlingemann, and Reinhard F. Werner. Semilocal operations are semilocalizable. *Europhysics Letters*, 57(6):782–788, 2002.

[Kar96] Jarkko Kari. Representation of reversible cellular automata with block permutations. *Mathematical Systems Theory*, 29(1):47–61, 1996.

[Kar99] Jarkko Kari. On the circuit depth of structurally reversible cellular automata. *Fundam. Inf.*, 38(1-2):93–107, 1999.

[MG10]   Andrés Moreira and Anahí Gajardo. Time-symmetric Cellular Automata. In TUCS, editor, *Proceedings of JAC 2010*, pages 180–190, Turku, Finlande, December 2010.

[SW04]   Benjamin Schumacher and Reinhard F. Werner. Reversible quantum cellular automata. `arXiv:quant-ph/0405174`, May 2004.