

Attractor-Based Obstructions to Growth in Homogeneous Cyclic Boolean Automata

Bilal Khan^{1*}, Yuri Cantor² and Kirk Dombrowski¹

¹Department of Sociology, University of Nebraska-Lincoln, Lincoln, Nebraska, USA

²Department of Computer Science, The Graduate Center, City University of New York, USA

Abstract

We consider a synchronous Boolean organism consisting of N cells arranged in a circle, where each cell initially takes on an independently chosen Boolean value. During the lifetime of the organism, each cell updates its own value by responding to the presence (or absence) of diversity amongst its two neighbours' values. We show that if all cells eventually take a value of 0 (irrespective of their initial values) then the organism necessarily has a cell count that is a power of 2. In addition, the converse is also proved: if the number of cells in the organism is a proper power of 2, then no matter what the initial values of the cells are, eventually all cells take on a value of 0 and then cease to change further. We argue that such an absence of structure in the dynamical properties of the organism implies a lack of adaptiveness, and so is evolutionarily disadvantageous. It follows that as the organism doubles in size (say from m to $2m$) it will necessarily encounter an intermediate size that is a proper power of 2, and suffers from low adaptiveness. Finally we show, through computational experiments, that one way an organism can grow to more than twice its size and still avoid passing through intermediate sizes that lack structural dynamics, is for the organism to depart from assumptions of homogeneity at the cellular level.

Keywords: Attractor-based obstructions; Cyclic boolean automata; Synchronous boolean; Robustness

Introduction

The subject of cellular automata has received much attention since John Von Neumann's seminal work [1] on the dynamics of a grid of cells which evolve in discrete time steps according to rules based on their neighbor's values (e.g., see the surveys in ref. [2,3]). Conway's Game of Life [4], perhaps the most famous example of cellular automata, consists of an infinite two-dimensional orthogonal grid of Boolean cells whose values are synchronously updated 1. Cellular automata are frequently studied by considering their collective dynamics. Wolfram [5], for example, examined the complexity of finding "Garden of Eden States" (i.e., states that are unreachable from any other state), as well as determining whether a network can reach a state in which all cells have value 1 (i.e., a question that is now known as the "All-Ones Problem"). Both these problems generally become computationally infeasible for all but the smallest one-dimensional networks [6].

A random graph model for automata was introduced by Stuart Kauffman in the course of his research on gene regulatory networks. These so-called "NK networks" [7] consist of N cells, each of which is connected to a randomly chosen subset of K cells. Kauffman and others considered self-organization and the spontaneous emergence of order [8] in NK and related networks. Consensus is a particular form of emergent order that has received particular attention, especially in the context of social systems. Miller considered consensus in the standing ovation problem as a means to examine behavior in social networks using computational models [9]. Arenas surveys network structures which lead to emergent features and reports on the implications of consensus emergence in a variety of settings [10]. In his work, the community structures of networks (i.e., clusters of densely interconnected cells, between which connections are sparse) play a crucial role. Ball describes how many natural systems rely on characteristics akin to community structure in order to reach a level of consensus robustly in the presence of noise [11]. Consensus problems are closely related to our research since both seeks to understand

dynamical systems which move towards uniformity irrespective of initial conditions [12], and to identify the social network properties that lead to stasis and uniformity [13].

The organisms we consider in this paper are discrete Boolean cellular automata of the NK type, though we restrict ourselves to $K=2$ [14] and require that the cells be connected deterministically to form a circle. Such cyclic networks have received considerable attention themselves [6]. Like most prior research, we too (at least initially) consider only cellular automata that are homogenous at the cellular level, that is, cyclic networks in which all cells operate according to an identical update rule. For simplicity, we only consider networks in which cells synchronously update their values—recent progress in sequential dynamical systems [15-17] has shown that the behavior of more general asynchronous systems with small temporal variations can be examined by "equivalent" synchronous systems [18].

Where the All Ones Problem asks if the state in which all cells have value 1 is reachable from any other state, here we seek to determine if there is a state that is reachable from every other state. The networks we will consider are so simple as to lack community substructures, and yet always reach consensus regardless of noise. This is possible because (as we shall prove) their dynamics exhibit a single unique attractor. This work extends earlier results on thermal robustness and attractor density in synchronously updated cyclic Boolean networks [19].

***Corresponding author:** Bilal Khan, Department of Sociology, University of Nebraska-Lincoln, Lincoln, Nebraska, USA, Tel: 2122378927; E-mail: grouptheory@gmail.com

Received October 17, 2015; **Accepted** October 29, 2015; **Published** November 04, 2015

Citation: Khan B, Cantor Y, Dombrowski K (2015) Attractor-Based Obstructions to Growth in Homogeneous Cyclic Boolean Automata. J Comput Sci Syst Biol 8:6 341-353. doi:10.4172/jcsb.1000209

Copyright: © 2015 Khan B, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Background

In this section we introduce basic terminology concerning Boolean automata through examples, and provide some motivating context. The terminology is rendered formally later, in Section 5.

In this paper we investigate cyclic (i.e., circular) organisms where each cell has an instantaneous value of either 0 or 1. The organism is homogenous in that it evolves over time by having each cell repeatedly apply the same update rule. Although there are many choices of update rules, the one we consider here is XOR, which takes into consideration just the two immediate neighbors of the cell [14]. The XOR rule specifies that if the values of a cell's two neighbors differ, then the cell takes a value of 1 at the next time step; conversely, if the values of a cell's two neighbors agree, then the cell takes a value of 0 at the next time step). So, for example, if a cell's two neighbors have values 0 and 1, then at the next time step, the cell will take on a value of 1—an example of such a cell appears at the top of the 5-cycle in Figure 1. On the other hand, if a cell's left and right neighbors both have value 0, then at the next time step, the cell will take on a value of 0—an example of such a cell appears at the bottom-left of the 5-cycle in Figure 1.

If one synchronously (i.e., simultaneously) applies the XOR update rule at each of the cells in Figure 1 then one arrives at the configuration in Figure 2. Each of these configurations is referred to as a state. If we fix index of the organism's cells, for example as shown in Figure 3, then the state depicted in Figure 1 can be named 00010, while the state depicted in Figure 2 can be named 00101. We say that the successor of 00010 is 00101. Following in this manner, one finds that the successor of 00101, is 11000, and the successor of 11000 is 11101 and its successor in turn is 00101. This sequence of state transitions is shown in Figures 4 and 5.

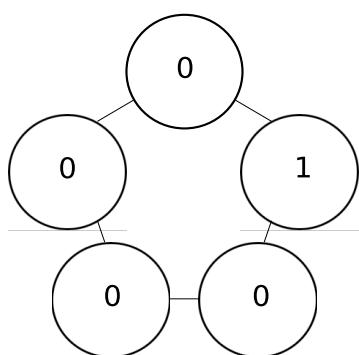


Figure 1: Initial state.

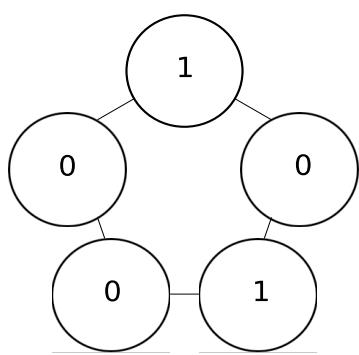


Figure 2: Successor state.

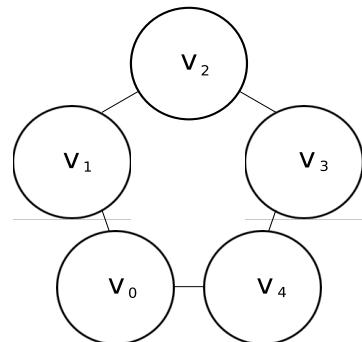


Figure 3: Size 5 network.

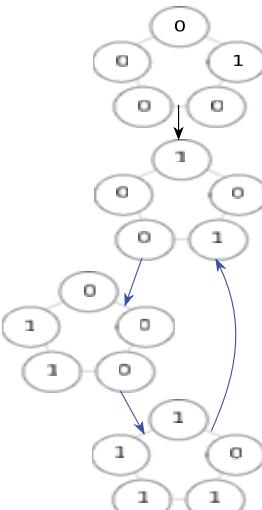


Figure 4: Successor states from Figure 1.

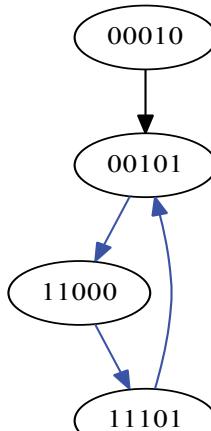


Figure 5: Attractor with network states.

Using software developed previously [19], we experimentally simulate the size 5 homogeneous XOR organism and output the dynamics as a file. From this output, we rendered Figure 5 using Graphviz [20]. If we start from the state 00101 and move forward 3 time steps, we

return back at the state 00101 (since $00101 \rightarrow 11000 \rightarrow 11101 \rightarrow 00101$), such a structure is called an attractor. Because it takes 3 time steps to go around this example attractor once, it is said to be an attractor of length 3.

Because the organism is of finite size, and each of its cells can only take on one of two values (either 0 or 1), there are only finitely many states the organism can be in. An organism of size 5 can be in one of $2 \times 2 \times 2 \times 2 \times 2 = 32$ different states; an organism with N cells can be in one of 2^N different states. Taken as a collection of states, this is referred to as the state space of the organism.

Suppose we draw an arrow from one state X to another state Y , whenever simultaneously applying XOR at each of the cells in the organism when it is in state X would lead to the organism being in state Y . Then each state would have exactly one arrow emerging from it, since the application of XOR is completely deterministic. The resulting directed network would have 2^N states as its vertices, and 2^N arrows as its edges. Such a rendering is called the dynamics graph of the organism. Figure 6, which we create in the same manner as before by simulating the network to produce an output file of the dynamics and

using Graphviz to render, shows the complete dynamics graph of the cyclic organism of size 5. We generate Figure 7 again using the same technique only this time with the binary output converted to decimal to show the same graph but vertices (states) labeled by the decimal equivalent of the binary name. From either of these diagrams, one can verify that there are 6 attractors in all. Five of these attractors have length 3, while one of them has length 1.

While the organism of size 5 has 6 attractors, it turns out the organism of size 6 has 10 attractors (4 of length 1, and 6 of length 2). This is seen in the dynamics graph of the organism of size 6, which is given in Figure 8. As before, we generate Figure 8 using Graphviz and the simulated dynamics output using our software. Indeed, as the organism grows from size 5, 6, 7, so on and onward, the number of attractors rises and falls abruptly. This is quantified in the plot in Figure 9 below, where the x-axis is the organism's size (number of cells), and the y-axis of this plot is the (base 2) logarithm of the number of attractors. Figure 9 is generated using collated experimental results from simulations of homogeneous XOR networks of sizes 2 through 20 where the output attractor counts are graphed using gnuplot [21]. We see from the plot that while the general trend is for the number of

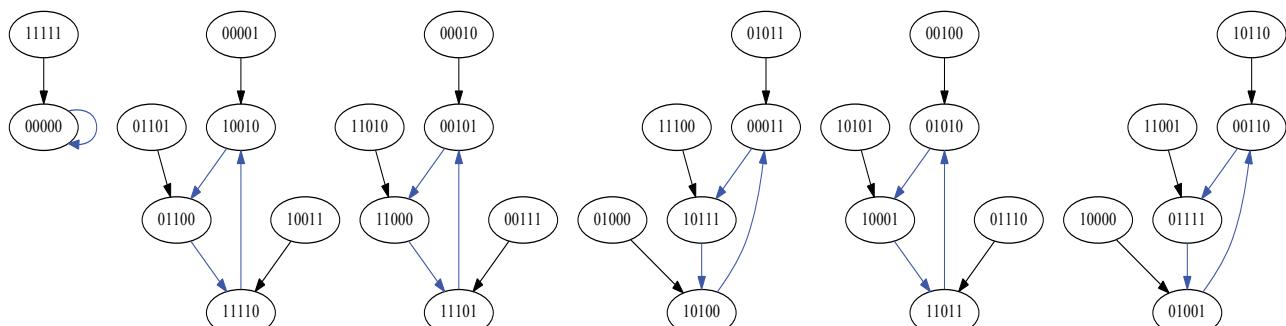


Figure 6: Dynamics graph of the cyclic organism of size 5, in binary.

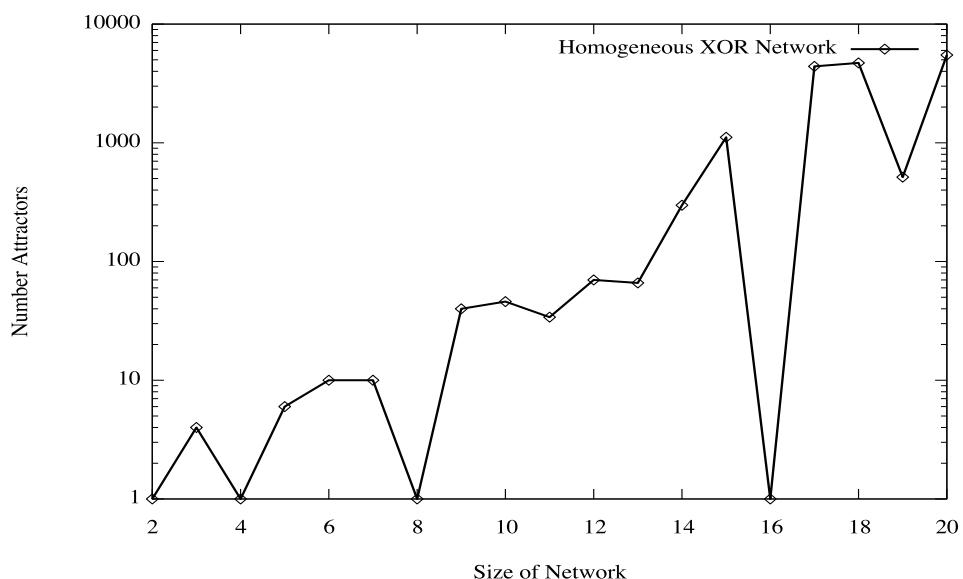


Figure 7: Number of Attractors vs Network Size.

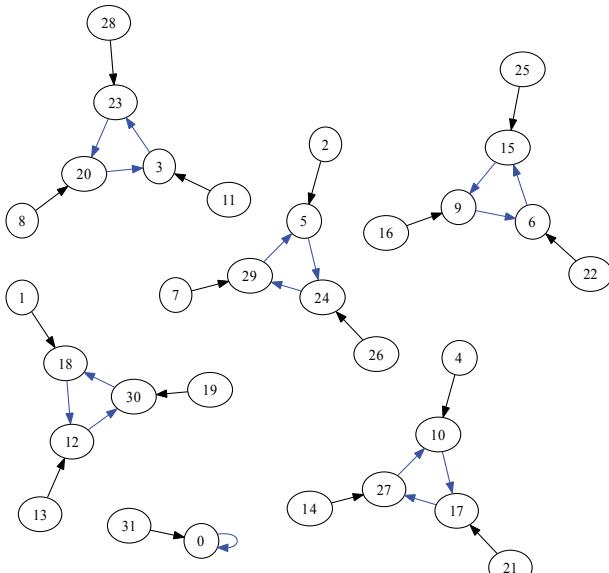


Figure 8: Dynamics graph of organism of size 5, in decimal.

attractors to grow exponentially as the organism gets larger, every so often the organism reaches a size for which the number of attractors plummets to 1—these are precisely those points where the curve passes through a point whose y -coordinate is $\log_2(1)=0$. One can see again from the plot in Figure 9 that this occurs precisely when the organism reaches size 2, 4, 8, and 16.

One might conjecture from the experimental data in the above graph that whenever the cyclic organism reaches a size that is a proper power of 2 (e.g., 2, 4, 8, 16, 32, 64, ...) then the number of attractors in its dynamics graph is exactly 1. The majority of this paper is devoted to proving both this statement, as well as its converse: if an organism has just one attractor in its dynamics graph, then the organism necessarily has a power of 2 many cells.

Interpreting the number of attractors

Biological organisms are subjected to Darwinian preferential selection based on the evolutionary advantages of their properties, within the context of the ecosystem. Dynamics of Boolean networks are simpler than biological networks while possessing some essential similar properties. To the extent that the mathematical model we have presented here is an idealized rendering of a biological organism, of what significance is the number of attractors? Let us examine, at the informal level of metaphor, what the implications of having a very “large” or very “small” number of attractors have on an organism within a real-world ecosystem. We will assume that this ecosystem provides the organism with two types of signals: environmental stimuli and thermal noise.

Environmental stimuli are high amplitude signals from the ecosystem to which the organism needs to be adaptive and react. Thermal noise, on the other hand, consists of low amplitude signals against which the organism needs to be robust and not react. An organism typically is spinning inside some attractor A . When it receives a signal from the outside, one or more of its cell’s values may be perturbed (the 0’s may become 1’s and vice versa). This effectively throws the organism out of its present state within A , over to some other vertex v in its dynamics graph. From v , the organism follows

the arrows until it lands once again in an attractor A' . The attractor A' may or may not be the same as the attractor A .

For example, if a size 5 organism (see Figure 7) is in the attractor $A=15\rightarrow 9\rightarrow 6\rightarrow 15$ and receives a stimulus, the organism might get thrown to state $v=28$, which case it would land in the attractor $A'=23\rightarrow 20\rightarrow 3\rightarrow 23$. In this case $A \neq A'$. If however, the stimulus resulted in the organism being thrown into state $v=22$, then it would land in the attractor $A'=15\rightarrow 9\rightarrow 6\rightarrow 15$. In this case $A=A'$. From combinatorial arguments alone then:

- In an organism which has a very large number of attractors, A' will almost never be the same as A . This makes the organism very adaptive (because almost any stimulus will cause it to leave its present attractor), but not very robust (for the same reason). Even the slightest amount of thermal noise will cause the organism to jump into a different attractor. The organism has low robustness.

- In an organism which has a very small number of attractors, A' will almost always be the same as A . This makes the organism very robust (because almost no stimulus will cause it to leave its present attractor), but not very adaptive (for the same reason). Even the greatest amount of environmental stimulus will not cause the organism to jump into a different attractor. The organism has low adaptiveness.

An alternate interpretation of attractor counts comes to us from the world of neural networks. If the organism is viewed as a set of interconnected neurons, then the state space of that organism can be interpreted as a mental space. Following this analogy, the dynamics of the state space are mental processes and those individual processes each lead to an attractor. The formalism of attractors has been used to describe memories in the brain [22,23]. The neural network cycles infinitely through the states of an attractor, until perturbed through stimulus by some outside force, introducing a change in dynamics. A mental process that has this property of persisting over time until disturbed by an outside force is a memory or a thought. The number of distinct memories or thoughts a network can represent reflects the quantity of information that the network can process or store. Attractors and their potential for information storage has received

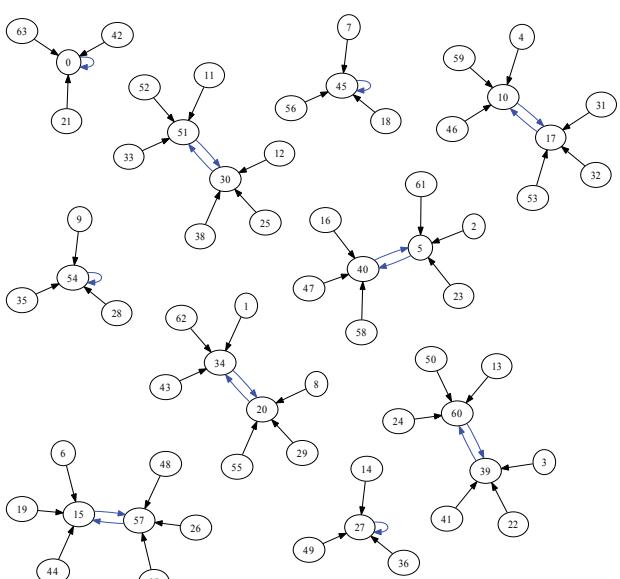


Figure 9: Dynamics graph of the organism of size 6, in decimal.

much attention especially in the areas of neural networks and neural science. The number and length of attractors have been studied extensively [8,24-28]. In ref. [29], simple dynamic systems are studied for their large number of attractors and the ease that they can be manipulated to perform neural processes.

It is clear then that to attain a reasonable trade-off between being robust and being adaptive, the organism requires an “intermediate” number of attractors. Having too many attractors implies a loss of robustness, and having too few means a loss of adaptiveness. What is too many and too few? We know that for an organism of size N, the number of states in the state space is 2^N , so this is an upper bound on the number of attractors. Since every state leads to another state and the total number of states is finite, there has to be at least one attractor in the dynamics graph; thus 1 is a lower bound on the number of attractors.

Given the above observations, the main result of this paper, that whenever the cyclic organism reaches a size that is a proper power of 2 (e.g., 2, 4, 8, 16, 32, 64, ...) the number of attractors in its dynamics graph is exactly 1, implies that every time an organism doubles in size, it necessarily passes through a size at which it has very low adaptiveness. In the next section of this paper, we show through computational experiments, that one way to remedy this low adaptiveness is for the organism to depart from homogeneity at the cellular level. These observations may reflect underlying evolutionary forces driving morphogenesis and cellular differentiation.

Avoiding low adaptiveness by departing from homogeneity

We explore whether low adaptivity can be avoided by introducing cellular differentiation. In the homogeneous network every cell applies the same update rule. We define a minimally heterogeneous organism to have cellular differentiation at a single cell meaning that only at this one cell is a different update rule applied. Experimentally, we simulate the state space dynamics of every possible minimally heterogeneous organism at sizes 2, 4, 8, and 16. For larger sizes 32, 64, 128, and 256 where complete simulation of the state space dynamics is computationally intractable, we sample the state space. To sample the state space, we randomly select 1000 initial states and successively compute their successor states until we find an attractor and record each attractor discovered. The number of attractors discovered by sampling provides a lower bound for the total number of attractors.

A cell's update rule takes as input the Boolean values of its neighbours. We refer to the neighbor to left v_{i-1} and the neighbor to the right as v_{i+1} where i refers to the index of cell applying the rule. For example, the state in Figure 1 can be indexed according to Figure 3. If we let $i=2$, then $v_{i-1}=v_1$ and $v_{i+1}=v_3$. Then we can determine that the value of the neighboring cells is 0 at v_{i-1} and 1 at v_{i+1} . Since there are 2 neighbors and each can have one of two possible values 0 or 1, then there are 2^2 or 4 possible input combinations of neighbor values. A cell's value must be either 0 or 1 as a result of each of the possible inputs from its neighbors. Since there are 2 choices for each of the 4 possible input combinations, there are 24 or 16 possible rules. In Table 1 we define these rules using Boolean logic. The rules are ordered in ascending order according to the Boolean value of the input values for which the rule sets the cell's value to have the value 1.

In Table 2, we see the results of the simulation. At increasing sizes which are proper powers of 2, minimally heterogeneous organisms where the single cell applies one of following rules {1, 3, 5, 7, 8, 10,

Rule 0=	0
Rule 1=	$\neg v_{i-1} \wedge \neg v_{i+1}$
Rule 2=	$\neg v_{i-1} \wedge v_{i+1}$
Rule 3=	$\neg v_{i-1}$
Rule 4=	$v_{i-1} \wedge \neg v_{i+1}$
Rule 5=	$\neg v_{i+1}$
Rule 6=	$(v_{i-1} \wedge \neg v_{i+1}) \vee (\neg v_{i-1} \wedge \neg v_{i+1})$
Rule 7=	$\neg(v_{i-1} \wedge v_{i+1})$
Rule 8=	$v_{i-1} \wedge v_{i+1}$
Rule 9=	$(v_{i-1} \wedge \neg v_{i+1}) \wedge (\neg v_{i-1} \wedge \neg v_{i+1})$
Rule 10=	v_{i+1}
Rule 11=	$v_{i+1} \vee (\neg v_{i-1} \wedge \neg v_{i+1})$
Rule 12=	v_{i-1}
Rule 13=	$v_{i-1} \wedge (\neg v_{i-1} \wedge \neg v_{i+1})$
Rule 14=	$v_{i-1} \vee v_{i+1}$
Rule 15=	1

Rule numbers expressed as Boolean logic with 2 inputs: The value of the left neighbor v_{i-1} , and the value of the right neighbor: v_{i+1}

Table 1: Table of update rules.

Size	Rule1	Rule 3	Rule 5	Rule 7	Rule 8	Rule 10	Rule 12	Rule 14
2	1	1	1	1	1	1	1	1
4	1	1	1	1	3	3	3	3
8	4	4	4	4	4	4	4	4
16	10	10	10	10	22	22	22	22
32	≥ 530	≥ 793	≥ 809	≥ 505	≥ 527	≥ 804	≥ 806	≥ 505
64	≥ 998	≥ 1000	≥ 1000	≥ 1000	≥ 998	≥ 1000	≥ 1000	≥ 998
128	≥ 1000							
256	≥ 1000							

Table 2: Table of minimally heterogeneous organism attractor count as size increases by powers of 2.

12 and 14} have an increase in number of attractors. Note, in Table 2 each column corresponds to one of these minimally heterogeneous organisms. For networks of sizes over 24, exhaustively enumerating the entire dynamics graph is computationally intractable [30-33]. The break in Table 2 between N=16 and N=32 corresponds to a shift in computational strategy, from finding the exact number of attractors by fully exploring the dynamics graph, to finding a lower bound of number of attractors by sampling the dynamics graph starting at 1000 random initial start states. The minimally heterogeneous organisms where the single cell applies one of the following rules {0, 2, 4, 9, 11, 13 and 15} are not shown in Table 2. For these minimally heterogeneous organisms cellular differentiation did not circumvent the collapse to a single attractor.

In Figures 10-13 we show the state space dynamics of minimally heterogeneous organisms at sizes 2, 4, 8 and 16 where the deviant cell applies rule 1 (instead of XOR) to determine its value. For comparison the state space dynamics of the homogeneous organism (in which all cells apply the XOR update rule) of corresponding sizes, can be seen in Figures 14-17. We generate Figures 10-13 as described earlier using experimentally simulated dynamics rendered by Graphviz.

Low Adaptivity and Organisms of Size 2^i

Our objective is to prove two theorems.

Theorem 1, if the number of cells in an organism is a proper power of 2, then the organism has exactly one attractor, which has length 1 and consists of the state where all cells have a value of 0.

Theorem 2, if regardless of initial state X, the organism always ends

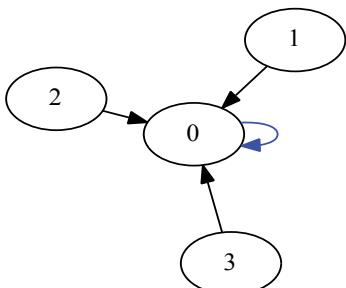


Figure 10: XOR organism size 2.

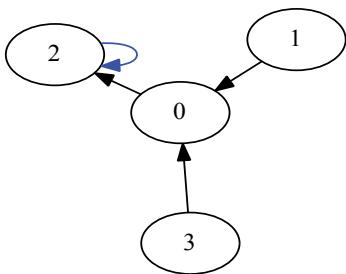


Figure 11: Minimally heterogeneous size 2 XOR+Rule 1.

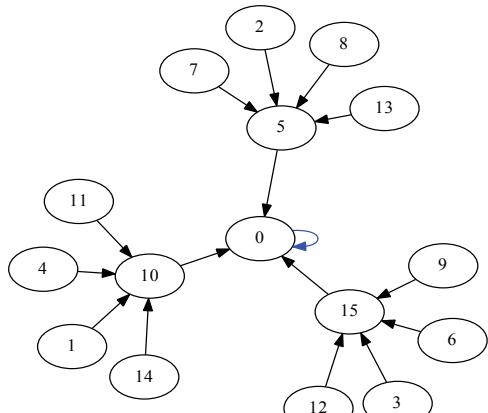


Figure 12: XOR organism size 4.

up in the same attractor, then the number of cells in the organism is a power of 2.

It will take some work to get the proofs of the above statements; they appear in finality on pp. 19. We begin with some definitions in Section 3.1 below. Then, in Section 3.2, we develop a theory of decompositions, which are useful to carry out arguments about the dynamics of the organism's state over time. In Section 3.3, this is used to prove results about infinite organisms (in which the cells exist on an infinite line). In Section 3.4, these results are wrapped into or "mapped down" to the dynamics of finite cyclic organisms.

Definitions

Structure: We consider organisms whose cellular structure may be modelled as an undirected cyclic graph $C=(V, E)$ of size n , whose vertices are considered "cells" and are enumerated $V=\{v_0, \dots, v_{n-1}\}$. Each

cell v_i in V is connected in cyclic order to two neighbors, so that $E=\{(v_i, v_{i+1} \pmod n) \mid i=0, \dots, n-1\}$. Microscopic cellular behavior within an organism is modeled by fixing a function $f: V \rightarrow F$ that assigns to each cell $v \in V$, a function $f(v)$ from $F = \{g : \{0, 1\} \times \{0, 1\} \rightarrow \{0, 1\}\}$, the set of all binary Boolean functions; note that $|F|=2^{2^2}=16$. The action of f at a vertex v_i can be thought of as a truth table mapping v_i 's left and right neighbors' current state, to v_i 's state at the next time step.

In Table 3 since each of the bits b_0, b_1, b_2, b_3 must be either 0 or 1, in what follows, we will frequently use the 4-bit binary string b_0, b_1, b_2, b_3 to name the function f . Together, the pair (C, f) define the microscopic structure of the organism. An organism is said to be homogeneous if $|Im(f)|=1$; otherwise it is said to be heterogeneous.

State: Since at each instant, a cell can have a value of either 0 or 1, the instantaneous state of the organism is specifiable as a function $V \rightarrow \{0, 1\}$. The state of the organism over (discrete) time may then be represented by a function $s: V \times N \rightarrow \{0, 1\}$ where $s(v_i, t)$ is the state of cell $v_i \in V$ at time t . Since cell v_i behaves (across all time) according to function $f(v_i)$, and all cells are assumed to operate synchronously, the state of the organism evolves over time according to the following law:

$$s(v_i, t+1) = f(v_i)(s(v_{i-1} \pmod n, t), s(v_{i+1} \pmod n, t))$$

For each $i=0, \dots, n-1$ and $t \geq 0$. Informally, the state of the organism's

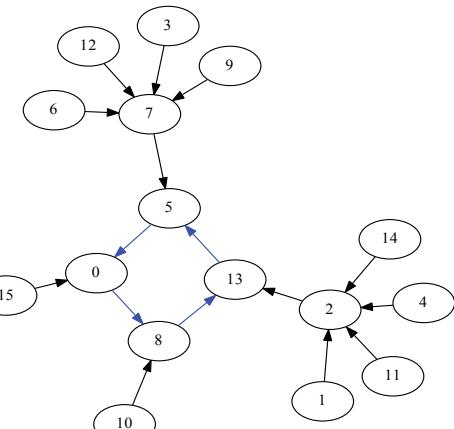


Figure 13: Minimally heterogeneous size 4 XOR+Rule 1.

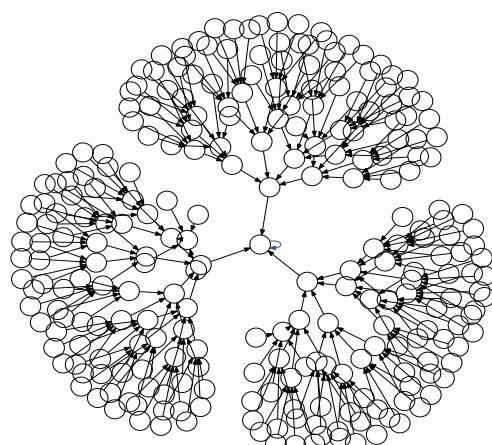


Figure 14: XOR organism size 8.

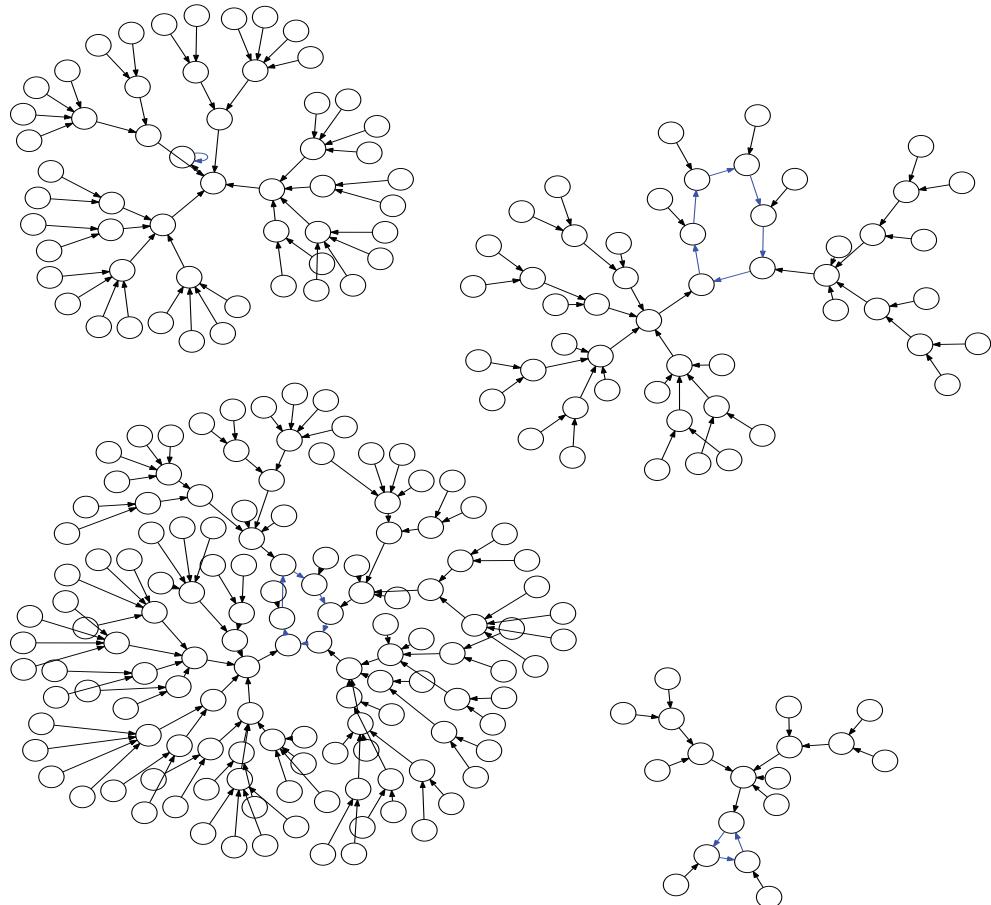


Figure 15: Minimally heterogeneous size 8 XOR+Rule 1.

constituent cells evolves according to the rule specified by Boolean function operating at that cell, together with the current state of its two adjacent cellular neighbors. We denote the subset of cells whose state is “on” (i.e., 1) at time t as $s^+(t) = \{v \in V \mid s(v, t) = 1\}$. Note that to identify the system’s state it suffices to know $s^+(t)$, since we can infer that the remaining cells are in state 0. In what follows, we will frequently identify the state of the organism at time t with the subset $s^+(t) \subset V$ [8].

Decompositions

The results presented in this section consider countably infinite populations of cells arranged in an infinite line. We will show that it is always possible to decompose the cells into independent segments, on which the successor function acts independently. One can thus compute the action of the successor function on the organism as a whole by amalgamating its action on each of the independent segments in the decomposition. This is the essential content of the final result in this section, Lemma 10 on pp. 16. Next, in Section 3.3 (pp.17), we use the decompositions to prove significant results about the dynamics of infinite linear organisms. We begin with the following definition (Table 4).

Definition 1: Let $(Z/2Z)^z$ be the set of functions from the integers Z to the two-element set $(Z/2Z) = \{0, 1\}$. Each function $x: Z \rightarrow \{0, 1\}$ in $(Z/2Z)^z$ may be represented as an indexed string where the constituent binary symbols are annotated with subscripts from the function’s

domain Z .

For example, if x is a function which maps the three integers 0, 1 and 7 all to 1 while mapping all other integers to 0, then we will write x as a subscripted string, as follows:

$$x = \bar{0}1_01_10_20_30_40_50_61_7\bar{0}.$$

Here $\bar{0}$ represents an abbreviation for the left-infinite sequence of 0s (for subscripts decreasing to $-\infty$), while $\bar{0}$ is an abbreviation that stands for the right-infinite sequence of 0s (for subscripts increasing to $+\infty$). The bijective correspondence between subscripted strings and functions is unambiguous. Abusing the notation, we denote both the function that is everywhere 0, and its associated indexed string, as \hat{S} .

In the discussion that follows, we shall frequently move back and forth between functions and their indexed string representations. We will adhere to a convention wherein functions in $(Z/2Z)^z$ shall be denoted by lowercase letters (e.g., x, y, z) while their bi-infinite binary string representations shall be denoted with the corresponding uppercase letters (e.g., X, Y, Z).

The next definition captures the fact that each individual responds uniformly to the presence/absence of local belief diversity, since XOR (and its negation) are the only two non-constant symmetric Boolean-valued functions on two inputs.

Definition 2: Let \oplus be a binary operator on $(Z/2Z)^z$ defined as

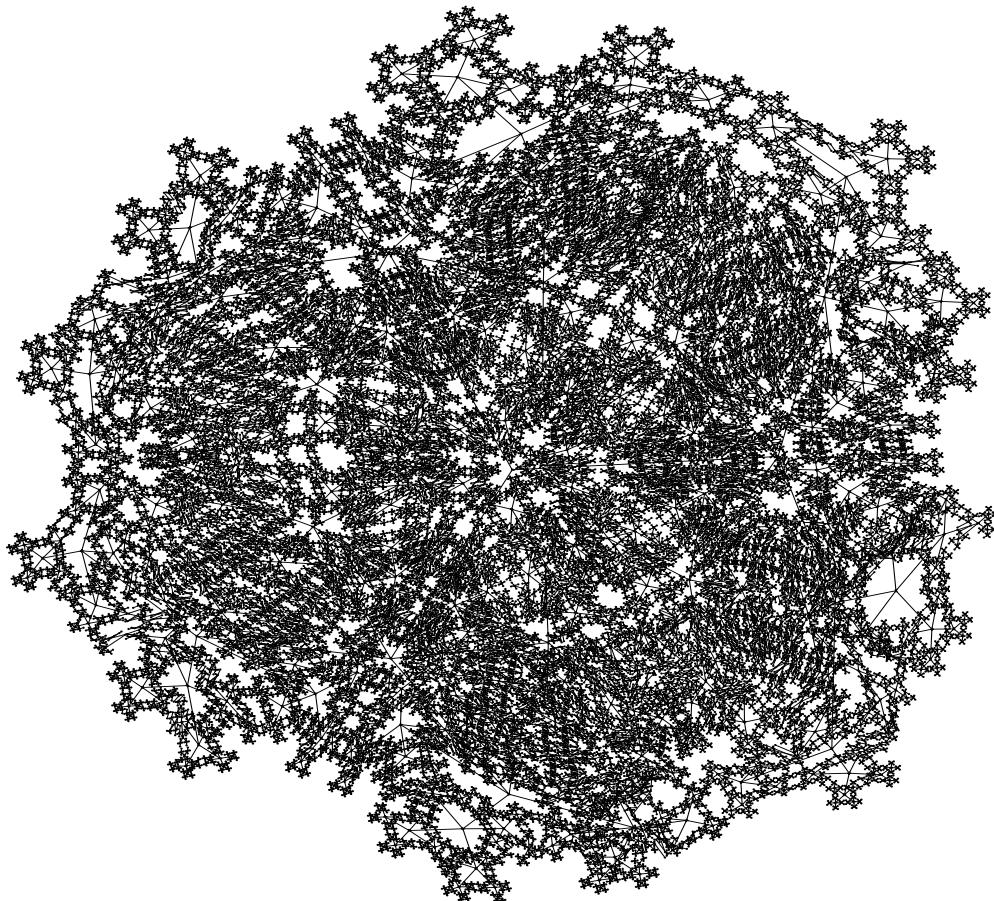


Figure 16: XOR organism size 16.

follows. Given two functions $x, y \in (Z/2Z)^Z$, the value of $(x \oplus y):(Z/2Z)^Z$ at integer i in Z , is defined in terms of the exclusive-or \oplus operation $(x \oplus y)(i)=x(i) \oplus y(i)$,

Where, the truth table for the \oplus operation is enumerated in Table 4.

We use Table 4 to define the successor function \hat{S} , which describes the state of the entire system at each successive time step by applying the XOR update rule synchronously at each constituent cell. For example, if $X = 01_0 1_1 0_2 0_3 0_4 0_5 0_6 1_7 \bar{0}$. then $\hat{S} X = \bar{0} 1_{-1} 0_0 1_2 0_3 0_4 0_5 0_6 1_7 1_8 \bar{0}$.

We intend to quantify the properties of \hat{S} using decompositions (see Definitions 11 and 12), but first we must introduce some notations and preliminary results; this is the objective of Definitions 3-10 and Lemmas 3-8 (on pp. 12-15), which follow.

Definition 3: Let $\hat{S} : (Z/2Z)^Z \rightarrow (Z/2Z)^Z$ be a unary operator defined such that for each function x in $(Z/2Z)^Z$, the value of $\hat{S} x : Z \rightarrow Z/2Z$ at i in Z is taken to be $\hat{S} x(i)=x(i-1) \oplus x(i+1)$.

As is customary notation for successive powers of operators, we define \hat{S}^0 to be the identity map on $(Z/2Z)^Z$ and then inductively put $\hat{S}^j = \hat{S} \circ \hat{S}^{j-1}$, for each $j > 0$. The successor function \hat{S} and \oplus enjoy a close relationship, as Lemmas 3 and 5 make evident.

Lemma 3: For all x, y in $(Z/2Z)^Z$, and all $i \in Z$, $\hat{S} (x \oplus y)(i)=\hat{S} x(i) \oplus \hat{S} y(i)$.

Proof: By Definitions 2 and 3, we know that

$$\hat{S} x(i) \oplus \hat{S} y(i)=(x(i-1) \oplus x(i+1)) \oplus (y(i-1) \oplus y(i+1))$$

$$\hat{S} (x \oplus y)(i)=(x(i-1) \oplus y(i-1)) \oplus (x(i+1) \oplus y(i+1)).$$

The right hand sides of the above equations are equal by the associativity and communicativity of the exclusive-or operation \oplus over $Z/2Z$, and thus so are the left-hand sides. The Lemma follows. The previous Lemma suggests that the associative and communicative properties of \oplus could be leveraged if a function x can be decomposed into a sum (w.r.t \oplus), since the action of \hat{S} to then be distributed over summands. This idea shall be brought to fruition in Lemma 10.

Definition 4: Two functions $x, y \in (Z/2Z)^Z$ are said to be shift-related, denoted $x \approx y$, if there exists a shift $t \in Z$ such that $x(i)=y(i+t)$ for all i in Z .

For example, if $X = \bar{0} 1_0 0_1 0_2 1_3 0_4 0_5 0_6 1_7 \bar{0}$ and $Y = \bar{0} 1_0 0_1 1_2 1_3 0_4 1_5 0_6 1_7 \bar{0}$, then $x(i)=y(i+t)$ where $t=1$ (for all i in Z), and hence x and y are said to be shift-related. On the other hand, if $Z = \bar{0} 1_0 0_1 1_2 1_3 0_4 1_5 0_6 1_7 \bar{0}$, then z is not shift-related to x , since there is no integer t such that $z(i)=x(i+t)$ for all i in Z . From this it follows that z is also not shift-related to y which is a specific application of the next Lemma.

Lemma 4: The shift-relation \approx is an equivalence relation on $(Z/2Z)^Z$.

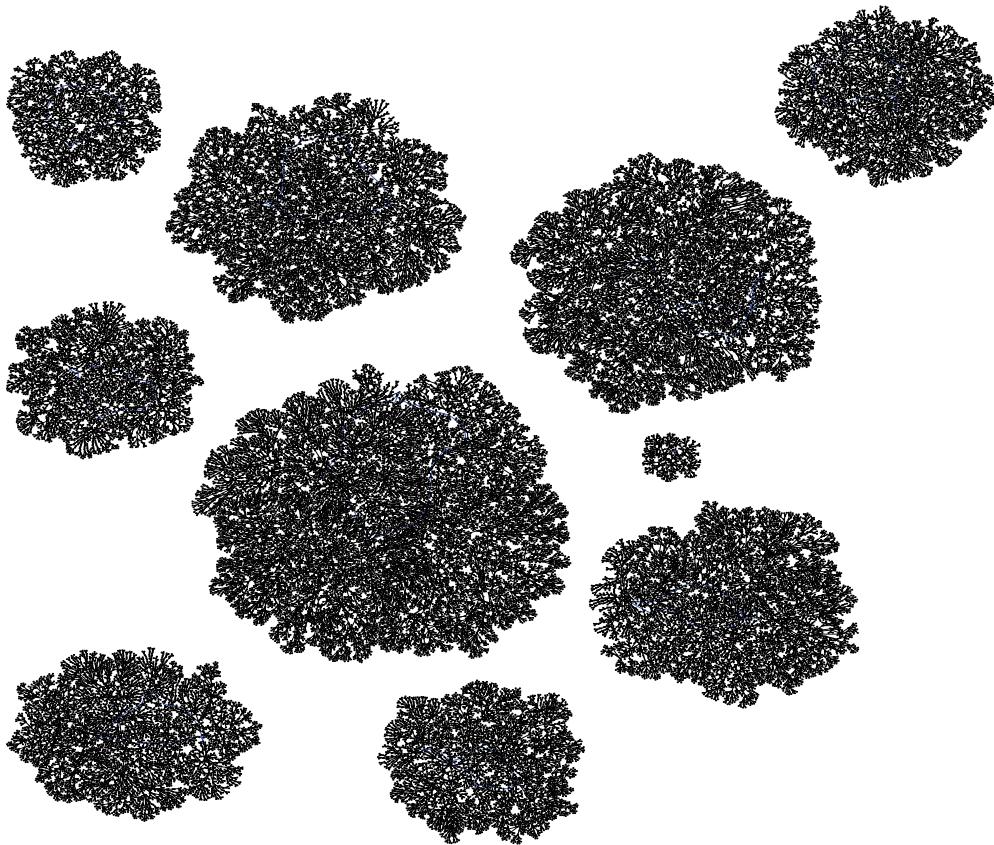


Figure 17: Minimally heterogeneous size 16 XOR+Rule 1.

$s(v_{i,t})$	$s(v_p,t)$	$s(v_{i+1},t)$	$s(v_p,t+1)$
0	*	0	b_0
0	*	1	b_1
1	*	0	b_2
1	*	1	b_3

Table 3: XOR truth table with inputs at time t and resulting output at time $t + 1$.

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Table 4: Table of XOR truth table.

Proof: Consider functions $x, y, z \in (Z/2Z)^Z$. Reflexivity is obvious since $x \approx x$ by taking $t=0$ in definition 4. If $x \approx y$ by shift t , then $y \approx x$ by shift $-t$, implying symmetry. Finally, transitivity holds since if $x \approx y$ by shift t_1 , and $y \approx z$ by shift t_2 , then $x \approx z$ by shift t_1+t_2 .

Informally, if two functions are shift equivalent then the results of their successors are also shift equivalent. This is clear from the example strings X and Y in Definition 4: $\hat{S} X = \bar{0}1_{-1}0_01_{-1}0_10_{-1}1_20_{-1}1_30_{-1}1_40_{-1}1_50_{-1}1_60_{-1}1_70_{-1}1_80$ and

$Y = \bar{0}1_00_{-1}1_10_{-1}1_20_{-1}1_30_{-1}1_40_{-1}1_50_{-1}1_60_{-1}1_70_{-1}1_80$ where $\hat{S} x_i = \hat{S} y_{i+1}$ therefore $\hat{S} X \approx \hat{S} Y$. The next Lemma proves the general case.

Lemma 5: If $x, y \in (Z/2Z)^Z$ and $x \approx y$, then $\hat{S} x \approx \hat{S} y$.

Proof: If $x \approx y$, then by Definition 4, there exists $t \in Z$ such that $x(i) = y(i+t)$ for all i in Z . By definition 3, we know that $\hat{S} x(i) = x(i-1) \oplus x(i+1)$ and $\hat{S} y(i) = y(i-1) \oplus y(i+1)$. Appealing again to definition 4, we see that $\hat{S} x(i) = \hat{S} y(i+t)$, from which it follows that $\hat{S} x \approx \hat{S} y$.

We shall use an ordinary, non-indexed string representation for \approx -equivalence classes of functions in $(Z/2Z)^Z$. Towards this, we introduce the next definition.

Definition 5: For each function $x \in (Z/2Z)^Z$, let $\bar{X} = \dots x(-2).x(-1).x(0).x(1).x(2)\dots$ be the associated bi-infinite binary (ordinary, non-indexed) string. While definition 1 reflects the fact that every function x in $(Z/2Z)^Z$ corresponds unambiguously to an indexed string, the next Definition and Lemma capture the fact that this correspondence is not 1-1 in the case of the ordinary non-indexed strings presented in Definition 5.

Definition 6: Associated with every bi-infinite binary (ordinary, non-indexed) string \bar{X} is a countably infinite 1-parameter family of functions $[\bar{X}] \subset (Z/2Z)^Z$, wherein $\bar{X} = \{x_t : \mathbb{Z} \rightarrow 2\mathbb{Z} | t \in \mathbb{Z}\}$, where $x_t(i) = x(t+i)$ for all i in Z .

Lemma 6: For any bi-infinite binary (ordinary, non-indexed) string \bar{X} , the set $[\bar{X}] \subset (Z/2Z)^Z$ is closed under shift equivalence; that is, (i) if $x_a, x_b \in [\bar{X}]$ then $x_a \approx x_b$, and (ii) if $x_a \in [\bar{X}]$ and $x_a \approx y$ then $y \in [\bar{X}]$.

Proof: To see (i) consider two functions $x_a, x_b \in [\bar{X}]$. By Definition 6 we know that $x_a(i)=x(a+i)=x(b+i+(a-b))=x_b(i+(a-b))$, and so it follows that $x_a \approx x_b$ by considering a shift of $t=a-b$ in Definition 4. To see (ii) suppose $x_a \approx y$ for some $x_a \in [\bar{X}]$ and some $y \in (Z/2Z)^Z$. Then by Definition 4, there exists t such that $x_a(i)=y(i+t)$ for all $i \in Z$, and thus $y \equiv x_a + t$, implying that $y \in [\bar{X}]$ by Definition 6. The set F of all binary valued functions having finite support (i.e., which take value 1 at only finitely many integers) shall turn out to be of special interest.

Definition 7: Let $F \subset (Z/2Z)^Z$ be the set of binary-valued functions on Z having finite support; that is, $x \in F$ if $x(i)=0$ for all but finitely many $i \in Z$. For example, $b(x)=\begin{cases} \min\{i|x(i)=1\} & x \neq \bar{0} \\ 0 & x = \bar{0} \end{cases}$ corresponds to a function x that lies in F , since X contains only three 1s. On the other hand, a function x' which sends all even integers to 1 and all odd integers to 0, lies in $(Z/2Z)^Z \setminus F$. For functions of finite support, it will frequently be useful to refer to the least and greatest integer which map to 1. Towards this, we introduce the next definition.

Definition 8 Let $b, e: F \rightarrow Z$ be defined as follows:

$$b(x)=\begin{cases} \min\{i|x(i)=1\} & x \neq \bar{0} \\ 0 & x = \bar{0} \end{cases}$$

$$e(x)=\begin{cases} \max\{i|x(i)=1\} & x \neq \bar{0} \\ -1 & x = \bar{0} \end{cases}$$

For each function x in F , the length of $|x|=e(x)-b(x)+1$ is taken to be the number of bits in the largest essentially non-zero subsegment of X . Continuing the previous example $X=\bar{0}_0, 0_0, 1_0, 0_4, 0_0, 1_0, \bar{0}$, we note that $b(x)=0$ and $e(x)=7$ and $|x|=8$. This suggests that we can “shell” the set F by partitioning it into disjoint subsets and assigning each function $x \in F$ to a specific subset on the basis of $|x|$. The subsequent Definition and Lemma achieves such a shelling.

Definition 9: Let B_0 denote the singleton set consisting of the empty string, and for each integer $n > 0$ let B_n denote the set of binary strings beginning and ending in 1 and having of length n . Put $B=\bigcup_{n=0}^{\infty} B_n$.

Note that the sets B_n consist of finite ordinary non-indexed binary strings of length n . The next Lemma places the set of \approx -equivalence classes of binary functions with finite support into 1-1 correspondence with the set of finite ordinary non-indexed binary strings.

Lemma 7: The quotient F/\approx is in natural bijective correspondence with B .

Proof: We map $\bar{0} \in F$ to the empty string in $B_0 \subset B$ having length 0. It remains to demonstrate a bijection ϕ between $F \setminus \{\bar{0}\}$ and the set of binary strings of finite positive length which begin and end with 1. Given $x \in F$, $x \neq \bar{0}$, we take $\phi(x) \in B_{e(x)-b(x)+1} \subset B$ to be the string $\phi(x)=x(b(x)) \cdot x(b(x)+1) \cdots x(e(x)-1) \cdot x(e(x))$.

Clearly if $x \neq x'$ as functions, then $\phi(x)$ and $\phi(x')$ are distinct members of B . Moreover, if $y \in F$ and $y \approx x$ then $\phi(x)=\phi(y)$.

In the reverse direction, given a binary string $X \in B_n \subset B$ of positive length $|X|=n>0$, we write X as a sequence of binary bits having finite positive length

$$X=X_0 X_1 \cdots X_i \cdots X_{n-2} X_{n-1}$$

and consider the function $x \in F$ given by

$$x(i)=\begin{cases} X_i & 0 \leq i < |X| \\ 0 & \text{otherwise.} \end{cases}$$

Since X has positive length, $X \neq Y$, and $\phi^{-1}(X)$ is taken to be the \approx -equivalence class of x . Clearly if $Y \in B$ and $X \neq Y$, then $\phi^{-1}(X) \cap \phi^{-1}(Y)=\emptyset$.

Definition 10: By Lemma 5, the operator \hat{S} factors through the \approx relation, and thus the action of \hat{S} on $F \subset (Z/2Z)^Z$ presented in Definition 3 induces an operator (which we shall denote as S) on the quotient set F/\approx . Since F/\approx was shown to correspond to the set B in Lemma 7, we arrive at an induced unary operator $S: B \rightarrow B$. The function S is thus a self-map of B , which is a set of strings that contains all finite strings beginning and ending with 1 (as well as the empty string).

With the preceding definitions in hand, we return to the evolution of the dynamical systems over time under the action of the successor function \hat{S} . The next Lemma shows that for functions with finite support, the function’s support interval expands outwards under the action of \hat{S} ; in particular $|\hat{S}x|=|x|+2$.

Lemma 8: Let $x \in F \setminus \{\bar{0}\}$. Then

$$b(\hat{S}x)=b(x)-1$$

$$e(\hat{S}x)=e(x)+1$$

Proof: Since $x(b(x))=1$ and $x(b(x)-2)=0$, by Definition 3, $\hat{S}x(b(x)-1)=1$ and since for all $i < b(x)-1$, $x(i-1)=x(i+1)=0$, it follows that $b(\hat{S}x)=b(x)-1$. Analogously, since $x(e(x))=1$ and $x(e(x)+2)=0$, by Definition 3, $\hat{S}x(e(x)+1)=1$ and since for all $i > e(x)+1$, $x(i-1)=x(i+1)=0$, it follows that $e(\hat{S}x)=e(x)+1$.

Given a function $x \in F$, we can decompose its string representation X into c disjoint component strings, where each of the components has 0 or as a prefix and suffix. Such decomposition shall be useful to factor the action of \hat{S}^r on x into a set of independent action on each of the c components. The Definition below renders the decomposition formally.

Definition 11: Given a function $x \in F$, integers $r \geq 0$ and $c \geq 1$. Choose $r_j > 0$ and $g_j \geq 0$ (for $j=1, \dots, c$), and let P be a partition of the set $\{b(x)-r_1, \dots, e(x)+rc\} \subseteq Z$

$$P=\{(b_1, b_1+1, \dots, e_1), (b_2, b_2+1, \dots, e_2), \dots, (bc, bc+1, \dots, ec)\}$$

into c contiguous integer subsequences, in a manner which additionally satisfies:

$$1. b_1=b(x)-r_1; ec=e(x)+r_c$$

2. For $j=1, \dots, c$:

$$\bullet e_j \geq b_{j+1} - 2r_j;$$

$$\bullet x(b_{j+1} - r_j) = x(e_j - r_j) = 1;$$

$$\bullet \text{For all } i \text{ satisfying } b_j \leq i < b_{j+1} - r_j \text{ or } e_j - r_j < i \leq e_j, x(i)=0.$$

$$3. b_{j+1} = e_j + g_j + 1, \text{ for all } j=1, \dots, c-1.$$

Then, for $j=1, \dots, c$, define

$$\tilde{W}^j = x(b_j + r_j) \cdot x(b_j + r_j + 1) \cdots x(e_j - r_j - 1) \cdot x(e_j - r_j)$$

and take $W^j = 0^{r_j} \cdot \tilde{W}^j \cdot 0^{r_j}$. Note that $\tilde{W}^j \in B_{e_j - b_j - 2r_j + 1}$. Take $r = \min_{j=1}^c \{r_j\}$

We refer to the tuple $(b_1, W^1, g_1, W^2, g_2, \dots, g_{c-1}, W^c)$ as an (r, c) -decomposition of x .

To compute, for example, a $(1, 3)$ decomposition of our ongoing example $X=\bar{0}_0, 0_0, 1_0, 0_4, 0_0, 1_0, \bar{0}$, we need to provide a size 3 partition

of the sequence $(-1, 0, 1, \dots, 7, 8)$ into contiguous integer sequences. If we take $P=\{(-1, 0, 1), (2, 3, 4), (6, 7, 8)\}$, then conditions 1-3 of Definition 11 can be verified directly, noting that $b_1=-1, e_1=1, g_1=0, b_2=2, e_2=4, g_2=1, b_3=6, e_3=8$; note that g_2 maintains the gap between the sub-segments of indices $(2, 3, 4)$ and $(6, 7, 8)$. It follows that $X = \bar{0}1_00_10_21_30_40_50_61_7\bar{0}$ is a $(1, 3)$ decomposition of X .

The structure of (r, c) -decompositions factor through the equivalence relation \approx . For example, referring to the strings $X = \bar{0}1_00_10_21_30_40_50_61_7\bar{0}$ and $Y = \bar{0}1_00_20_31_00_50_60_71\bar{0}$ introduced subsequent to Definition 4, we see that $(\bar{0}1_0, 0, \bar{0}1_0, 1, \bar{0}1_0)$ is a $(1, 3)$ -decomposition of X , while $(\bar{0}1_0, 0, \bar{0}1_0, 1, \bar{0}1_0)$ is a $(1, 3)$ -decomposition of Y . The fact that Y is a $t=1$ shift of X is reflected in the fact that $b_1=0$ decomposition of Y , a value that is 1 greater than its value in the decomposition of Y . This observation is stated formally below:

Lemma 9: Let $X \in B$, and $x, x' \in [X]$. Let t be an integer for which $x'(i+t)=x(i)$ for all $i \in Z$. If $(b_1, W^1, g_1, W^2, g_2, \dots, g_{c-1}, W^c)$ is an (r, c) -decomposition of x , then $(b_{1+t}, W^1, g_1, W^2, g_2, \dots, g_{c-1}, W^c)$ is an (r, c) -decomposition of x' .

The above allows us to extend the definition of (r, c) -decompositions to \approx -equivalence classes of functions.

Definition 12: For each $X \in B$, take $x \in [X]$ and let $(b_1, W^1, g_1, W^2, g_2, \dots, g_{c-1}, W^c)$ is an (r, c) -decomposition of x . We refer to the tuple $(W^1, g_1, W^2, g_2, \dots, g_{c-1}, W^c)$ as an (r, c) -decomposition of the \approx -equivalence class $[X]$.

Continuing our example, $(\bar{0}1_0, 0, \bar{0}1_0, 1, \bar{0}1_0)$ is a $(1, 3)$ -decomposition of $[X]=[Y]$. We note by definition each (r, c) -decomposition $(b_1, W^1, g_1, W^2, g_2, \dots, g_{c-1}, W^c)$ of $x \in F$ gives rise to a set of functions $x_j: (Z/2Z)^Z$ (for $j=1, \dots, c$) where,

$$x_j(i) = \begin{cases} x(i) & b_j \leq i \leq e_j \\ 0 & \text{otherwise.} \end{cases}$$

Satisfying the relation $x=x_1 \oplus x_2 \oplus \dots \oplus x_c$. This identity quantifies the manner in which we decompose x into a \oplus sum, each summand of which may be seen as being acted upon independently by \hat{S} .

Lemma 10: Given $X \in B$, let $(W^1, g_1, W^2, g_2, \dots, g_{c-1}, W^c)$ be an (r, c) -decomposition of $[X]$, for fixed integers $r \geq 0$ and $c \geq 1$. Then for each integer $0 < t \leq r$, the tuple

$(0^{r-t} \cdot S'W^1 \cdot 0^{r-t}, g_1, 0^{r-t} \cdot S'W^2 \cdot 0^{r-t}, g_2, \dots, g_{c-1}, 0^{r-t} \cdot S'W^c \cdot 0^{r-t})$ is an $(r-t, c)$ -decomposition of $[SX]$.

Proof: Fix $x \in [X]$ and let $(b_1, W^1, g_1, W^2, g_2, \dots, g_{c-1}, W^c)$ be an (r, c) -decomposition of x . For $j=1, \dots, c$ by Definition 11, we know that $x(b_j+r)=x(e_j-r)=1$, and $x(i)=0$ whenever $b_j \leq i < b_{j+1}-r$ or

$e_j-r < i \leq e_j$. Moreover,

$$W^j = \underbrace{0^r \cdot x(b_j+r) \cdot x(b_j+r+1) \cdots x(e_j-r-1) x(e_j-r) \cdot 0^r}_{W^j}$$

Viewing W^j as a substring, by Lemma 8 we see that $b(\hat{S}^t x)=b(x)-t$, and $e(\hat{S}^t x)=e(x)+t$. Since (by assumption) $t \leq r$, it follows that

$$(b_1-t, 0^{r-t} S'W^1 \cdot 0^{r-t}, g_1, 0^{r-t} \cdot S'W^2 \cdot 0^{r-t}, g_2, \dots, g_{c-1}, 0^{r-t} \cdot S'W^c \cdot 0^{r-t})$$

is an $(r-t, c)$ -decomposition of $\hat{S}^t x$. The conclusion of the Lemma follows by taking the above $(r-t, c)$ -decomposition of $S^t x$ and considering it as the basis of an (r, c) decomposition of the \approx -equivalence class $[SX]$, as per Definition 12.

The previous Lemma demonstrates that (r, c) -decompositions are a parsimonious way of describing the action of \hat{S} on $x \in F$ as an aggregation of separate independent actions of S smaller sub-segments of X . This will be useful repeatedly in the arguments that follow.

The infinite case

The main theorem of this section is the formal proof of the assertion that if you start with a state that consists of just two 1s separated by some number of zeros, and then simulate forward, you will again at some point enter a state that has just two 1s separated by (an even larger) number of zeros. More precisely, if you start with two 1s separated by 2^i-1 zeros, then after 2^i-1 steps, you will arrive at a state where you have two 1s separated by $2^{i+1}-1$ zeros. Next, in Section 3.4 (pp. 19), we use this theorem to prove important results about the dynamics of finite cyclic organisms.

Formally stated:

$$\text{Theorem 11: } \forall i \geq 2 \quad S^{2^{i-1}}(10^{2^i-1}) = 10^{2^{i-1}}$$

Recalling $S: B \rightarrow B$ from Definition 10, we introduce the following named assertion ϕ :

Definition 13: For fixed integer $i \geq 2$, put

$$\phi_i : S^{2^{i-1}}(10^{2^i-1}) = (10)^{2^{i-1}}$$

The main result proved in this section (Proposition 18) is that for all $i \geq 2$, assertion ϕ_i is true. This proof shall proceed by induction, for which the next Lemma provides the base case.

Lemma 12: ϕ_2 is true.

Proof: It suffices to show $S^1(10^3) = (10^3)1$. Noting that $(\overline{00100})$ is a $(1, 1)$ -decomposition of $[10^3]1$, by Lemma 10 we know $(S^1(10^3))1$ is a $(0, 1)$ -decomposition of $[S^1(10^3)]1$, and since $S^1(10^3)1=1010101=(10^3)1$, the assertion is proved.

Lemma 13: $S^2(1)=10^31$.

Proof: Noting that $(\overline{00100})$ is a $(2, 1)$ -decomposition of $[1]$, by Lemma 10 we know $(S^2(00100))$ is a $(0, 1)$ -decomposition of $[S^2(1)]$, and since $S^2(00100)=10001=10^31$, the assertion is proved.

Lemma 14: For all $k \geq 1$, $S^1((10)^{k-1}1)=10^{2k-1}1$.

Proof: Since $(\overline{0}(10)^{k-1}10)$ is a $(1, 1)$ -decomposition of $[(10)^{k-1}1]$, by Lemma 10 we know $(0 \cdot S^1((10)^{k-1}1) \cdot 0)$ is a $(0, 1)$ -decomposition of $[S^1((10)^{k-1}1)]$, and since $S^1((10)^{k-1}1)=10^{2k-1}1$, the assertion is proved.

Lemma 15: If $\forall i > j \geq 2$, ϕ_j is true, then $S^{2^{i-1}}(10^{2^i-1}) = \frac{(10^{2^i-1})}{S^{2^{i-1}}(10^{2^{i-1}-1}) \cdot S^{2^{i-1}}(10^{2^{i-1}-1})}$

Proof: First we write $S^{2^{i-1}-1}(10^{2^i-1}) = S^1(S^{2^{i-1}-1}(10^{2^{i-1}-1}))$. Now, by the inductive hypothesis:

$$S^{2^{i-1}-1}(10^{2^i-1}) = (10)^{2^{i-1}}$$

By appealing to Lemma 14 we $S^1((10)^{2^i-1}) = 10^{2^{i+1}-1}$, which completes the proof.

Lemma 16: If $\forall i < x$, ϕ_i is true, then $0 < k < x$ implies $S^{2^k-3}(10^3) = (10)^{2^k-1}$.

Proof: We begin by noting that

$$2^k - 3 = (2^{k-1} - 1) + (2^{k-1} - 2) = (2^{k-1} - 1) + \sum_{j=1}^{k-2} 2^j$$

Thus,

$$S^{2^k-3}(10^3) = (10^3)^{2^{k-1}-1} \circ (10^3)^{2^{k-2}-1} \circ (10^3)^{2^{k-3}-1} \cdots (10^3)^{2^2-1} \circ (10^3)^{2^1-1}$$

Repeated application of Lemma 15 yields

$$\begin{aligned}s^2^1(10^{2^2-1}1) &= 10^{2^3-1}1 \\ s^2^2(10^{2^3-1}1) &= 10^{2^4-1}1 \\ s^2^3(10^{2^4-1}1) &= 10^{2^5-1}1 \\ &\dots \\ s^{2^{k-2}}(10^{2^{k-1}-1}1) &= 10^{2^k-1}1\end{aligned}$$

It remains to compute $s^{2^{k-1}-1}(10^{2^k-1}1)$. Since $k < x$, we may assume the inductive hypothesis: ϕ_k is true. From this it follows that $s^{2^{k-1}-1}(10^{2^k-1}1) = 10^{2^k-1}1$.

Lemma 17: If ϕ_i is true $\forall i < x$ then ϕ_x is true.

Proof: It suffices to show: $s^{2^{x-1}-1}(10^{2^x-1}1) = 10^{2^x-1}1$. We begin by noting that

$$10^{2^{x-1}-1}1 = 10^{2^{x-2}/2} \cdot 0^{2^{x-2}-1}1 = 10^{2^{x-1}}1 \cdot 0^{2^{x-1}-1}1$$

and thus,

$$10^{2^{x-1}-1}1 = 10^{2^{x-2}/2} \cdot 0^{2^{x-2}-1}1 = 10^{2^{x-1}}1 \cdot 0^{2^{x-1}-1}1$$

is a $(2^{x-1}-1, 2)$ -decomposition of $10^{2^{x-1}-1}1$. So, by Lemma 10

$$(S^{2^{x-1}-1}(0^{2^{x-1}-1}10^{2^{x-1}-1}), 0, S^{2^{x-1}-1}(0^{2^{x-1}-1}10^{2^{x-1}-1}))$$

is a $(0, 2)$ -decomposition of $[S^{2^{x-1}-1}(10^{2^x-1}1)]$. Using Lemma 13, $S^{2^{x-1}-1}(1)$ may be re-expressed as:

$$S^{2^{x-1}-1-2} \circ S^2(1) = S^{2^{x-1}-1-2}(10^3)$$

Appealing to Lemma 16 we determine that $m S^{2^{x-1}-1-2}(10^3) = (10)^{2^{x-1}-1}1$. Thus, the $(0, 2)$ -decomposition of $[S^{2^{x-1}-1}(10^{2^x-1}1)]$ is in fact $(\overline{(10)^{2^{x-1}-1}1}, \overline{(10)^{2^{x-1}-1}1})$.

Concatenating the two factors and the intervening zero (since $g_1=1$), we conclude that $(\overline{(10)^{2^{x-1}-1}1})$ is a $(0, 1)$ -decomposition of $[S^{2^{x-1}-1}(10^{2^x-1}1)]$. The assertion is proven.

The proof of Proposition 18 is now immediate.

Proposition 18: For all $i \geq 2$, ϕ_i is true.

Proof: The base case is given by Lemma 12, and the inductive step by Lemma 17.

We are now ready to prove Theorem 11.

Proof, directly from Lemma 15 where $i \geq 2$, applying Proposition 18 shows that ϕ_i is true for all $i \geq 2$.

Going from infinite to finite

Suppose now that instead of operating with infinite strings (functions on Z), the operation is taking place on a cycle of N cells (numbered $0, 1, \dots, N-1$), each of which could take a value of 0 or 1.

Lemma 19: If $X=0^N$ then $S(X)=0^N$

Proof: This is by definition of the XOR function. Any cycle in which all cells have the value 0 will remain unchanged over time, that is $S(0^N)=0^N$.

Lemma 20: If there is one attractor, then the attractor is 0^N .

Proof: Suppose we have one attractor. Because an initial state $X=0^N$ is possible by definition of the networks, applying Lemma 19 completes the proof.

Definition 14: A state X is said to lead to a state Y denoted as $X \rightarrow Y$ if $\exists k$ such that $S^k(X)=Y$.

We are now ready to prove Theorem 1 (stated originally on pp. 8): If the number of cells in an organism is a proper power of 2, then the organism has exactly one attractor, which has length 1, and consists of the state where all cells have a value of 0.

Proof: Suppose N is a power of 2. Consider the starting state $0^{N-1}1$. By Lemma 8, simulating forward from this start state produces a wave of non-zero values expanding outwards along the cycle from cell 0. The two wave frontiers proceed in opposite directions, eventually colliding on the cycle's topology at cell $N/2$ that is antipodal to cell 0. By combining Lemma 13 and Lemma 16 we see that, $S^{2^{i-1}-1}(1) = (10)^{2^{i-1}-1}1$, a string of length 2^{i-1} . Thus, at discrete time step $2^{i-1}-1$, the cells of the cycle are in state: $11:\overline{10^{2^{i-1}-1}1}$, implying that cells strictly alternate as 0, 1, 0, 1, ... in their value. Now at this time, because all cells witness local homogeneity (that is, for every cell, either both neighbors are 0 or both neighbors are 1), at the next discrete step, all cells in the system take value 0 (since $0 \oplus 0=1 \oplus 1=0$). Thus, starting from a simple initial state in which precisely one cell has the value 1 and all others have the value 0, we see that the cycle of $N=2^i$ cells converges in $2^{i-1}=N/2$ discrete time steps to being uniformly 0 everywhere.

Since every complex initial state can be decomposed into an \oplus sum of simple states by taking one summand for each cell that has the value 1—Lemma 3 can be applied to analyze the evolution of the system from complex states as well. Because every simple initial state converges to the state in which all cells have the value 0 in $T=2^{i-1}$ steps, Lemma 3 implies that every complex initial state also converges to the state in which all cells have the value 0 in $T=2^{i-1}$ steps. In other words, every initial state $X \rightarrow 0^N$, we have shown that the organism has precisely one attractor, namely 0^N .

We are also ready to prove Theorem 2 (stated originally on pp. 8): If regardless of initial state X the organism always ends up in the same attractor, then the number of cells in the organism is a power of 2.

Proof: By applying Lemma 19 and Lemma 3, it suffices to show that for a simple initial state $X^i=0^{N-1}1$ where i is the index of the cell with the value 1 if $X^i \rightarrow 0^N$ then for every complex intial state X composed of any set of X^i , $X \rightarrow 0^N$.

Applying Lemma 13, we know $S^2(N-1)=10^3$. Then, repeatedly apply Lemma 11 so that after each additional 2^{i-1} successor steps we have two cells of value 1 separated by $2^{i+1}-1$ cells with value 0. These cells wrap around a cyclic network of N cells. $1 \oplus 1=0$, by definition of the XOR function. In wrapping the cell values around the network of N cells, resulting state would be 0^N if the only two cells with value 1 collide $i+1$ at the same index. For this collision to occur the number of intervening zeros in $\overline{10^{2^{i+1}-1}1}$ equal $N-1 \bmod N$.

Therefore, in order for every state $0^{N-1}1$ to lead to the state 0^N for a network of size N the following must be true:

$$2^{i+1}-1 \equiv N-1 \bmod N$$

$$2^{i+1} \equiv 0 \bmod N$$

In other words, N divides 2^{i+1} .

We have shown that the organism must have size $N=2^j$ for some integer j .

Conclusions

In this work, we used an experimental approach to explore the

dynamics of homogeneous cyclic organisms of size $N=2,\dots,20$ cells, using software developed previously [8]. In these homogeneous organisms, each cell synchronously determines its successive states by computing the \oplus of the value of its neighbors. From observations of these computationally simulated dynamics, we conjectured that if the number of cells in an organism is a proper power of 2, then the organism has exactly one attractor, which has length 1 and consists of the state where all cells have a value of 0. We then formally proved this statement as well as its converse: that if regardless of its initial state the organism always ends up in the same size 1 attractor, then the number of cells in the organism is a proper power of 2. Some of the evidence for this now-proven “if and only if” relationship is rendered in Figures 14, 15 and 16 versus Figures 7 and 8.

Since the act of incrementing the size of an organism until it reaches double its size necessarily requires traversing a number that is a proper power of 2, any organism that grows to more than twice its original size will necessarily encounter a stage in which it has minimal adaptivity and maximal robustness. If the organism seeks to always maintain “intermediate” values of adaptivity and robustness as it grows, then alternative growth patterns that exhibit more than one attractor at powers of 2 sizes will be evolutionarily advantageous. The alternative growth pattern we explore experimentally in this work, is one in which the organism departs from cellular homogeneity to a minimal extent—allowing a single constituent cell to apply a rule that is different from XOR. Through experiments, we show that at sizes that are powers of 2, such minimally heterogeneous organisms avoid manifesting the low adaptivity that is provably exhibited in homogenous organisms. We conclude that cellular differentiation is one way an organism can avoid low adaptivity configurations that would otherwise necessarily be encountered during organism growth. It follows that if there is evolutionary pressure selecting for adaptivity, then the phenomenon of organism growth may express this pressure as a drive towards cellular differentiation and the progression from homogeneity towards heterogeneity. Figures 10-13 show the dynamics graphs of minimally heterogeneous organisms of sizes 2, 4, 8, and 16 respectively, and have increasing numbers of attractors. These figures are placed side by side with the dynamics graphs of homogeneous organism of the same size, to further illustrate their contrasting dynamics.

Future work will entail simulation of more complex growth patterns, beyond merely homogeneous and minimally heterogeneous growth. One pattern we plan to explore is probabilistic cellular differentiation during growth. Future work needs to look at both larger and more diverse organisms, but for larger organisms exhaustive simulation is computationally intractable, requiring advances in random sampling and estimation theory. By considering dynamics data from a more diverse and larger range of systems, it may be possible to identify other evolutionary pressures (beyond cell differentiation) and meta-phenomena that arise as organisms attempt to maintain a balance between adaptivity and robustness during growth.

References

1. Akutsu T, Kuhara S, Maruyama O, Miyano S (1998) A System for Identifying Genetic Networks from Gene Expression Patterns Produced by Gene Disruptions and Overexpressions. *Genome Inform Ser Workshop Genome Inform* 9: 151-160.
2. Amit D (1992) Modeling Brain Function: The World of Attractor Neural Networks. Cambridge University Press.
3. Arenas A, Díaz-Guilera A, Kurths J, Moreno Y, Zhou C (2008) Synchronization in complex networks. *Physics Reports* 469: 93-153.
4. Ball E (2011) Dynamic spread of social behavior in Boolean networks. University of Nebraska, Omaha.
5. Barrett C, Hunt HB, Madhav I, Marathe V, Ravi SS (2003) Reachability problems for sequential dynamical systems with threshold functions. *Theoretical Computer Science* 295: 1-3.
6. Barrett C, Hunt HB, Madhav I, Marathe V, Ravi SS, et al. (2003) Predecessor and permutation existence problems for sequential dynamical systems. *Discrete Mathematics and Theoretical Computer Science* 69-80.
7. G. Buzsaki (2006) *Rhythms of the Brain*. Oxford University Press.
8. Cantor Y, Khan B, Dombrowski K (2011) Heterogeneity and its impact on thermal robustness and attractor density. *Procedia Computer Science*, 6: 15-21.
9. Drossel B (2005) Number of attractors in random Boolean networks. *Phys Rev E Stat Nonlin Soft Matter Phys* 72: 016110.
10. Drossel B, Mihaljev T, Greil F (2005) Number and length of attractors in a critical Kauffman model with connectivity one. *Phys Rev Lett* 94: 088701.
11. Ellson J, Gansner ER, Koutsofios E, North SC, Woodhull G (2001) Graphviz - Open Source Graph Drawing Tools. *Graph Drawing* 2265: 483-484.
12. Ganguly N, Sikdar BK, Deutsch A, Canright G, Chaudhuri PP (2003) A survey on cellular automata. Technical report, Dresden University of Technology.
13. Gardner M (1970) The fantastic combinations of John Conway's new solitaire game "life". *Scientific American* 223: 120-123.
14. Gershenson C (2002) Classification of random boolean networks. Citeseer*.
15. Gershenson C, Kauffman SA, Shmulevich I (2005) The role of redundancy in the robustness of random boolean networks. Technical Report.
16. Ghosh K, Lerman K (2012) The role of dynamic interactions in multi-scale analysis of network structure.
17. Grabisch M, Rusinowska A (2010) A model of influence in a social network. *Theory and Decision*, 69: 69-96.
18. Kauffman SA (1993) *The Origins of Order: Self-Organization and Selection in Evolution*. (1stedn) Oxford University Press, USA.
19. Martin O, Odlyzko AM, Wolfram S (1984) Algebraic properties of cellular automata. *Comm Math Phys* 93: 219-258.
20. Mi Y, Zhang L, Huang X, Qian Y, Hu G, et al. (2011) Complex networks with large numbers of labelable attractors. *EPL (Europhysics Letters)* 95: 58001.
21. Miller JH, Page SE (2004) The standing ovation problem. *Complexity* 9: 8-16.
22. Mortveit HS, Reidys CM (2001) Discrete, sequential dynamical systems. *Discrete Mathematics*, 226: 281-295.
23. Nehaniv CL (2002) *Evolution in asynchronous cellular automata*. MIT Press, UK.
24. Neumann JV (1966) *Theory of Self-Reproducing Automata*. University of Illinois Press, Champaign, IL, USA.
25. Pai R, Ivanov I, Datta A, Bittner ML, Dougherty ER (2005) Generating Boolean networks with a prescribed attractor structure. *Bioinformatics* 21: 4021-4025.
26. Qiu Y, Tamura T, Ching WK, Akutsu T (2014) On control of singleton attractors in multiple Boolean networks: integer programming-based method. *BMC Syst Biol* 8 Suppl 1: S7.
27. Samuelsson B, Troein C (2003) Superpolynomial growth in the number of attractors in Kauffman networks. *Phys Rev Lett* 90: 098701.
28. Sarkar P (2000) A brief history of cellular automata. *ACM Computing Surveys* 32: 80-107.
29. Socolar JE, Kauffman SA (2003) Scaling in ordered and critical random boolean networks. *Phys Rev Lett* 90: 068702.
30. Sutner K (1989) Linear cellular automata and the garden-of-eden. *The Mathematical Intelligencer* 11: 49-53.
31. Williams T, Kelley C (2010) Gnuplot 4.4: an interactive plotting program.
32. Wolfram S (1989) Twenty problems in the theory of cellular automata. *Physica Scripta* (T9): 170-183.
33. Zhang SQ, Hayashida M, Akutsu T, Ching WK, Ng MK (2007) Algorithms for finding small attractors in Boolean networks. *EURASIP J Bioinform Syst Biol*.