



HAL
open science

Associating parallel automata network dynamics and strictly one-way cellular automata

Pacôme Perrotin

► **To cite this version:**

Pacôme Perrotin. Associating parallel automata network dynamics and strictly one-way cellular automata. *Natural Computing*, 2023, 22 (3), pp.453-461. 10.1007/s11047-023-09948-z . hal-04440102

HAL Id: hal-04440102

<https://hal.science/hal-04440102v1>

Submitted on 5 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Associating parallel automata network dynamics and strictly one-way cellular automata

Pacôme Perrotin
Aix-Marseille Univ., Univ. de Toulon, CNRS,
LIS, UMR 7020, Marseille, France

Abstract

We prove that the limit dynamics of any finite automata network under the parallel update schedule correspond exactly to the fixed points of a so-called strictly one-way cellular automaton. This correspondence is proven to be exact, as any strictly one-way cellular automaton can be transformed into a corresponding automata network, the limit dynamics of the latter corresponding exactly to the fixed points of the former. This transformation is operated using output functions, which have been developed in the author's previous works.

1 Introduction

Automata networks (ANs) are used to model gene regulatory networks [10, 19, 12, 4, 5]. In these applications the dynamics of automata networks are used to understand how the biological systems might evolve. As such, there is motivation in improving our computation and characterisation of automata network dynamics. Cellular automata (CA) are a well studied family of dynamical systems with broad applications in the study of physical phenomena [3] and in many other fields [1, 7, 13]. If automata networks and cellular automata are similar in definition, automata networks operate on a finite graph using a variety of local functions, while cellular automata are defined by their regular (and usually infinite) topology as well as their uniform local rule. As such, automata networks are described by some as non-uniform cellular automata [2].

One-way cellular automata, a subset of cellular automata which only allows local rules defined on one side of the updated cell, have been throughoutly studied [18, 9, 11]. Strictly one-way cellular automata are, as far as the authors are aware, a sub-class of one-way cellular automata that have not been studied yet. Being strictly one-way is a very restrictive property, as we can characterise the fixed points of such cellular automata thanks to automata networks: Corollary 1 implies that a strictly one-way cellular automata has a finite set of fixed points which are all periodic.

Our work is motivated in improving the formal description of the limit dynamics of automata networks. A complete characterisation of the limit

dynamics has only been provided to simply defined families of network which still have complex behavior, such as isolated cycles and disjunctive intersection of pairs of cycles [14]. The general approach of the field is to look for families or properties which allow for limit dynamics that we might be able to characterise [6, 8].

To help in that effort, we developed the formalism of modules [15] which are automata networks with inputs. By focusing on modules with acyclic interaction digraph, we proved that the limit dynamics was determined (up to the renaming of configurations) by so called output functions [16]. By focusing on output function sets rather than automata networks, we allow for the automatic removal of many specificities of the networks which do not determine the limit dynamics.

This paper centers itself on the transformation of any so-called sufficient output function into the local rule of a strictly one-way cellular automata, and vice versa. This transformation also shows that the limit dynamics of the automata networks and the fixed points of the cellular automata are direct translations of one another. This result proposes an interesting correspondence between cellular automata and automata networks, as any AN has an associated strictly one-way CA, and vice-versa. As such, this result states that computing the limit dynamics of an automata network under the parallel update schedule and computing the fixed points of a strictly one-way cellular automaton is the same exact computational problem. In the case of strictly one-way CA, this means that enumerating their fixed points can be done with the use of a finite object.

Additional results are provided that utilise this transformation to apply an existing result about automata networks to the fixed points of strictly one-way cellular automata, as well as further transformations into subshifts of finite type as well as aperiodic necklaces with forbidden patterns.

2 Definitions

2.1 Preliminaries

A *digraph* is a pair of sets $G = (V, E)$ where V is a finite set of *nodes* and E is a set of *edges*, defined as a subset of $V \times V$. We say that two digraphs $G = (V, E), G' = (V', E')$ are *isomorphic* if there exists a bijection $g : V \rightarrow V'$ such that $(u, v) \in E \Leftrightarrow (g(u), g(v)) \in E'$.

Given a function $f : A \rightarrow B$, we denote $\text{img}(f)$ the subset of B which contains all the elements b such that $f(a) = b$ for some $a \in A$. For A' any subset of A , we denote $f|_{A'} : A' \rightarrow B$ the function defined as $\forall a \in A', f|_{A'}(a') = f(a')$. For $f : A \rightarrow B$ and $g : B \rightarrow C$ two functions, we denote $g \circ f : A \rightarrow C$ the composition of f and g defined such that for all $a \in A$, $(g \circ f)(a) = g(f(a))$. We will be abusing this notation in some specific cases throughout this paper, and every instance of this abuse of notation will be explicitly defined in context. We say that a function $f : A \rightarrow B$ is injective if and only if for two $a, a' \in A$, $a \neq a' \Rightarrow f(a) \neq f(a')$. For $f : A \rightarrow B$ an injective function, we define the inverse of f , denoted

$f^{-1} : \text{img}(f) \rightarrow A$, as the injective function which to any $b \in \text{img}(f)$ associates the only value a for which $f(a) = b$ holds. We denote \mathbb{N}^* the set of strictly positive integers.

2.2 Automata networks

Let Σ be some finite alphabet, and n a positive integer. The set of all words of size n over Σ is denoted Σ^n , and we call such words *configurations*. Given a configuration w , we denote \bar{w} the bi-infinite (that is, infinite in both directions) word defined as the endless repetition of w . For $x \in \Sigma^n$, and for any index $1 \leq k \leq n$, we denote x_k the k -th letter of the configuration x . An *automata network (AN)* is a function $F : \Sigma^n \rightarrow \Sigma^n$, where n is the size of the network. We also call F the *global function* of the network. The global function F can be divided into functions that are local to each automaton: $\forall k, f_k : \Sigma^n \rightarrow \Sigma$, which we call *local functions*. The global function can be redefined as the parallel application of each local function : $\forall 1 \leq k \leq n, F(x)_k = f_k(x)$. For convenience, the set of automata $\{1, \dots, n\}$ is denoted S , and the automata are denoted using letters, starting from a . We will abuse notations and for any automaton a write x_a to mean the value of a in x , whatever the corresponding index might be. Similarly, the notation $F(x)_a$ refers to the value of a in the configuration obtained by the application of the network F on x . In general, the set of configurations Σ^n will be equivalently defined as the set Σ^S , the set of words which associate a letter in Σ to every automata in S . For $g : S' \rightarrow S$ some function, S' a set and $x \in \Sigma^S$ some configuration, we denote $x \circ g$ as the configuration over S' such that for any $s' \in S'$, $(x \circ g)_{s'} = x_{g(s')}$.

For (i, j) any pair of automata, i is said to *influence* j if and only if changing the value of x_i can change the value of $f_j(x)$. More formally, i influences j if and only if there exists $x, x' \in \Sigma^n$ such that $\forall k \in \{1, 2, \dots, n\}, x_k = x'_k \Leftrightarrow k \neq i$ and $f_j(x) \neq f_j(x')$.

We denote $\mathbb{B} = \{0, 1\}$ the Boolean set. An AN is called Boolean if it is defined over the alphabet \mathbb{B} .

Example 1. Let $S = \{a, b, c\}$. Let F be the Boolean AN with local functions $f_a(x) = x_c$, $f_b(x) = x_a$ and $f_c(x) = \neg x_a \vee x_b$.

It is common to represent an automata network F as a digraph using its automata as nodes so that (i, j) is an edge if and only if i influences j . This digraph is called the *interaction digraph* of F . The automata network described in Example 1 is illustrated as an interaction digraph in Figure 1.

Given an automata network F , we define the *dynamics* of F as the digraph with the configurations $x \in \Sigma^n$ as nodes such that (x, y) is an edge of the dynamics if and only if $F(x) = y$. The *limit dynamics* of F is the subgraph which contains only the limit cycles and the fixed points of the dynamics. In other terms, $x \in \Sigma^n$ is part of the limit dynamics of F if and only if there exists some $k \in \mathbb{N}^*$ such that $F^k(x) = x$. The dynamics of the network from Example 1 are illustrated in Figure 2.

As the dynamics of a network are a graph that is exponential in size

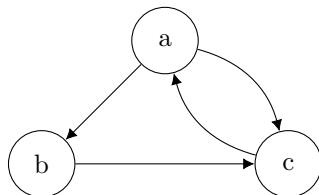


Figure 1: Interaction digraph of the network F described in Example 1 with local functions $f_a(x) = x_c$, $f_b(x) = x_a$ and $f_c(x) = \neg x_a \vee x_b$.

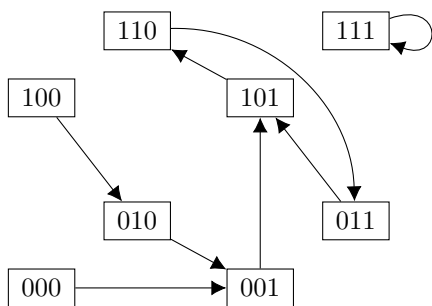


Figure 2: Dynamics of the network F as described in Example 1 with local functions $f_a(x) = x_c$, $f_b(x) = x_a$ and $f_c(x) = \neg x_a \vee x_b$.

compared to the number of automata, directly computing the limit dynamics of a family of networks is a computationally expensive process.

2.3 Modules

Working with the limit dynamics of automata networks is difficult. Our general approach is to divide the networks in smaller networks, in the hope that understanding the limit behavior of the smaller networks would inform us on the limit behavior of the larger network.

We introduced modules in [15] as a way to execute this division. Modules are automata networks with added inputs. For Σ a finite alphabet, we define a module as a function $M : \Sigma^{n+m} \rightarrow \Sigma^n$, where n is the number of automata of the network and m is the number of inputs. Inputs have influence over automata, as the local function of a module are defined over $f : \Sigma^{n+m} \rightarrow \Sigma$. To update a module, it is necessary to decide how to evaluate the inputs of the model. For $i \in \Sigma^m$ some configuration over the inputs and $x \in \Sigma^n$ some configuration over the automata, the update of x with inputs i is denoted $M(x, i)$. Interaction graphs can be defined for modules in the same way as they are defined for ANs. A module is *acyclic* if and only if its interaction digraph is cycle-free. For convenience, the set of inputs $\{1, \dots, m\}$ is denoted I , and the inputs are denoted using greek letters, starting with α . Example 2 describes a module with

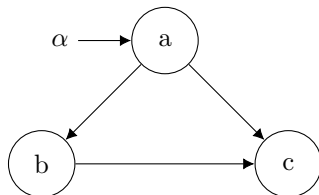


Figure 3: Interaction digraph of module M described in Example 2 with local functions $f'_a(x, i) = i_\alpha$, $f'_b(x, i) = x_a$ and $f'_c(x, i) = \neg x_a \vee x_b$.

three automata and one input, and its interaction digraph is represented in Figure 3.

Example 2. Let $S = \{a, b, c\}$ and $I = \{\alpha\}$. Let M be the module with local functions $f'_a(x, i) = i_\alpha$, $f'_b(x, i) = x_a$, $f'_c(x, i) = \neg x_a \vee x_b$. The module M is acyclic.

A module can be transformed into an automata network by the application of a total recursive wiring. A *total recursive wiring* is a function $\mathbf{w} : I \rightarrow S$, and its application to a module M results in the network denoted $\circ_{\mathbf{w}} M$ where the influence of any input $\alpha \in I$ is replaced by the influence of the automaton $\mathbf{w}(\alpha)$, that is, any literal i_α is substituted for the literal $x_{\mathbf{w}(\alpha)}$. The *output set* of a total recursive wiring \mathbf{w} is the set $\text{img}(\mathbf{w})$, which contains the automata used to replace the influence of the inputs of the module. Intuitively, the application of \mathbf{w} to M just ‘connects together’ the outputs and the inputs of the network, to form a loop. While the inputs are defined by the local functions of the module, the choice of the outputs is done through the choice of \mathbf{w} which is arbitrary and depends on the desired result. Example 3 describes the application of a wiring to the module defined in Example 2.

Example 3. Let M be as defined in Example 2. Let \mathbf{w} be the total recursive wiring such that $\mathbf{w}(\alpha) = c$. The resulting local functions of $\circ_{\mathbf{w}} M$ are $f''_a(x) = x_{\mathbf{w}(\alpha)} = x_c$, $f''_b(x) = x_a$ and $f''_c(x) = \neg x_a \vee x_b$. This network coincides with the network F as defined in Example 1. The output set of \mathbf{w} is $\text{img}(\mathbf{w}) = \{c\}$.

2.4 Output functions

A module M can be viewed as a machine which, given an initial configuration x and an infinite sequence of input configurations J , computes an infinite sequence of configurations over the automata. If M is acyclic, the resulting infinite sequence only depends on the initial configuration x up to a certain point, which is only for as many input configurations as the size of the longest path between an input and an output in M [16], after which all trace of the initial configuration x has effectively been ‘flushed out’ of the network. This is true because cycles are fundamental to the conservation of information in a network. This results in a shift of perspective, and our module M can now be seen as a machine that takes in an infinite sequence of configurations over the inputs, and produces the next

configuration of the network. This machine is called the output function of M , and is defined as follows.

An output function is a function O which takes in an infinite sequence of input sequences $J = (i^k)_{k \in \mathbb{N}^*}$ and outputs a configuration in Σ^n , that is, a configuration over the automata of the module. This infinite sequence is called the history of the inputs of the module, where the precedence of an input configuration grows with its index; that is, the input configuration i^1 is the most recent, i^2 precedes it in time, etc. To be correctly defined, the output function must only depend on a finite section of the infinite sequence J . For any automaton $s \in S$, we define the *local output function* O_s such that $O_s(J) = O(J)_s$. Example 4 presents the local output functions associated with each automaton of the module presented in Example 2, which is possible since that module is acyclic. In this example, the literal α_1 denotes the value of the input α in the input configuration i^1 of the sequence J . For any literal α_k , k is called the delay of the literal.

Example 4. Consider module M as developed in Example 2. As M is acyclic, we can define for it the following local output functions:

$$\begin{aligned} O_a(J) &= \alpha_1 \\ O_b(J) &= \alpha_2 \\ O_c(J) &= \neg\alpha_2 \vee \alpha_3, \end{aligned}$$

where J is an infinite sequence of configurations over I . For example, if $J = (0, 1, 1, \dots)$, we have that $\alpha_1 = 0, \alpha_2 = 1$ and $\alpha_3 = 1$. Hence, $O_a(J) = 0, O_b(J) = 1$ and $O_c(J) = \neg 1 \vee 1 = 1$.

When considering some acyclic module M paired with some total recursive wiring \mathfrak{w} , it has been shown in [16] that the limit dynamics of the automata network $\circ_{\mathfrak{w}} M$ only depend on the output functions of the nodes in the output set $\text{img}(\mathfrak{w})$. This theorem is paragraphed here as Lemma 1.

Lemma 1. Let M, M' be two acyclic modules on the automata sets S and S' respectively, and the same input set I . Let \mathfrak{w} be a total recursive wiring defined on M , and let $g : \text{img}(\mathfrak{w}) \rightarrow S'$ be an injective function. If for every infinite sequence of configurations of inputs J and every input $\alpha \in I$,

$$O_{\mathfrak{w}(\alpha)}(J) = O'_{(g \circ \mathfrak{w})(\alpha)}(J) \circ g \quad (\in \Sigma^{\text{img}(\mathfrak{w})})$$

then the automata networks $\circ_{\mathfrak{w}} M$ and $\circ_{g \circ \mathfrak{w}} M'$ have isomorphic limit dynamics.

Informally, this lemma expresses that if two automata networks have equivalent output functions up to some function g on their outputs, then the limit dynamics resulting from recursively wiring these modules into networks in a symmetric way respective to g are equivalent up to isomorphism. This still holds even if the networks have different sizes. In other terms, as long as we are only interested in the limit behavior of a network defined by an acyclic module M and a total recursive wiring \mathfrak{w} , studying its output functions over $\text{img}(\mathfrak{w})$ is enough.

For the rest of the article, we consider an automata network F as being described by an acyclic module M paired with a total recursive wiring \mathfrak{w} such that $\circ_{\mathfrak{w}} M = F$. And since we are only interested in the

limit dynamics of any given network, we use Lemma 1 and only consider necessary the description of the local output functions of the automata in $\text{img}(\mathfrak{w})$. For convenience, we will call the *sufficient output function* of M and denote $O_{\mathfrak{w}}$ the output function which takes an infinite sequence $J = (i^k)_{k \in \mathbb{N}^*}$ and produces a configuration over $\Sigma^{|\text{img}(\mathfrak{w})|}$, such that for all $s \in \text{img}(\mathfrak{w})$, $O_{\mathfrak{w}}(J)_s = O_s(J)$. In other terms, $O_{\mathfrak{w}}$ is the function that takes an infinite sequence of configurations over the inputs of M , and produces a configuration on the outputs of the network. The sufficient output function of the acyclic module developed in Example 2 and total recursive wiring from Example 3 is described in Example 5.

Example 5. *Consider the module M as developed in Example 2, and the total recursive wiring \mathfrak{w} developed in Example 3. As $\text{img}(\mathfrak{w}) = \{c\}$, the sufficient output function of M is $O_{\mathfrak{w}}(J) = O_c(J) = \neg\alpha_2 \vee \alpha_3$. For any module M' and total recursive wiring \mathfrak{w}' such that $O_{\mathfrak{w}'}(J) = \neg\alpha_2 \vee \alpha_3$, we can apply Lemma 1 which yields that $\circlearrowleft_{\mathfrak{w}} M$ and $\circlearrowleft_{\mathfrak{w}'} M'$ have isomorphic limit dynamics.*

2.5 Strictly one-way cellular automata

For Σ an alphabet and $r \in \mathbb{N}$, a *cellular automaton* (CA) is defined by a local rule $f : \Sigma^{2r+1} \rightarrow \Sigma$. The number r is the radius of the CA, and its global rule is defined as $F : \Sigma^{\mathbb{Z}} \rightarrow \Sigma^{\mathbb{Z}}$, with $F(x)_k = f(x_{k-r}, x_{k+1-r}, \dots, x_{k+r})$. A given CA is said to be *strictly one-way* if its local function is defined over $f : \Sigma^r \rightarrow \Sigma$, and its global rule as $F(x)_k = f(x_1, x_2, \dots, x_r)$ instead. Example 6 presents a strictly one-way CA.

Example 6. *Let $\Sigma = \mathbb{B}^2 = \{00, 01, 10, 11\}$. The local rule $f(x) = x_1 \oplus x_2$, where \oplus denotes the bit by bit XOR operator, has radius 2 and defines a strictly one-way cellular automaton.*

3 Associating sufficient output functions and local rules

Using the notion of global output function defined in Section 2.4 and the definition of strictly one-way CA detailed in Section 2.5, we define a transformation from any sufficient output function $O_{\mathfrak{w}}$ to the local rule of a strictly one-way cellular automata. Multiple steps are involved in this transformation, and we start with some more definitions around the sufficient output function $O_{\mathfrak{w}}$.

Let us define $\psi_{\mathfrak{w}} : \Sigma^{|\text{img}(\mathfrak{w})|} \rightarrow \Sigma^I$ such that for all $x \in \Sigma^{|\text{img}(\mathfrak{w})|}$ and $\alpha \in I$, $\psi_{\mathfrak{w}}(x)_{\alpha} = x_{\mathfrak{w}(\alpha)}$. The function $\psi_{\mathfrak{w}}$ effectively takes a configuration defined over the outputs of M and transforms it into a configuration over the inputs of M , following the wiring instructions provided by \mathfrak{w} . Let $x \in \Sigma^{|\text{img}(\mathfrak{w})|}$ denote the configuration obtained by calling $O_{\mathfrak{w}}$ over some sequence J . We define $i \in \Sigma^I$ the *input configuration derived from x* such that for any $\alpha \in I$, $i_{\alpha} = x_{\mathfrak{w}(\alpha)}$. This input configuration can be equivalently defined as $i = \psi_{\mathfrak{w}}(x|_{\text{img}(\mathfrak{w})})$. We call *update* of J using $O_{\mathfrak{w}}$ the operation denoted $(\psi_{\mathfrak{w}} \circ O_{\mathfrak{w}})(J)$ which produces a sequence $J' = (i'^k)_{k \in \mathbb{N}^*}$

defined such that for all $k > 1$, $i'^k = i^{k-1}$ (with i^{k-1} being the element of index $k - 1$ in the sequence J), and i'^1 is the input configuration derived from $O_{\mathfrak{w}}(J)$. Effectively, J' is obtained by shifting the infinite sequence J by one, and inserting the update of J using $O_{\mathfrak{w}}$ as its new first element.

Repeating this update operation starting from some sequence J will always result in a periodic behavior. This is true because the update of J using some sufficient output function $O_{\mathfrak{w}}$ is a deterministic process, and the fact that $O_{\mathfrak{w}}$ by definition only depends on a finite portion of J . We call J a periodic sequence generated by $O_{\mathfrak{w}}$ if there exists $m \in \mathbb{N}^*$ such that $(\psi_{\mathfrak{w}} \circ O_{\mathfrak{w}})^m(J) = J$. It is a consequence of Lemma 1 that these periodic sequences correspond one-to-one to the fixed points and limit cycles of any automata network from which the sufficient output function $O_{\mathfrak{w}}$ can be derived. At the end of all this, $O_{\mathfrak{w}}$ is just another way to express the limit computation done by an automata network. This is the central result of this paper, and the most significant step in the transformation from automata networks to strictly one-way cellular automata. We formalise this step in the following property.

Property 1. *Let M be an acyclic module and \mathfrak{w} a total recursive wiring. Any configuration part of a limit cycle of size $m \geq 1$ in the limit dynamics of the automata network $\circlearrowleft_{\mathfrak{w}} M$ has a one-to-one correspondence with an infinite sequence J for which $(\psi_{\mathfrak{w}} \circ O_{\mathfrak{w}})^m(J) = J$ holds.*

Proof. This proof shows how to construct the sequence J from any limit cycle, and how to construct a limit cycle from any sequence J , then shows that these transformations describe a pairing between the two sets.

Let x be a configuration over $\circlearrowleft_{\mathfrak{w}} M$. We denote $x|_{\text{img}(\mathfrak{w})}$ as the configuration in $\Sigma^{\text{img}(\mathfrak{w})}$ such that for all $s \in \Sigma^{\text{img}(\mathfrak{w})}$, $(x|_{\text{img}(\mathfrak{w})})_s = x_s$. We now suppose that x is part of a limit cycle of size m in the limit dynamics of $\Sigma^{\text{img}(\mathfrak{w})}$. Let (x^1, x^2, \dots) denote the infinite sequence defined such that $x^1 = x$, and for all $i > 1$, $x^i = \circlearrowleft_{\mathfrak{w}} M(x^{i-1})$. This backward recursive definition is possible by the hypothesis that x is part of the limit dynamics, which implies that this infinite sequence is actually an infinite repetition of the limit cycle x is part of, ending in x , and reversed. We now define the resulting sequence

$$J = (\psi_{\mathfrak{w}}(x^1|_{\text{img}(\mathfrak{w})}), \psi_{\mathfrak{w}}(x^2|_{\text{img}(\mathfrak{w})}), \dots).$$

Note that this sequence has period m . Let us now prove that $\psi_{\mathfrak{w}}(x^1|_{\text{img}(\mathfrak{w})}) = (\psi_{\mathfrak{w}} \circ O_{\mathfrak{w}})(\psi_{\mathfrak{w}}(x^2|_{\text{img}(\mathfrak{w})}), \psi_{\mathfrak{w}}(x^3|_{\text{img}(\mathfrak{w})}), \dots)$. By definition $O_{\mathfrak{w}}$ accurately computes the values of the automata in $\text{img}(\mathfrak{w})$ provided an infinite history of the values over the inputs of the network. The sequence x^2, x^3, \dots is an history of the values of the network over the attractor which contains x stopping just before x , with x^2 being the direct successor to x^3 , and so on. The sequence $x^2|_{\text{img}(\mathfrak{w})}, x^3|_{\text{img}(\mathfrak{w})}, \dots$ is the same history, but restricted to the automata that are part of the output set of the network. Finally the sequence $\psi_{\mathfrak{w}}(x^2|_{\text{img}(\mathfrak{w})}), \psi_{\mathfrak{w}}(x^3|_{\text{img}(\mathfrak{w})}), \dots$ is the periodic sequence of the configurations of inputs that generates the limit cycles which contains x in M . By the definition of output functions (from which Lemma 1 is a consequence), it holds that $O_{\mathfrak{w}}(\psi_{\mathfrak{w}}(x^2|_{\text{img}(\mathfrak{w})}), \psi_{\mathfrak{w}}(x^3|_{\text{img}(\mathfrak{w})}), \dots) = x^1|_{\text{img}(\mathfrak{w})}$, which implies that $(\psi_{\mathfrak{w}} \circ O_{\mathfrak{w}})(\psi_{\mathfrak{w}}(x^2|_{\text{img}(\mathfrak{w})}), \psi_{\mathfrak{w}}(x^3|_{\text{img}(\mathfrak{w})}), \dots) =$

$\psi_{\mathfrak{w}}(x^1 \parallel_{\text{img}(\mathfrak{w})})$. Note that this argument is agnostic to the choice of x inside of the limit cycle that contains it. For J the sequence derived this way from the configuration x , it holds that the input sequence $(\psi_{\mathfrak{w}} \circ O_{\mathfrak{w}})(J)$ is derived from $\circlearrowleft_{\mathfrak{w}} M(x)$; both update equivalently. As a result, and since the limit cycle that contains x is of period m , the sequence J is such that $(\psi_{\mathfrak{w}} \circ O_{\mathfrak{w}})^m(J) = J$.

We now argue that this transformation works the same from the other direction; any sequence J such that $(\psi_{\mathfrak{w}} \circ O_{\mathfrak{w}})^m(J) = J$ can be used to reconstruct a limit cycle of same periodicity starting at a specific point. This is true because we can use the output function defined by M and \mathfrak{w} (and not the sufficient output function) to generate configurations over S from J , and it holds that the resulting sequence is periodic and corresponds to a limit cycle in the dynamics of $\circlearrowleft_{\mathfrak{w}}$. It also holds that both transformations are reverses of each other. It results that these transformations define a pairing between the limit cycles of size m in $\circlearrowleft_{\mathfrak{w}} M$ and the periodic sequences J such that $(\psi_{\mathfrak{w}} \circ O_{\mathfrak{w}})^m(J) = J$. \square

We now transform $O_{\mathfrak{w}}$ directly into the local rule of a strictly one-way cellular automata. For $O_{\mathfrak{w}}$ a sufficient output function, we define the associated local rule as $f_{\mathfrak{w}}(x) = (\psi_{\mathfrak{w}} \circ O_{\mathfrak{w}})(J_x)$, where J_x is the finite sequence of inputs defined as $J_x = (i^k)_{k \in \{1, \dots, r\}}$ such that $i^k = x_k$ for any $k \in \{1, \dots, r\}$. It is important to note that this definition implies an abuse of notation; while $O_{\mathfrak{w}}$ only takes as input infinite sequences, by definition it only depends on a finite portion of it. The radius r is thus defined as the exact length of the portion on which $O_{\mathfrak{w}}$ depends, in which case the finiteness of J_x poses no practical problems. We call the associated CA of the sufficient output function $O_{\mathfrak{w}}$ the CA defined by its associated local rule. The associated CA of the sufficient output function developed in Example 5 is detailed in Example 7.

Example 7. *Consider the sufficient output function $O_{\mathfrak{w}}$ developed in Example 5. Its associated CA is defined on the alphabet \mathbb{B} and its local rule $f_{\mathfrak{w}}$ is defined as $f_{\mathfrak{w}}(x) = \neg x_2 \vee x_3$. The fixed points of that automata network are the configurations $\overline{110}$, $\overline{101}$ and $\overline{011}$, which are all equivalent up to shifting, and the uniform configuration $\overline{1}$.*

We can now state the main result of this paper: the infinite sequences J for which $(\psi_{\mathfrak{w}} \circ O_{\mathfrak{w}})^m(J) = J$ for some $m \geq 1$ have a one-to-one correspondence with the fixed points of the cellular automata defined by the associated local rule $f_{\mathfrak{w}}$. This fact, chained with the equivalence showed in Property 1, is stated as the following theorem.

Theorem 1. *Let M be an acyclic module and \mathfrak{w} a total recursive wiring. Any configuration part of a limit cycle in the limit dynamics of the automata network $\circlearrowleft_{\mathfrak{w}} M$ has a one-to-one correspondence with a fixed point of the strictly one-way cellular automata with local rule $f_{\mathfrak{w}}$.*

Proof. Let x be part of a limit cycle of size m in the limit dynamics of $\circlearrowleft_{\mathfrak{w}} M$. We use Property 1 to obtain an infinite sequence J which corresponds to the limit cycle x , such that $(\psi_{\mathfrak{w}} \circ O_{\mathfrak{w}})^m(J) = J$. The sequence J can be used to update the local rule $f_{\mathfrak{w}}$, as it is defined on the same alphabet. Let us now consider a bi-infinite sequence J^+ , which is

the limit of the process of $(\psi_{\mathfrak{w}} \circ O_{\mathfrak{w}})(J)$. This is the sequence obtained by continuing the repetition of the period in J infinitely many times, in both directions. This sequence J^+ is now a configuration over $(\Sigma^I)^{\mathbb{Z}}$, and we argue that it is a fixed point of the cellular automata with local rule $f_{\mathfrak{w}}$. This directly derives from the fact that this sequence is constructed from repeated calls to $(\psi_{\mathfrak{w}} \circ O_{\mathfrak{w}})$, and that $f_{\mathfrak{w}}(J) = (\psi_{\mathfrak{w}} \circ O_{\mathfrak{w}})(J)$ by definition. his process maps every initial configuration x (there are m different such initial configurations) to this infinite sequence J^+ in a specific shift, which results in m different fixed point of the automata, all equivalent up to shifting.

This process is equivalent from the other direction. By considering some fixed point of the cellular automata, we show that it always has some period m ; this results from the deterministic and limited nature of the local rule $f_{\mathfrak{w}}$. This fixed point can then be decomposed into a sequence J^+ which can be cut at any point to produce a sequence J which, by application of Property 1, results in a configuration part of a limit cycle in $\circ_{\mathfrak{w}} M$. Both of these transformations are inverses of each other, hence the result. \square

As an example of the application of this theorem, we can observe that the limit dynamics contained in the dynamics in Figure 2 and the fixed points of the cellular automata presented in Example 7 are equivalent; the fixed points $\bar{1}10$, $10\bar{1}$ and $0\bar{1}\bar{1}$ of the cellular automata correspond to the configurations 110, 101 and 011 of the automata network, and the uniform fixed point $\bar{1}$ corresponds to the fixed point 111 in the dynamics of the automata network. Note that the syntactical equivalence of these objects does not hold in general, and is a consequence of the choice of the automata network, as more than one automata network can be described using a given sufficient output function $O_{\mathfrak{w}}$ (Lemma 1).

Theorem 1 is made stronger by the observation that any strictly one-way cellular automata is the associated cellular automata $f_{\mathfrak{w}}$ of some module M with total recursive wiring \mathfrak{w} . Which means that this result goes both ways; the limit dynamics of an automata network corresponds to the fixed points of an associated cellular automaton, but also the fixed point of any strictly one-way cellular automaton correspond to the limit dynamics of an associated automata network.

More formally, let us consider f the local rule of a strictly one-way cellular automaton of radius r and alphabet Σ . Transforming f into an automata networks takes multiple steps, which operate in the opposite direction from the previous transformation. The resulting automata network will need a set of automata S , which is not specified outside of the fact that it contains an automaton named s . Through this process, we will define an acyclic module M and associated total recursive wiring \mathfrak{w} , for which we define the set of inputs $I = \{\alpha\}$, a singleton. Let us now transform. In the previous transformation, the local function $f_{\mathfrak{w}}$ was defined as the function $f_{\mathfrak{w}}(x) = (\psi_{\mathfrak{w}} \circ O_{\mathfrak{w}})(J_x)$, in which J_x was the values in x were encoded as an infinite sequence. In this transformation, we operate in reverse; from the function f , we define the *associated sufficient output function* denoted O_f as $O_f(J) = \mathfrak{w}^{-1} \circ f(x^J)$, where x^J is a configuration such that $x_k^J = J_k$, for any $1 \leq k \leq r$, and in which the total recursive

wiring \mathfrak{w} is defined as $\mathfrak{w} : \alpha \mapsto s$, hence $\mathfrak{w}^{-1} : s \mapsto \alpha$. As a reminder, s is an automaton of S . We now consider M an acyclic module which when associated with \mathfrak{w} generates a sufficient output function $O_{\mathfrak{w}}$ equal to O_f . This acyclic module always exists, and since it only has one input, it can be computed with a minimal amount of automata in polynomial time [17]. Let us denote M_f and \mathfrak{w}_f such an associated acyclic module and total recursive wiring derived from f .

Example 8. *Let us consider the local rule f developed in Example 6, and the set of inputs $I = \{\alpha\}$. Taking the alphabet $\mathbb{B}^2 = \{00, 01, 10, 11\}$, let O_f be a sufficient output function such that*

$$O_f(J) = \alpha_1 \oplus \alpha_2,$$

where \oplus is the bit by bit XOR operator. In this example, O_f is the associated sufficient output function of f . Let us construct an example of an associated acyclic module M_f and total recursive wiring \mathfrak{w}_f . We define $S = \{a, b\}$, and for M_f the following local functions:

$$\begin{aligned} f_a(x, i) &= i_\alpha \\ f_b(x, i) &= i_\alpha \oplus x_a. \end{aligned}$$

We also define $\mathfrak{w}_f(\alpha) = b$, which implies that the output set of M_f is $\text{img}(\mathfrak{w}_f) = \{b\}$. Here are the local output functions of M_f :

$$\begin{aligned} O_a(J) &= \alpha_1 \\ O_b(J) &= \alpha_1 \oplus \alpha_2. \end{aligned}$$

Hence the sufficient output function of M_f is equivalent to O_b , which is equivalent to O_f , and by application of Lemma 1 the limit dynamics of $\circlearrowleft_{\mathfrak{w}_f} M_f$ are equivalent to the fixed points of the cellular automata with local rule f .

The fact that any strictly one-way cellular automata can be transformed into an automata network is stated as the following corollary.

Corollary 1. *For f the local rule of a strictly one-way cellular automaton F and M_f , \mathfrak{w}_f the associated acyclic module and total recursive wiring, the fixed points of F correspond one-to-one with the limit cycles of the automata network $\circlearrowleft_{\mathfrak{w}_f} M_f$.*

Proof. While transforming a CA into an associated automata network is not the exact opposite transformation from computing the associated CA of an automata network, it holds that for any strictly one-way CA F , applying the transformation into an automata network and then back to a CA is the same as doing nothing at all. Hence any F can be redefined as the associated cellular automata of some automata network $\circlearrowleft_{\mathfrak{w}} M$, and the result follows from the application of Theorem 1. \square

4 An application: interlacing cellular automata

As an example of the usefulness of this correspondence, we propose a generalisation of a known result about automata networks [8]: if the length

of all the cycles in the interaction digraph of an AN are divisible by some factor d greater than 1, then this AN is reducible into a smaller AN, from which its limit dynamics can be derived. This theorem has an intuitive correspondence in strictly one-way cellular automata. Before stating the result, some extra formalism is needed.

Let us define that for $f : A^B \rightarrow C$ some function and $b \in B$, the variable of index b is said to influence f if and only if there exist $x, x' \in A^B$ such that $x|_{B \setminus \{b\}} = x'|_{B \setminus \{b\}}$, $x_b \neq x'_b$ and $f(x) \neq f(x')$.

For some positive integers r and d such that $d|r$, let $\mu_d : \Sigma^{\{1, \dots, \frac{r}{d}\}} \rightarrow \Sigma^{\{1, \dots, r\}}$ the function defined by $\mu_d(x)_k = x_{d \times k}$.

Let Σ be a set of bi-infinite words, and d a positive integer. The *d-interlacing* of Σ is the set of bi-infinite words Σ^d such that $w' \in \Sigma^d$ if and only if there exists some sequence of words $\{w^1, w^2, \dots, w^d\}$ in Σ such that for all $k \in \mathbb{Z}$, $w'_k = w^a_b$, for a and b the reminder and quotient of the division of k by d respectively. This word is also denoted $w' = w^1 \sim w^2 \sim \dots \sim w^d$.

Theorem 2. *Let f be the local rule of a strictly one-way CA. If there exists some integer $d > 1$ that divides all integers in $\{k \mid x_k \text{ influences } f\}$, then the set of fixed points of the CA with local rule f is the d -interlacing of the set of fixed points of the CA with local rule $f \circ \mu_d$.*

Proof. Let C_z^1 be the set of cells which contains c_z and the cells that influence the update of cell c_z . We define C_z^{n+1} as the union of C_z^n and the set of cell that influence any cell in C_z^n . As there exists some factor d that divides the distance between any cell and its influences, this property is also true of any cell in C_z^ω . The same argument follows for the cells influenced by the cell at z .

Taking cells of index $a + d \times b$ for some a and all b , we obtain a band of cells that are independant from every other. That is, their values does not have influence over the value of the rest of the configuration, and the rest of the configuration does not influence their values; as such, we can consider every band (one defined by each $a < d$) as a different cellular automata with local rule $f \circ \mu_d$.

From this, the result is obtained by observing that any fixed point in the CA with local rule f is obtained by composing the independant fixed points of the d bands which can be simulated by the CA with local rule $f \circ \mu_d$. To compose all possible fixed points in the right shape we use the notion of d -interlacing, which intuitively constructs a valid fixed point for the CA with local rule f . \square

Example 9. *Let us consider the Boolean local rule $f(x) = \neg x_2 \vee x_4$. As all the indexes of the variables that influence f are divisible by $d = 2$, the fixed points of the CA with local rule f are therefore the 2-interlacing of the fixed points of the CA with local rule $f \circ \mu_2$. This rule is defined as $f \circ \mu_2(x) = \neg x_1 \vee x_2$, and the fixed points of the related CA are $\overline{01}$ and $\overline{1}$, and their shifted equivalents. Let us name those fixed points a and b respectively. By application of the 2-interlacing, the fixed points of the CA with local rule f are $a \sim a = \overline{0011}$, $a \sim b = \overline{0111}$, $b \sim a = \overline{1011} \equiv \overline{0111}$ and $b \sim b = \overline{11} = \overline{1}$, for a total of 3 distinct fixed points up to shifting.*

5 Other equivalent combinatorial problems

In the pursuit of more combinatorial intuition in the nature of the correspondence highlighted in Theorem 1, the following section explores other objects which can be proven to be equivalent to the dynamics of automata networks and thus equivalent to the fixed points of strictly one-way cellular automata.

5.1 Subshifts of finite type

Subshifts of finite type are collections of infinite words usually defined through sets of local allowed (or equivalently forbidden) patterns.

It is known that subshifts of finite type and cellular automata are very similar objects; as any example of the latter can be seen as an example of the former with an extra dimension. However, since in our case we are only interested in the fixed points of our cellular automata, it is thus very natural to consider the equivalent one-dimensional subshift that characterises our problem.

Remark 1. *For any sufficient output function $O_{\mathfrak{w}}$ of some acyclic module M and total recursive wiring \mathfrak{w} , the associated cellular automata can be converted to a one dimensional subshift of finite type, using the set of allowed patterns defined by the local rule of the cellular automata. The words allowed by this subshift of finite type are equivalent to the limit dynamics of $\circ_{\mathfrak{w}} M$.*

Remark that encoding a function using allowed or forbidden patterns can be an expensive process if the initial function is encoded as, for example, a circuit, as it is an equivalent process to the enumeration of the table of all the possible evaluations of that function.

5.2 Aperiodic necklaces

Necklaces are cyclic words, finite words without a beginning nor an end. They are a classical model of combinatorial problems. As the fixed points of our strictly one-way cellular automata always are periodic, we can focus on characterising their period, which leads to using necklaces to describe them.

Remark 2. *For any sufficient output function $O_{\mathfrak{w}}$ of some acyclic module M and SB the associated subshift of finite type constructed from the cellular automata associated to $O_{\mathfrak{w}}$, the sequences admitted by SB are all periodic. We can define them as equivalent aperiodic necklaces defined by the same allowed patterns as SB .*

The same remarks applies; the allowed patterns by a sufficient output function $O_{\mathfrak{w}}$ are exponential in number if $O_{\mathfrak{w}}$ is provided as a circuit. The aperiodic necklaces represented in Figure 4 are the aperiodic necklaces equivalent to the limit dynamics of the automata network detailed in Example 1, which are equivalent to the fixed points of the strictly one-way cellular automata detailed in Example 7.

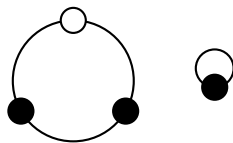


Figure 4: The two aperiodic binary necklaces that do not contain any pattern in $\{0b0, 1bb'0, 01b1 \mid b, b' \in \mathbb{B}\}$. Black beads stand in for 1, and white beads for 0. As these patterns encode the output function described in Example 7, these necklaces correspond to the limit cycles of size 3 and 1 showcased in Figure 2.

6 Final words

To us, the main interest of Theorem 1 is to show that the intricate task of describing the limit behavior of automata networks, which are diverse in both interaction graphs and local functions, can actually be done by describing the fixed points of cellular automata, with their uniform structure and local behavior. In another perspective, this shows that the strictly one-way property in cellular automata is strongly restrictive, since it implies that their fixed points can be described through a finite object.

We believe that the connections described in this paper are valuable. We aim to continue the study of strictly one-way cellular automata, in particular of their orbits, in the hope of results that could have meaning in the domain of automata networks, as well as to bring more results concerning automata networks in the domain of cellular automata.

Acknowledgments. This work has been partially funded by ANR-18-CE40-0002 FANs project and ECOS-Sud CE19E02.

References

- [1] TV Alekseyevskaya and AG Malenkov. Mathematical model of the initiation of near weekly fluctuations in tissue (cellular automata). *Biophysics*, 36(2):356–360, 1991.
- [2] K. Bhattacharjee, N. Naskar, S. Roy, and S. Das. A survey of cellular automata: types, dynamics, non-uniformity and applications. *Natural Computing*, 19:433–461, 2020.
- [3] B. Chopard and M. Droz. Cellular automata modeling of physical systems, 2005.
- [4] M.I. Davidich and S. Bornholdt. Boolean network model predicts cell cycle sequence of fission yeast. *PLoS One*, 3:e1672, 2008.
- [5] J. Demongeot, E. Goles, M. Morvan, M. Noul, and S. Sené. Attraction basins as gauges of robustness against boundary conditions in biological complex systems. *PLoS One*, 5:e11793, 2010.
- [6] J. Demongeot, M. Noul, and S. Sené. Combinatorics of Boolean automata circuits dynamics. *Discr. Appl. Math.*, 160:398–415, 2012.

- [7] MA Fuentes and MN Kuperman. Cellular automata and epidemiological models with spatial dependence. *Physica A: Statistical Mechanics and its Applications*, 267(3-4):471–486, 1999.
- [8] Z. Gao, X. Chen, and T. Başar. Stability structures of conjunctive boolean networks. *Automatica*, 89:8–20, 2018.
- [9] J. Kari. Theory of cellular automata: A survey. *Theoretical computer science*, 334(1-3):3–33, 2005.
- [10] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, 22:437–467, 1969.
- [11] M. Kutrib. Complexity of one-way cellular automata. In *International Workshop on Cellular Automata and Discrete Complex Systems*, pages 3–18. Springer, 2014.
- [12] L. Mendoza and E. R. Alvarez-Buylla. Dynamics of the genetic regulatory network for *Arabidopsis thaliana* flower morphogenesis. *J. Theor. Biol.*, 193:307–319, 1998.
- [13] K. Nagel and M. Schreckenberg. A cellular automaton model for freeway traffic. *Journal de physique I*, 2(12):2221–2229, 1992.
- [14] M. Noulal. *Updating Automata Networks*. PhD thesis, École Normale Supérieure de Lyon, 2012.
- [15] K. Perrot, P. Perrotin, and S. Sené. A framework for (de)composing with Boolean automata networks. In *Proc. of MCU’18*, volume 10881 of *LNCS*, pages 121–136, 2018.
- [16] K. Perrot, P. Perrotin, and S. Sené. On the complexity of acyclic modules in automata networks. In *Proc. of TAMC’20*, pages 168–180. Springer, 2020.
- [17] K. Perrot, P. Perrotin, and S. Sené. Optimising attractor computation in boolean automata networks. pages 68–80, 2021.
- [18] P Sarkar. A brief history of cellular automata. *Acm computing surveys (csur)*, 32(1):80–107, 2000.
- [19] R. Thomas. Boolean formalization of genetic control circuits. *J. Theor. Biol.*, 42:563–585, 1973.