9-2017

Morphogenesis and Growth Driven by Selection of Dynamical Properties

Yuri Cantor

The Graduate Center, City University of New York

How does access to this work benefit you? Let us know!
Follow this and additional works at: http://academicworks.cuny.edu/gc_etds

Part of the Bioinformatics Commons, Computational Neuroscience Commons, Dynamical Systems Commons, OS and Networks Commons, Other Cell and Developmental Biology Commons, Other Computer Sciences Commons, and the Theory and Algorithms Commons

Recommended Citation
http://academicworks.cuny.edu/gc_etds/2377

This Dissertation is brought to you by CUNY Academic Works. It has been accepted for inclusion in All Graduate Works by Year: Dissertations, Theses, and Capstone Projects by an authorized administrator of CUNY Academic Works. For more information, please contact deposit@gc.cuny.edu.
MORPHOGENESIS AND GROWTH DRIVEN BY SELECTION OF DYNAMICAL PROPERTIES

by

YURI ALEXANDER LOPAUR CANTOR

A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

2017
This manuscript has been read and accepted by the Graduate Faculty in Computer Science in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

Professor Bilal Khan

Date Chair of Examining Committee

Professor Robert Haralick

Date Executive Officer

Supervisory Committee

Professor Bilal Khan
Professor Nancy Griffeth
Professor Matthew Johnson
Professor Kirk Dombrowski

THE CITY UNIVERSITY OF NEW YORK
Abstract

MORPHOGENESIS AND GROWTH DRIVEN BY SELECTION OF DYNAMICAL PROPERTIES

by

YURI ALEXANDER LOPAUR CANTOR

Organisms are understood to be complex adaptive systems that evolved to thrive in hostile environments. Though widely studied, the phenomena of organism development and growth, and their relationship to organism dynamics is not well understood. Indeed, the large number of components, their interconnectivity, and complex system interactions all obscure our ability to see, describe, and understand the functioning of biological organisms.

Here we take a synthetic and computational approach to the problem, abstracting the organism as a cellular automaton. Such systems are discrete digital models of real-world environments, making them more accessible and easier to study then their physical world counterparts. In such simplified synthetic models, we find that the structure of the cellular network greatly impacts the dynamics of the organism as a whole. In the physical world, for example, the network property wherein some cells depend on phosphorus produces the cyclical boom-bust dynamics of algae on the surface of a pond. Using techniques of synthetic biology and cellular automata, such local properties can be abstractly specified, and the long-term, system-wide, and dynamical consequences of localized assumptions can be carefully explored.

This thesis explores the potential impacts of Darwinian selection of dynamical properties on long term cellular differentiation and organism growth. The focus here is on the relationship between organism homogeneity (or heterogeneity) and the dynamical properties of robustness, adaptivity, and chromatic symmetry. This dissertation applies an experimental
approach to test the following three hypotheses: (1) cellular differentiation increases the expected robustness in an organism’s dynamics, (2) cellular differentiation leads to more uniform adaptivity as the organism grows, and (3) for organisms with symmetry, growth by segment elongation is more likely than growth by segment reduplication. To explore these hypotheses, we address several obstacles in the experimental study of dynamical systems, including computational time limits and big data.
Acknowledgments

This dissertation work was conducted during the development of the NSF Office of Behavioral, Social, and Economic Sciences, Anthropology Program Grant BCS-0752680. The research was also supported by the CUNY University Fellowship.

I thank my advisor, Prof. Bilal Khan, for his patience, support, encouragement, advice, and guidance. I remember coming to him with a topic in network security and realizing our shared interest in evolution, biology, social networks, and dynamical systems. Bilal’s breadth of knowledge, creativity, precision, and concise explanations helped bridge the various disciplines and underlying theories that my research explores. Were it not for his shared interdisciplinary interest and support, I wouldn’t have had the courage to undertake this topic and research. Thank you for challenging, supporting, and inspiring me both in this thesis work and in my other projects!

I thank Prof. Nancy Griffeth, who was my advisor for the first half of my time in the doctoral program and stayed on my thesis committee after I changed topics. Nancy’s constant support and encouragement gave me an opportunity to research networks and ultimately focus my area of interest. Her advice and manner of teaching helped me both as a student and as an adjunct. It was inspiring to see how she engaged with students and encouraged them to apply what they learned. I will always remember our seminars and how we would start with topical, political, or just funny videos.

I thank my thesis committee members, Prof. Kirk Dombroski and Prof. Matthew John-
son, for their support, insightful comments, feedback, and encouragement throughout my research and dissertation work.

I thank Prof. Ted Brown for his support and guidance in finding supportive committee members and advisors and for his support in connecting me with CISDD research projects and funding.

I thank Prof. Kent Boklan for his encouragement, support, and advice both while on my committee and after when my topic changed. His insights and guidance in classes, research, and throughout my graduate work helped immensely.

I thank my colleagues and friends, Ali Assarpour, Anna Wisniewska, Constantinos Djouvas, David Brizan, Ou Liu, for being available to practice my presentations, to brainstorm, their advice, and for their help throughout. Without a supportive community, colleagues, and friends, this would not have been possible.

I also thank my family, Amber, Brian, Kryštof, Paul, and Yvonne, for their love, support, encouragement, and patience throughout.

Thank you!
# Contents

Contents viii

List of Tables xii

List of Figures xiii

1 Introduction 1

1.1 Background ................................................. 1

1.2 Preliminary Definitions ..................................... 6

2 How robustness drives differentiation 14

2.1 Introduction .................................................. 14

2.2 Definitions .................................................. 15

2.3 Formal Hypothesis 1 .......................................... 18

2.4 Methodology .................................................. 19

2.4.1 Program .................................................. 19

2.4.2 Simulation ............................................... 21

2.5 Results ...................................................... 23

2.5.1 Homogeneous Organisms ................................. 23

2.5.2 Heterogeneous Organisms ............................... 26

2.5.3 Comparison ............................................... 27
CONTENTS

2.5.4 Analysis ......................................................... 32
2.5.5 Summary ......................................................... 32

3 How adaptivity drives differentiation 35

3.1 Introduction ..................................................... 35
3.2 Definitions ....................................................... 37
3.3 Formal Hypothesis 2 ............................................ 37
3.4 Methodology ..................................................... 38
  3.4.1 Program ...................................................... 38
  3.4.2 Simulation ................................................... 39
3.5 Results .......................................................... 40
  3.5.1 Homogeneous Organisms .................................. 40
  3.5.2 Minimally Heterogeneous Organisms ...................... 42
  3.5.3 Comparison .................................................. 43
3.6 The Collapse of Adaptivity at Critical Sizes
  in Homogeneous Organisms ..................................... 46
  3.6.1 Definitions .................................................. 48
  3.6.2 Decompositions ............................................. 50
  3.6.3 The infinite case .......................................... 61
  3.6.4 Going from infinite to finite .............................. 65
  3.6.5 Summary .................................................... 67

4 How symmetry drives growth 69

4.1 Introduction ..................................................... 69
4.2 Definitions ....................................................... 71
4.3 Formal Hypothesis 3 ............................................ 74
4.4 Methodology ..................................................... 77
4.4.1 Program ................................................. 78
4.4.2 Simulation ............................................ 80
4.5 Results ..................................................... 81
  4.5.1 Reduplication ....................................... 81
  4.5.2 Elongation .......................................... 81
  4.5.3 Comparison ......................................... 82
  4.5.4 Summary ............................................. 83
5 Obstacles ..................................................... 85
  5.1 Introduction .......................................... 85
  5.2 Trading time for space ................................. 87
    5.2.1 Limitations ................................... 90
  5.3 Trading accuracy for time ............................. 91
    5.3.1 Technique 1 ................................... 94
    5.3.2 Technique 2 ................................... 95
    5.3.3 Limitations ................................... 96
  5.4 Parallel distributed processing ......................... 96
    5.4.1 Limitations ................................... 98
6 Software .................................................... 100
  6.1 Introduction .......................................... 100
  6.2 Structure ................................................ 101
    6.2.1 Client Daemon ................................ 101
    6.2.2 Server Daemon ................................ 102
    6.2.3 Database ....................................... 103
  6.3 Libraries ................................................. 106
  6.4 Usage .................................................... 107
CONTENTS

7 Future Work 111

7.1 Growth ......................................................... 112

7.1.1 Bilateral Symmetry ......................................... 113

7.1.2 Preliminary Results ........................................ 114

8 Conclusion 116

8.1 Results ......................................................... 117

8.1.1 Hypothesis 1 ................................................. 117

8.1.2 Hypothesis 2 ................................................. 118

8.1.3 Hypothesis 3 ................................................. 119

8.1.4 Analysis ...................................................... 120

8.2 Software ....................................................... 121

8.2.1 Trading time for space .................................... 121

8.2.2 Trading accuracy for time .................................. 122

8.2.3 Parallel distributed processing ............................. 123

8.3 Applications ................................................... 124

Bibliography 130
List of Tables

1.1 Truth table mapping with inputs at time $t$ and resulting output at time $t + 1$.
   Source [9] ................................................................. 7
1.2 XOR $b_0 b_1 b_2 b_3 = 0110$ truth table mapping .............................. 7
1.3 Table of update rules. [7] ................................................. 8
2.1 Heterogeneous organism sampled space ........................................ 21
3.1 Update rule 6 = XOR. ....................................................... 42
3.2 Table of minimally heterogeneous organism attractor count as size increases
3.3 Table of XOR truth table ................................................... 51
4.1 Results of testing growth by reduplication and by elongation. Source: [10] . . 83
6.1 Database tables and fields and their types ..................................... 106
6.2 Program usage for hypothesis 1 experiment .................................... 110
7.1 Results of testing bilateral symmetry resulting from single cell mutation.
   Source: [10] ....................................................................... 115
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Initial state. Source [7]</td>
<td>9</td>
</tr>
<tr>
<td>1.2</td>
<td>Successor state. Source [7]</td>
<td>9</td>
</tr>
<tr>
<td>1.3</td>
<td>Size 5 network. Source [7]</td>
<td>9</td>
</tr>
<tr>
<td>1.4</td>
<td>Attractor length 4.</td>
<td>11</td>
</tr>
<tr>
<td>1.5</td>
<td>Attractor length 2.</td>
<td>12</td>
</tr>
<tr>
<td>1.6</td>
<td>Attractor length 1.</td>
<td>13</td>
</tr>
<tr>
<td>2.1</td>
<td>Homogeneous OR size 4 Organism dynamics space with mutation edges.</td>
<td>16</td>
</tr>
<tr>
<td>2.2</td>
<td>Homogeneous OR size 4 Organism dynamics space with robust mutation edges.</td>
<td>17</td>
</tr>
<tr>
<td>2.3</td>
<td>Homogeneous organisms Class 1 with bounded numbers of attractors $\alpha$. [9]</td>
<td>24</td>
</tr>
<tr>
<td>2.4</td>
<td>Homogeneous Class 2 organisms with unbounded numbers of attractors $\alpha$. [9]</td>
<td>25</td>
</tr>
<tr>
<td>2.5</td>
<td>Class 1 Homogeneous organism robustness. [9]</td>
<td>26</td>
</tr>
<tr>
<td>2.6</td>
<td>Class 2 Homogeneous organism robustness. [9]</td>
<td>27</td>
</tr>
<tr>
<td>2.7</td>
<td>Attractors in heterogeneous organisms with std $\sigma$. [9]</td>
<td>28</td>
</tr>
<tr>
<td>2.8</td>
<td>Robustness in heterogeneous organisms with std $\sigma$. [9]</td>
<td>29</td>
</tr>
<tr>
<td>2.9</td>
<td>Robustness of homogeneous organisms. [9]</td>
<td>30</td>
</tr>
<tr>
<td>2.10</td>
<td>Robustness of heterogeneous organisms. [9]</td>
<td>31</td>
</tr>
<tr>
<td>2.11</td>
<td>Probability of heterogeneous organisms exceeding Class 2 robustness. [9]</td>
<td>31</td>
</tr>
<tr>
<td>2.12</td>
<td>Attractors in heterogeneous organisms exceeding class 2 robustness. [9]</td>
<td>32</td>
</tr>
</tbody>
</table>
3.1 Darwin Finches. Source: [70] .......................... 36
3.2 $\alpha$ for homogeneous XOR organisms size $n = 2, \ldots, 20$. Source [7] .......................... 41
3.3 XOR organism size 5. Source [7] .......................... 42
3.5 Homogeneous XOR vs Minimally Heterogeneous. [43] .......................... 44
3.6 XOR organism size 2. Source [7] .......................... 45

4.1 Starfish possible growth patterns. .......................... 71
4.2 The 4096 vertex dynamics graph of a size $n=12$ homogeneous organism in which all cells operate function 6 (XOR) Source: [10] .......................... 72
4.3 Attractor of size 4 in dynamics of homogeneous XOR organism size $n = 12$. Source: [10] .......................... 74
4.4 Coloring in orbit of attractor length 6 in dynamics of homogeneous XOR organism of size $n = 10$ Source: [10] .......................... 75
4.5 Coloring in orbit of attractor length 4 in dynamics of homogeneous XOR organism of size $n = 12$ Source: [10] .......................... 76
4.6 Coloring in orbit of attractor length 3 in dynamics of homogeneous XOR organism of size $n = 15$ .......................... 76
4.7 Attractor lengths of homogeneous organisms of sizes $n = 2 \ldots 50$ Source: [10] .......................... 78
LIST OF FIGURES

4.8 Estimated Percentage of Attractors with Nontrivial Rotational Symmetry

Source: [10] ................................................................. 79

4.9 % Symmetric: Reduplication vs Elongation ........................................... 84

5.1 Distributed Processing in Cluster. ....................................................... 99
Chapter 1

Introduction

1.1 Background

Humans regularly interact with, navigate, and create complex systems ranging from social networks and computer networks to ecological networks and biological networks; the importance of understanding the dynamics of complex networked systems can’t be overstated. The research herein seeks to understand and explore why networks have their particular dynamics, and how networks came to have those dynamics. In other words, this dissertation explores the co-evolution of networks and their dynamics.

Biological organisms are an example of a complex system, being made up of cellular networks. Studying the evolution of biological organisms might give insight into how evolutionary pressures could select for particular properties of these networks. The diverse set of biological organisms that populate the planet evolved through some selection process. If there is any correlation between network properties and dynamical properties, then evolutionary pressure that favors a subset of network properties simultaneously expresses a bias towards particular dynamical properties (and vice versa). Identifying correlations between properties of networks and system dynamics could facilitate understanding how we evolved
CHAPTER 1. INTRODUCTION

to where we are, predicting future evolution, and in the engineering of artificial networks with prescribed dynamical properties.

In this dissertation, the process of growth and the feature of homogeneity/heterogeneity are explored in relationship to three dynamical properties: robustness, adaptivity, and symmetry. To identify properties in the dynamics of an organism, the organism will be viewed at a fixed size. To identify how those dynamics came to exist in an organism, the organism will be considered as it “grows” or develops.

**Robustness** is the capacity for stasis—resilience to fluctuations in an environment or input. Consider, for example, reptiles and their robustness with respect to environmental temperature. When the environment is hot, a reptile’s body will heat up; when the environment is cold, a reptile’s body will cool down. Generally, reptiles are at the mercy of their environment when regulating their body temperature. In contrast consider a mammal’s body, which can maintain thermal equilibrium through metabolism and actions such as sweating to cool down or shivering to maintain warmth. Mammals have a robustness to environmental temperature that reptiles do not have. As described in the example of the horse and heat dissipation, the development of mechanisms like sweat glands and shivering can impact thermoregulation, that is, robustness against temperature changes in the environment.

**Adaptiveness** is the capacity to change—responsiveness to varied environments or inputs. Consider a plant and its adaptivity to the environment in contrast to a mammal. Cacti have robustness to hot and arid desert environments. These plants develop mechanisms for water retention but not for surviving freezing temperatures. Maple trees are able to survive freezing temperatures but lack the water retention mechanisms necessary to survive in a desert environment. In contrast, mammals have greater adaptivity and potential responsiveness to the varied climates through organs or body parts including limbs for movement, hair or fur, sweat glands, and fat or blubber. Returning to the example of heat dissipation and
horses, growth and development of sweat glands, panting, limbs for mobility to find shelter from solar radiation, and a long neck for greater conduction and convection of heat to air— all provide adaptivity to hot environments. A horse’s growth of fur and fatty tissue, limbs and muscles for movement, and organs that control metabolism provide adaptivity to cold environments.

When considering growth in organisms, there are several possibilities for types of growth: growth by the addition of another identical cell or growth by the addition of a different cell. Growing through the addition of another identical cell results in a **homogeneous** organism. Growing through the addition of a different cell results in cellular differentiation and a **heterogeneous** organism. Consider the differences between moss and grass. Moss lacks complex plant structures, growing without specialized tissue types including the xylem and phloem. These tissues transport water and nutrients throughout plants and trees enabling them to grow to larger sizes. In other words, water loss and nutrient transport can act as limiters on the maximal size of an organism like moss that lacks specialized structures formed through cellular differentiation. Simple organisms made up of homogeneous cells seem to reach a limit of how large they can grow based upon environmental factors. The more “advanced” a biological organism is, the greater the diversity in its cell structure. Does this latter growth type suggest that cellular differentiation is accompanied by an increase in a favorable dynamical property?

When considering growth in organisms, there are several possibilities for the rate of growth: growth by cell division resulting in a single additional cell or growth by multiple cells $m$ dividing resulting in $m$ additional cells. Consider a starfish and its growth. If a starfish starts from a single cell and always grows by a single cell, then its limbs will grow asymmetrically with one limb growing before the next or perhaps by adding a new limb. Alternatively, the starfish could grow symmetrically by a number of cells matching the number of limbs. In other words, each limb would increase by a cell at a time. Is there
a growth rate that has been selected for to preserve symmetry in biological organisms?

Mapping dynamics of biological organisms or their nervous systems poses challenges of creating discrete time intervals in analogue systems and direct cellular manipulation of the organisms themselves. A nervous system is a network of interconnected cells whose dynamics are both widely studied and not well understood. Biological nervous systems are immensely complex and testing on live organisms poses ethical concerns.

Computer simulations are well suited to address these limitations and challenges. Computer simulations are themselves discrete. The simulations can be manipulated at any level since simulations are being generated rather than a black box where only the input and output is accessible. Computer simulations are not considered live organisms and are not accompanied with ethical concerns for their creation, manipulation, testing, or destruction.

And yet, biological neural network behavior is difficult to model computationally. These networks are composed of densely connected neurons, have a state of excitation that is continuous in degree, and transmit state asynchronously [60]. There has been extensive research into biological networks and cellular automata (see [52, 18] for brief surveys). Cellular automata as described by Von Neumann [47] can be connected to form networks that serve as abstractions of biological networks. A computational simulation of these simplified networks of cellular automata can lead to understanding the behavior in correlated complex biological networks [31, 38, 55, 20].

In an experimental approach, abstractions are necessary to maximize the number of networks and depth of simulation that can be modeled. Further, it is known that decomposing super networks into sub networks can lead to understanding pathways and signaling in biological circuitry [63]. Random Boolean Networks (RBNs) have been used to abstract from continuous state values to discrete state values. Kauffman’s NK network model abstracts from densely connected to K connections [21, 22]. The trivial case where K = 1, restricts
the nodes to connect along a line while Conway’s “Game of Life” is an example of a Boolean \( NK \) network represented using a two dimensional grid where \( K = 8 \) [19]. Asynchronous transmissions with small temporal tolerances can be modeled as synchronous [46].

This thesis lies in the area of theoretical synthetic biology, using the more restricted class of synchronous Boolean cyclic \( NK \) networks (where \( K = 2 \)) as a formal basis. This class of networks has received considerable attention [57], and exhibits many of the phenomena seen in more general \( NK \) counterparts [54]. Even with these reductions in complexity, fully simulating the dynamics remains computationally intractable except for networks of relatively small size [61].

Understanding the relationship between network structural properties and dynamical properties is crucial to understanding how and why biological networks have evolved. This relationship can be used to construct networks with desirable properties [49]. Dynamical systems have been previously evaluated for their landscape ruggedness [40, 39], redundancy [22], reversibility and surjectivity (reachability and Garden of Eden states [57]) [30, 42]. I take an experimental approach to evaluate the relationship between network structural properties of heterogeneity, homogeneity, and growth to dynamical properties of robustness [9], adaptivity [7], and symmetry [10].

This thesis is divided into four parts. The first three parts correspond to three hypotheses intended to provide insight to the relationship between network structural properties and dynamical properties: cellular differentiation increases the expected robustness in an organism’s dynamics, cellular differentiation leads to increasing adaptivity, and for organisms with symmetry, growth by segment elongation is preferred to growth by segment reduplication. The fourth part will address practical concerns to the experimental approach arising from computational limitations and big data in experimental approaches to studying dynamical systems.
1.2 Preliminary Definitions

Throughout this thesis, networks of cellular automata will be referred to as organisms where each node in a network will be a cell of the organism. The application of biological terminology to describe theoretical and synthetic networks is common practice as seen since Von Neumann’s seminal work on cellular automata [47, 35]. As noted in the introduction the structure I am focusing on is selected to decrease computational complexity in order to facilitate exploration of larger organisms and deeper exploration of state space.

Structure. Informally, linear cyclic organisms are composed of cells that are directly connected to their two neighboring cells such that the cells form a ring where an initial cell is connected to the next cell and the last cell. Formally, the linear cyclic structure is modeled as an undirected cyclic graph $C = (V, E)$ of size $n$. Vertices are the cells and are enumerated $V = \{v_0, \ldots, v_{n-1}\}$ where each cell $v_i$ in $V$ is connected in cyclic order to two neighbors such that edges $E = \{(v_i, v_{i+1 \ (\text{mod} \ n)}) | i = 0, \ldots, n - 1\}$. The set of possible linear cyclic organisms of size $n$ are a subset of Kaufman’s NK-networks [12] of size $n$ where for each cell the number of inputs is $K = 2$. Figure 1.3 is an example of a size 5 linear cyclic organism.

The state of each cell at any point in time is Boolean, either 0 or 1, and deterministic by fixing a function $f : V \rightarrow F$ that assigns to each cell $v \in V$, a function $f(v)$ from $F = \{g : \{0, 1\}\{0, 1\} \rightarrow \{0, 1\}\}$, the set of all binary Boolean functions. The action of $f$ at a vertex $v_i$ can be thought of as a truth table mapping from the two inputs $K$, $v_i$’s left and right neighbors’ current state, to $v_i$’s cell state at the next time step. This function differs from the traditional Wolfram model [61, 64] in that Wolfram’s model used $K = 3$ and considered the self-input or state of cell $v_i$ as well. The bits $b_0$, $b_1$, $b_2$, $b_3$, as in Table 1.1, must be either 0 or 1 and the 4-bit binary string $b_0b_1b_2b_3$ is used to name the function $f$ as in Table 1.3. Note that because there are 2 choices for each of the 4 possible input, a cell’s
Table 1.1: Truth table mapping with inputs at time $t$ and resulting output at time $t+1$. Source [9]

<table>
<thead>
<tr>
<th>$s(v_{i-1}, t)$</th>
<th>$s(v_i, t)$</th>
<th>$s(v_{i+1}, t)$</th>
<th>$s(v_i, t+1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>*</td>
<td>0</td>
<td>$b_0$</td>
</tr>
<tr>
<td>0</td>
<td>*</td>
<td>1</td>
<td>$b_1$</td>
</tr>
<tr>
<td>1</td>
<td>*</td>
<td>0</td>
<td>$b_2$</td>
</tr>
<tr>
<td>1</td>
<td>*</td>
<td>1</td>
<td>$b_3$</td>
</tr>
</tbody>
</table>

Table 1.2: XOR $b_0b_1b_2b_3 = 0110$ truth table mapping

<table>
<thead>
<tr>
<th>$s(v_{i-1}, t)$</th>
<th>$s(v_i, t)$</th>
<th>$s(v_{i+1}, t)$</th>
<th>$s(v_i, t+1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>*</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>*</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>*</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>*</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

state must be 0 or 1 as a result of the possible inputs from its neighbors, $|F| = 2^{2^2} = 16$. For example, Table 1.2 is the XOR function truth table. Note that the space of possible functions for Wolfram’s model is a more expansive set such that $|F| = 2^{2^3} = 256$ [62]. In Table 1.3 these rules are defined using Boolean logic and displayed in ascending order according to the Boolean input values. For reference, rule 6 with name 0110 in Table 1.3 corresponds to the earlier example truth table shown in Table 1.2.

The organisms in this thesis are synchronous where every cellular state at the next time step is instantaneously determined according to:

$$s(v_i, t+1) = f(v_i) \left(s(v_{i-1} \text{ (mod } n), t), s(v_{i+1} \text{ (mod } n), t)\right)$$

for each $i = 0, \ldots, n - 1$ and $t \geq 1$. Note that results from synchronous models can be transformed and be realized in asynchronous models where synchronization stays within small tolerances locally [46]. This implies that research into synchronous organisms has implications for asynchronous organisms.
Table 1.3: Table of update rules. [7]

<table>
<thead>
<tr>
<th>Name</th>
<th>Rule</th>
<th>Boolean Logic</th>
<th>Dual</th>
<th>Common name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>Rule 0=</td>
<td>0</td>
<td>15</td>
<td>zero</td>
</tr>
<tr>
<td>0001</td>
<td>Rule 1=</td>
<td>(\neg v_{i-1} \land \neg v_{i+1})</td>
<td>14</td>
<td>nor</td>
</tr>
<tr>
<td>0010</td>
<td>Rule 2=</td>
<td>(\neg v_{i-1} \land v_{i+1})</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>0011</td>
<td>Rule 3=</td>
<td>(\neg v_{i-1})</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>0100</td>
<td>Rule 4=</td>
<td>(v_{i-1} \land \neg v_{i+1})</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>0101</td>
<td>Rule 5=</td>
<td>(\neg v_{i+1})</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0110</td>
<td>Rule 6=</td>
<td>((v_{i-1} \land \neg v_{i+1}) \lor (\neg v_{i-1} \land v_{i+1}))</td>
<td>9</td>
<td>xor</td>
</tr>
<tr>
<td>0111</td>
<td>Rule 7=</td>
<td>((\neg v_{i-1} \land v_{i+1}))</td>
<td>8</td>
<td>nand</td>
</tr>
<tr>
<td>1000</td>
<td>Rule 8=</td>
<td>(v_{i-1} \land v_{i+1})</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>1001</td>
<td>Rule 9=</td>
<td>((v_{i-1} \land v_{i+1}) \lor (\neg v_{i-1} \land \neg v_{i+1}))</td>
<td>6</td>
<td>xnor</td>
</tr>
<tr>
<td>1010</td>
<td>Rule 10=</td>
<td>(v_{i+1})</td>
<td>5</td>
<td>shift left</td>
</tr>
<tr>
<td>1011</td>
<td>Rule 11=</td>
<td>(v_{i+1} \lor (\neg v_{i-1} \land \neg v_{i+1}))</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>1100</td>
<td>Rule 12=</td>
<td>(v_{i-1})</td>
<td>3</td>
<td>shift right</td>
</tr>
<tr>
<td>1101</td>
<td>Rule 13=</td>
<td>(v_{i-1} \lor (\neg v_{i-1} \land \neg v_{i+1}))</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1110</td>
<td>Rule 14=</td>
<td>(v_{i-1} \lor v_{i+1})</td>
<td>1</td>
<td>or</td>
</tr>
<tr>
<td>1111</td>
<td>Rule 15=</td>
<td>(1)</td>
<td>0</td>
<td>one</td>
</tr>
</tbody>
</table>

Homogeneity and heterogeneity have previously been studied with respect to cell connectivity to other cells [37] and function determining cell state [25]; however, in this thesis they will refer only to functions. In this thesis, cell connectivity is homogeneous such that each cell will be connected in a cycle with only two direct neighbors. If every cell in an organism has the same function assigned to it, then the organism is homogeneous. There are 16 unique homogeneous organisms of any given size \(n\); the set of all homogeneous organisms of size \(n\) will be denoted \(\text{Hom}(n)\), and these organisms are named explicitly as \(X_0^n, X_1^n, \ldots, X_{15}^n\).

If not every cell in an organism has the same function assigned to it, then the organism is heterogeneous. There are \(16^n - 16\) heterogeneous organisms of size \(n\). The set of all heterogeneous organisms of size \(n\) will be denoted \(\text{Het}(n)\). Often we will consider a random sample of \(\text{Het}(n)\) because its size makes it too large to explore exhaustively.

Formally, an organism is homogeneous if \(|\text{Im}(f)| = 1\); otherwise it is heterogeneous. There has been research into the distance between functions using hamming distance as a
measure of proximity [32, 64]. Here, I simplify and look only at the number of different functions instead of the specificity of what those functions are and their distance from each other. For example, I will later consider organisms that are minimally heterogeneous; these will be homogeneous organisms in which just one cell has deviated to a different function.

The organism state, as in Figure 1.3, is the ordered set of its cell states at time \( t \). This set of states for example in Figure 1.1, can be expressed as the sequence of states \( v_0v_1v_2v_3v_4 = 00010 \).

**Dynamics.** The dynamics of the organism are represented as a directed graph \( S \) connecting organism states, as in Figure 1.1, to their successor, seen in Figure 1.2, representing the organism’s phase space, in which \((X,Y)\) is an edge if it can be said that whenever the organism is in state \( X \) at time \( t \), it is necessarily (absent noise) in state \( Y \) at time \( t + 1 \). Formally, the dynamics is a directed graph \( S = (2^V, D) \) whose vertex set consists of all possible states of the organism (i.e. the power set of \( V \)), and whose edge set \( D \) includes every ordered pair \((X,Y)\) for which \( s^+(t) = X \implies s^+(t + 1) = Y \). If an organism state \( X \) is connected to organism state \( Y \) by a directed edge from \( X \) to \( Y \), then \( Y \) is the successor state of \( X \). Not only have successor functions been studied extensively, but their resulting dynamics and dynamical components have been the focus of research problems [62].

In the dynamics space of an organism, a **Garden of Eden state** is a state that cannot be reached from any other state by a directed edge. In other words, no state in the dynamics
of the organism has as its successor a Garden of Eden state. This dynamical component has
drawn attention as part of the problem of reachability: to determine if a state is a Garden of Eden state [57]. Research into Garden of Eden states involves developing algorithms for inverting the successor function or reversibility [30, 42]. Wuensche and Lesser have introduced a reverse algorithm [69].

An attractor state is an organism state that occurs in a cycle in the dynamics space of an organism. In other words, if \( X \) is an attractor state\(^1\), then there exists some discrete number of time intervals \( j \) such that when starting at state \( X \) and time \( t \) and computing the successor states at times \( t + 1, \ldots, t + j \) then at the state at time \( t = t + j \) will be \( X \). An attractor is the set of all states that are in the same cycle in the dynamics space of an organism. The attractor length is the number of states in an attractor. Given an organism \( X \) we will denote the set of all of its attractors as \( \text{Att}(X) \).

A tributary state is a state that is not in an attractor. When starting at a tributary state and computing the successor states, eventually an attractor state will be encountered because the state space has a discrete number of possible states. A tributary is the set of states composing a path of tributary states that lead to the first encountered attractor state when computing forward from a Garden of Eden state. An attractor state can have many tributaries or no tributaries. The basin of attraction [66] is formed by these tributaries.

An attractor with its tributaries composes a single connected component of the dynamics graph of an organism. The graph of an organism’s dynamics space is made up of these components. Figures 1.4, 1.5, and 1.6 show an example decomposition of the size 12 homogeneous XOR organism. The complete dynamics graph for the homogeneous XOR organism of size 12 has 6 attractors of length 2 (Figure 1.4), 60 attractors of length 4 (Figure 1.5), and 4 attractors of length 1 (Figure 1.6). In the Figures, nodes are labeled with the

\(^1\)Here the term attractor encompasses the classes of attractors defined by Wolfram including: fixed point, periodic, and strange [61]. In Hopfield networks, attractors are the content-addressable memory [29].
organism state, which is composed of the Boolean state of the cells. Directed edges lead from an organism state to its successor state. Black edges connect tributaries to their successor state and blue edges connect attractor states to their successor state [10].

Figure 1.4: Attractor length 4.
Figure 1.5: Attractor length 2.
Figure 1.6: Attractor length 1.
Chapter 2

How robustness drives differentiation

2.1 Introduction

The evolutionary trend in organisms has been away from homogeneity and towards heterogeneity and complex structures. Returning to the comparison of moss to grass, it is clear that grass has more complex structures involving a greater degree of cellular differentiation. In particular, grass has specialized tissue types including the xylem and phloem that transport water and nutrients throughout the plants enabling them to grow taller. Grass also has root structures that enable it to draw water from the soil. Being able to draw water from the soil and transport it throughout the plant makes grass more robust to drier environments than moss, which relies on diffusion and osmosis to absorb and transport water and nutrients. The variety and complexity of grass is derived from differentiation or heterogeneity of its cellular structure. Since natural selection is a driving force for evolution, one might expect some properties of the underlying structure [59] and the resulting dynamics are being selected for [13], driving evolution towards heterogeneity. This can be observed in nature by comparing the set of initial and early organisms to more recent contemporary organisms. Testing this hypothesis with synthetic organisms requires simulating the two sets of organ-
isms, homogeneous and heterogeneous, and identifying a set of dynamical properties that exists in the later set that is missing in the former set.

Heterogeneity in the genetics of species is crucial to survival, and a lack of diversity in genetics often carries with it the risk of extinction [16]. Genetic homogeneity in a species results in a narrow set of features, which in turn implies vulnerability to environmental change. In the context of synthetic biology, the changes we will consider are thermal noise that induces mutations in cell state. Since an attractor is a form of dynamic stability, we will take dynamic robustness to mean the ability to preserve dynamic stability in the presence of thermal noise. Following this analogy, informally I assert the following hypothesis: **Cellular differentiation increases the expected robustness in an organism’s dynamics.**

### 2.2 Definitions

Here we consider thermal noise that is capable of inducing mutations in the state of random cells in the organism. A **mutation** is achieved by flipping a single bit in the organism state [32]. The organism state that results from a mutation of a single random cell state is a **mutation state**. Wuensche describes graphs of these perturbations as attractor jump-graphs [67]. Mutation states are connected to the originating attractor state by a **mutation edge**. Figure 2.1 shows the dynamics space of the size 4 homogeneous OR organism with mutation edges; edges leaving a tributary state are black, edges leaving an attractor state are blue, and mutation edges leaving an attractor state are red. Each organism state of an organism of size $n$ has $n$ possible resulting mutation states and outgoing mutation edges. Note that in the example in Figure 2.1, each attractor state has 4 red mutation edges leaving it [9]. Over infinite time, if thermal noise is low an organism spends most of the time cycling in attractors. As such, in this thesis the focus is only on mutation edges leaving attractor states.
This type of perturbation is one of many that have been explored. Perturbations can also be made in the cell function [65], resulting in a change in the wiring or circuitry of the organism or in the timing or synchronicity of the signal [8]. There has been research into the distance between cell states and cell functions that uses the hamming distance between two binary values [32, 64]. For example, the distance \( D \) between a state 00000 and another state 00111 is \( D = 3 \) as there are three bits with differing values.

![Homogeneous OR size 4 Organism dynamics space with mutation edges.](image)

A mutation state is an organism state that is either in an attractor or connected to an attractor through directed edges in the dynamics graph. If this destination attractor is the originating attractor itself, then the mutation edge is robust; otherwise, the mutation edge is not robust. Returning to the example of the size 4 homogeneous OR organism, Figure 2.2
shows the dynamics space of this organism with non-robust mutation edges in red and robust mutation edges in green [9].

Robustness is the ability to resist external influences [6, 11], and in the case of the organism dynamics, robustness is the ability to conserve the dynamical topology [14, 24, 51]. In the case of synchronicity of signals, robustness is the ability to resist change arising from perturbations in synchronicity in the update scheme [8].

Figure 2.2: Homogeneous OR size 4 Organism dynamics space with robust mutation edges.

We will define the robustness of attractor $A$ as the number of robust mutation edges $R[A]$ divided by the total number of mutation edges $E[A]$ leaving $A$’s states. An attractor $A$ of length $|A| = L$ in the dynamics space of an organism of size $n$ has $nL$ mutation edges; thus, if the attractor $A$ had $R[A] = r$ robust mutation edges its attractor robustness would be
\[ \rho(A) = \frac{r}{nL}. \]

We define the robustness of organism \( X \) as the average attractor robustness of all its attractors \( \text{Att}(X) \). If the total number of attractors in the dynamics space of \( X \) is \( \alpha(X) = |\text{Att}(X)| \), we define organism robustness to be:

\[
\rho(x) = \frac{1}{\alpha(X)} \sum_{A \in \text{Att}(X)} \rho(A) = \frac{1}{\alpha(X)} \sum_{A \in \text{Att}(X)} \frac{R[A]}{E[A]}.
\] (2.1)

Note that organism robustness could alternately be computed as

\[
\frac{\sum_{A \in \text{Att}(X)} R[A]}{\sum_{A \in \text{Att}(X)} E[A]}.
\]

In order to keep the attractor properties distinct from each other, in this thesis I apply the definition (2.1), where organism robustness is taken to be average attractor robustness.

### 2.3 Formal Hypothesis 1

Throughout this work if \( X \) is an organism, then \( |X| \) denotes the number of cells in \( X \). The organism of size \( n \) where all cells operate according to function \( i \) is denoted \( X^n_i \). The set of all homogeneous organisms of size \( n \) is denoted

\[ \text{Hom}(n) = \{ X^n_i \mid i = 0, \ldots, 15 \}. \]

The set of all heterogeneous organisms of size \( n \) is denoted

\[ \text{Het}(n) = \{ X \mid X \text{ is a heterogeneous organism of size } n \}. \]
CHAPTER 2. HOW ROBUSTNESS DRIVES DIFFERENTIATION

Applying the definitions, Hypothesis 1 can be posed formally as follows

\[ \text{mean}\{\rho(X) \mid X \in \text{Het}(n)\} > \text{mean}\{\rho(X) \mid X \in \text{Hom}(n)\}. \]

If the hypothesis is true, it would imply that as an organism is developing, if robustness is to be selected for, then the organism is likely to undergo cellular differentiation. At the scale of populations, we will be more likely to see that large organisms are heterogeneous while smaller ones tend to be homogeneous. The hypothesis reflects what is observed in nature: the bigger the organism the more likely it is to be heterogeneous and contain complex specialized structures formed of differentiated cells.

2.4 Methodology

The methodology for testing this hypothesis has three components: developing a program, executing the program to simulate and generate experimental data, and finally analyzing the collected data. In this section I explain the program and its execution; data analysis is covered in the results section.

2.4.1 Program

The program to test hypothesis 1 is written in C and sequentially computes the full organism state space, the number of attractors, attractor robustness, and organism robustness. The program outputs data files for a dynamics graph with mutation edges that can be rendered using graphviz [15]. Note, there is existing software, in particular Discrete Dynamics lab [68], that has been used extensively to model, simulate, and study dynamical systems. However, since the focus in this thesis is to explore properties and dynamics that differ from ones previously explored, the benefit of developing a code base is that the program is highly cus-
tomizable. Developing a program instead of using the existing software enables optimization for larger organisms where exhaustive search of state space is limited, $n \leq 12$ [64]. Developing a program also allows implementation of novel algorithms for simulation (see Chapter 5 section 5.2).

For each organism $X$ simulated, the program fully explores the state space and computes the number of attractors by sequentially iterating through each binary organism state and simulating forward to the successor states until an attractor is found. The number of attractors in an organism is the count of the unique attractors $\alpha(X)$ discovered while fully exploring the state space of an organism.

The program computes the robustness of each attractor $A$ by sequentially iterating its states, and for every possible single bit mutation of each cell simulating forward through successor states until the next attractor is found. Attractor robustness $\rho(A)$ is the probability that a random mutation edge of the attractor state will lead back to the same attractor. Organism robustness $\rho(X)$ is computed as the average attractor robustness, over all of the organism’s attractors.

In this experiment, the program selects random heterogeneous organisms. To select a heterogeneous organism of size $n$, the program iterates through each of the $n$ cells and uses a function $f_r$ to assign to each cell $v$ in the organism a randomly chosen binary Boolean function from the 16 choices available. The probability that this sampling generates a homogeneous organism is $\frac{1}{16^{n-1}}$, and tends to 0 for large $n$. This function $f_r$ uniformly randomly samples an organism from the total population of $16^n$ unique organisms of size $n$.

The program also selects all 16 unique homogeneous organisms of size $n$. For each unique homogeneous organism $X^n_i = 0, \ldots, 15$ the corresponding Boolean function $i$ is assigned to each of the $n$ cells of that organism.
Table 2.1: Heterogeneous organism sampled space

<table>
<thead>
<tr>
<th>( n )</th>
<th># sampled</th>
<th>population size</th>
<th>% sampled</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1000</td>
<td>256</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>1000</td>
<td>4.10E+03</td>
<td>24.4</td>
</tr>
<tr>
<td>4</td>
<td>1000</td>
<td>6.55E+04</td>
<td>1.53</td>
</tr>
<tr>
<td>5</td>
<td>1000</td>
<td>1.05E+06</td>
<td>0.095</td>
</tr>
<tr>
<td>6</td>
<td>1000</td>
<td>1.68E+07</td>
<td>0.006</td>
</tr>
<tr>
<td>7</td>
<td>1000</td>
<td>2.68E+08</td>
<td>3.73E-04</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>4.30E+09</td>
<td>2.33E-05</td>
</tr>
<tr>
<td>9</td>
<td>1000</td>
<td>6.87E+10</td>
<td>1.46E-06</td>
</tr>
<tr>
<td>10</td>
<td>1000</td>
<td>1.10E+12</td>
<td>9.10E-08</td>
</tr>
<tr>
<td>11</td>
<td>1000</td>
<td>1.76E+13</td>
<td>5.68E-09</td>
</tr>
<tr>
<td>12</td>
<td>1000</td>
<td>2.82E+14</td>
<td>3.55E-10</td>
</tr>
<tr>
<td>13</td>
<td>1000</td>
<td>4.50E+15</td>
<td>2.22E-11</td>
</tr>
<tr>
<td>14</td>
<td>1000</td>
<td>7.21E+16</td>
<td>1.39E-12</td>
</tr>
<tr>
<td>15</td>
<td>1000</td>
<td>1.15E+18</td>
<td>8.67E-14</td>
</tr>
</tbody>
</table>

2.4.2 Simulation

The experiment to test hypothesis 1 simulates and computes organism robustness both for a set of heterogeneous organisms at fix sizes and for all of the homogeneous organisms at the same fixed sizes. Simulation and data collection testing hypothesis 1 was conducted on organism sizes that could be fully explored. The program explored all 16 unique homogeneous organisms for sizes \( n = 2, \ldots, 16 \) [9].

The number of distinct heterogeneous organisms, \(|F|^{|V|} = 16^n\), is too large to exhaustively simulate except for small values of \( n \). Consequently, in this experiment I assume a randomly sampled population of heterogeneous organisms of a fixed size \( n \) to be representative of the complete set of heterogeneous organisms of the same fixed size \( n \). The experiment selects this subset of all possible heterogeneous organisms by randomly sampling 1,000 heterogeneous organisms of size \( n \) from the \( 16^n \) organisms possible [9].

The expected \( \alpha \) and \( \rho \) for heterogeneous organisms of size \( n \) are estimated from the average robustness from corresponding subsets of heterogeneous organisms.
The data from the simulations is used to compute dynamical properties between the sets of homogeneous and heterogeneous organisms and reference the following metrics:

The **average attractor count** \( \hat{\alpha}(n) \) for size \( n \) homogeneous organisms is \( \hat{\alpha}(n) = mean(Hom(n)) \). This value is computed as:

\[
\hat{\alpha}(n) = \frac{\sum_{i=0}^{15} \alpha(X_i^n)}{16}
\]

where \( \alpha(X_i^n) \) is the number of attractors of the homogeneous organism \( X_i^n \) (having size \( n \) and operating using function \( i \) at every cell).

Average attractor count for size \( n \) heterogeneous organisms \( Het(n) \) must be estimated because the space of possible heterogeneous organisms is too large to be enumerated for all but the smallest \( n \). The expected average attractor count \( \bar{\alpha} \) for \( Het(n) \) is estimated by sampling \( s \) heterogeneous organisms \( Y_1, \ldots, Y_s \) from \( Het(n) \) uniformly at random and then taking

\[
\bar{\alpha}(n) \approx \frac{\sum_{i=1}^{s} \alpha(Y_i)}{s}.
\]

The **average robustness** \( \hat{\rho}(n) \) for size \( n \) homogeneous organisms \( X_i^n \) where \( f_i \) assigns the same function to every cell is computed as follows:

\[
\hat{\rho}(n) = \frac{\sum_{i=0}^{15} \rho(X_i^n)}{16}.
\]

Average robustness for heterogeneous organisms \( Het(n) \) of size \( n \) must also be estimated. Expected average robustness \( \bar{\rho}(n) \) is estimated by sampling \( s \) heterogeneous organisms \( Y_1, \ldots, Y_s \) from \( Het(n) \) uniformly at random and then taking:

\[
\bar{\rho}(n) \approx \frac{\sum_{i=1}^{s} \rho(Y_i)}{s}.
\]
Note, that $\hat{\alpha}(n)$ and $\hat{\rho}(n)$ take the average across a complete set while $\bar{\alpha}(n)$ and $\bar{\rho}(n)$ estimate the average across a random sample.

### 2.5 Results

The results are broken into three parts: homogeneous organisms (2.5.1), heterogeneous organisms (2.5.2), and the comparison of homogeneous to heterogeneous (2.5.3).

#### 2.5.1 Homogeneous Organisms

The number of attractors found from fully exploring all 16 homogeneous organisms of sizes $2 \ldots 16$ can be divided into two classes based upon the number of attractors the organism has as the organism increases in size. A homogeneous organism of size $n$ with function $f$ is of **Class 1** if $\alpha$ remains bounded by some constant $b$ as the organism grows $n \to \infty$ (see Figure 2.3). A homogeneous organism of size $n$ with function $f$ is of **Class 2** if for all constants $b$, there exists a size $n_b$ at which the number of attractors $\alpha$ exceeds $b$ (see Figure 2.4) [9].

In this experiment the state space of the organisms is fully explored; therefore, $\alpha$ is directly computed as the count of the unique attractors discovered. Figure 2.3 plots the resulting $\alpha$ for homogeneous organisms of increasing size where each organism’s plot is labeled by the function every cell of the homogeneous organisms uses to determine its state. Figure 2.3 plots $\alpha$ for the following Class 1 homogeneous organisms: $0000, 0001, 0111, 1111$ with the y-axis as $\alpha$ and the x-axis as organism size $n$. In these 4 organisms $\alpha$ is bounded as the organisms grow in size. Figure 2.3 shows $\alpha$ is either constantly 1, or oscillates between 2 and 3 [9].

Figure 2.4 plots $\alpha$ for homogeneous organisms of increasing size where each organism’s plot is labeled by the function every cell of the homogeneous organisms uses to determine
its state. Figure 2.4 plots $\alpha$ for the Class 2 homogeneous organisms: 0010, 0011, 0100, 0101, 0110, 1000, 1001, 1010, 1011, 1100, 1101, 1110 with the y-axis as $\alpha$ and the x-axis as organism size. In Figure 2.4 $\alpha$ for these 12 organisms is visibly increasing without bound as the organisms grow in size [9].

**Homogeneous Class 1 Robustness**

Organism robustness $\rho$ is computed as the average of the attractor robustness for every unique attractor in the organism. Figure 2.5 plots the resulting $\rho$ for homogeneous organisms of increasing size where each organism’s plot is labeled by the function every cell of the homogeneous organisms uses to determine its state. Figure 2.3 plots robustness $\rho$ for homogeneous organisms: 0000, 0001, 0111, 1111 with the y-axis as $\rho$ and the x-axis as organism size $n$. For the Class 1 homogeneous organisms 0000 and 1111 that have constant $\alpha = 1$
shown in 2.3, it is expected that $\rho = 1$ because there is no other attractor that can be reached from a mutation edge. Class 1 homogeneous organism 0001 and 0111 whose attractors are bound between 2 and 3 for $n > 2$ have $\rho = 0.5$. Consequently, the Class 1 homogeneous organisms maintain relatively high robustness [9].

**Homogeneous Class 2 Robustness**

Figure 2.6 plots $\rho$ for homogeneous organisms of increasing size where each organism’s plot is labeled by the function every cell of the homogeneous organisms uses to determine its state. Figure 2.6 plots $\rho$ for homogeneous organisms: 0010, 0011, 0100, 0101, 0110, 1000, 1001, 1010, 1011, 1100, 1101, 1110 with the y-axis as $\rho$ and the x-axis as organism size $n$. Class 2 homogeneous organisms have decreasing robustness with $\liminf \rho$ tending toward a value $< 0.2$. The exceptions preventing this from being a uniform limit occur for organisms 0110 and
1001 that spike to $\alpha = 1$ in Figure 2.4 and $\rho = 1$ in Figure 2.6 at sizes $2^i$. Consequently, the Class 2 homogeneous organisms exhibit relatively low robustness, which diminishes during organism growth towards a value < 0.2 [9].

### 2.5.2 Heterogeneous Organisms

**Attractors**

The experiment uses a set of 1000 randomly sampled heterogeneous organisms at sizes $2, \ldots, 16$ and $\bar{\alpha}$ is the mean attractor count for the sampled set of heterogeneous organisms. Figure 2.7 plots the $\bar{\alpha}$ for these sampled heterogeneous organisms at increasing sizes with $\bar{\alpha}$ as the y-axis and organism size as the x-axis. The plot clearly shows that the expected number of attractors $\bar{\alpha}$ increases unboundedly as the sizes of heterogeneous organisms increase [9].
CHAPTER 2. HOW ROBUSTNESS DRIVES DIFFERENTIATION

Robustness

\( \bar{\rho} \) is the mean robustness for the sampled set of heterogeneous organisms. Figure 2.7 plots the experimental results showing the mean robustness \( \bar{\rho} \) for the sampled heterogeneous organisms at increasing sizes with \( \bar{\rho} \) as the y-axis and organism size \( n \) as the x-axis. The plot clearly shows that the expected robustness \( \bar{\rho} \) decreases approaching 0.5 as the sizes of heterogeneous organisms increase [9].

2.5.3 Comparison

Hypothesis 1 compares the average robustness of homogeneous organisms to the average robustness of heterogeneous organisms. Due to computational limitations in this experiment, the average robustness of a randomly sampled heterogeneous organism is computed instead and assumed to be representative of the average robustness of heterogeneous organ-
isms re-framing Hypothesis 1 to be: Let the average robustness of the sampled set of all heterogeneous organisms \( X \) at any fixed size \( n \) be \( \bar{\rho}(X_n) \) and let the average robustness of the set of all homogeneous organisms \( Y \) at the fixed size \( n \) be \( \hat{\rho}(Y_n) \) then for \( n \geq 2 \)

\[
\bar{\rho}(X_n) > \hat{\rho}(Y_n).
\]

Figure 2.9 shows the proportion of all homogeneous organisms from both classes. The chart uses the percent of all classes of homogeneous organisms as the y-axis and the organism size as the x-axis. Thermal robustness is shown in three bands: the low band: \( 0.0 \leq \rho \leq 0.2 \), the middle band: \( 0.2 < \rho \leq 0.8 \), and the high band: \( 0.8 < \rho \leq 1 \). From this chart it is clear that only the four Class 1 homogeneous organisms maintain uniform robustness \( \rho > 0.2 \). The 12 class 2 homogeneous organisms suffer from low robustness.

Figure 2.10 applies the same bands for heterogeneous organisms but shows even though
CHAPTER 2. HOW ROBUSTNESS DRIVES DIFFERENTIATION

expected robustness $\bar{\rho}$ decreases for heterogeneous organisms as organism size increases, the relative proportion of heterogeneous organisms with low robustness $\rho \leq 0.2$ is insignificant.

Given the results of class 1 homogeneous organisms, comparison of all homogeneous organisms and heterogeneous organisms would be skewed by the high robustness of the class 1 homogeneous organisms. Therefore, the robustness of class 2 homogeneous organisms is directly compared to the robustness of the sampled set of heterogeneous organisms.

Homogeneous Class 2 vs Heterogeneous

Figure 2.11 plots the probability that $\bar{\rho} > \rho_2$ where $\rho_2$ is the maximum robustness of all class 2 homogeneous organisms with the y-axis as probability and the x-axis as organism size. In other words, the chart estimates the probability that robustness of a randomly chosen heterogeneous organism at size $n$ will be greater than the robustness of all Class 2 organisms.
at the same size. In Figure 2.11 this probability visibly tends to 1 except for sizes that are a power of 2 (a set of measure 0).

Figure 2.12 shows the expected number of attractors \( \bar{\alpha} \) for just those heterogeneous organisms whose \( \rho > \rho_2 \) shown in Figure 2.11. Note that at the sizes \( 2^i \) there are no heterogeneous organisms with \( \rho > \rho_2 \) and consequently there is no \( \bar{\alpha} \). Disregarding organism sizes \( 2^i \), as organism size increases Figure 2.12 shows the expected number of attractors grow without bound for the heterogeneous organisms whose robustness exceeds that of the class 2 homogeneous organisms of the same size.

Note, the dips seen in Figures 2.11 and 2.12 at sizes \( 2^i \) will be revisited in Chapter 3.
Chapter 2. How Robustness Drives Differentiation

Figure 2.10: Robustness of heterogeneous organisms. [9]

Figure 2.11: Probability of heterogeneous organisms exceeding Class 2 robustness. [9]
2.5.4 Analysis

The results from Figures 2.3, 2.4, 2.5, and 2.6 show two possibilities for homogeneous organisms that grow in size while remaining homogeneous. If an organism is class 1 then it will have high robustness and a bounded number of attractors. If an organism is class 2, then it will have low robustness but an unbounded number of attractors.

In contrast, Figures 2.7 and 2.8 show that heterogeneous organisms have both high expected robustness and unbounded expected number of attractors. Further, Figures 2.11 and 2.12 show that in comparison to the class 2 homogeneous organisms, a randomly selected heterogeneous organism of the same size will, with probability tending towards 1, have greater robustness and typically have increasing numbers of attractors as they grow in size.

2.5.5 Summary

Robustness is the resilience to an environment or input. It is known that more evolutionarily “advanced” animals have greater robustness than less evolutionarily advanced animals. For
example mammals have robustness to environmental temperature that reptiles do not have. This robustness is a result of specialization during morphogenesis and cellular differentiation.

Starting with the assumption that all organisms begin as a single cell, if increased adaptivity is selected for, then growth is required. By definition, a single cell organism is homogeneous. During growth by cellular division, the single cell organism increases in size by the addition of a single cell that is either of the same cell type thus remaining homogeneous or of a different cell type becoming heterogeneous.

Assume that organisms grow iteratively by a single cell in pursuit of increasing adaptivity. In the case of organisms remaining homogeneous during growth, the experimental results show that

- Class 1 homogeneous organisms exhibit constant or alternating adaptivity and
- Class 2 homogeneous organisms exhibit increasing adaptivity.

Hypothesis 1 states that the average robustness of the set of heterogeneous organisms will be greater than the average robustness of the set of homogeneous organisms:

\[
\text{mean}\{\rho(X) \mid X \in \text{Het}(n)\} > \text{mean}\{\rho(X) \mid X \in \text{Hom}(n)\}.
\]

This means that as an organism grows, if robustness is selected for, then at some point during growth the organism can be expected to undergo cellular differentiation and become heterogeneous. In other words, Hypothesis 1 suggests that selection for the dynamical property of robustness during growth will drive the cellular differentiation and morphogenesis.

The experimental results shown in Table 2.11 confirm Hypothesis 1 holds for the Class 2 homogeneous organisms showing that for the limited set of synchronous linear cyclic Boolean organisms sampled of sizes 2, \ldots, 16 with increasing adaptivity:

- heterogeneous organisms have a probability approaching 1 of having greater robustness than the Class 2 homogeneous organisms.
Table 2.12 confirms that the

- heterogeneous organisms that have greater robustness than Class 2 homogeneous organisms also have increasing adaptivity.

Hypothesis 1 contributes an evolutionary motivation for cell differentiation when selecting for dynamical properties of a network. Applying Hypothesis 1 can improve artificially constructed networks by linking network properties to expected dynamical properties thereby enabling networks to be constructed with prescribed dynamical properties.
Chapter 3

How adaptivity drives differentiation

3.1 Introduction

When an organism consists of just a single cell, it is homogeneous. However, nearly all multicellular organisms are heterogeneous. What properties resulting from cellular differentiation might be selected for during growth, and which in turn drive the transition from homogeneity to heterogeneity? This thesis seeks to identify a property that, selected for, might lead to the initial cellular differentiation, that first deviation from a homogeneous organism towards a minimally heterogeneous one.

Returning to the comparison of moss to grass, if moss is a precursor of grass what might drive the initial differentiated structures like the xylem and phloem? The more complex structures, including roots, xylem, and phloem, collectively enable the organism to adapt to a greater variety of environments and changes in the environment. For moss to continue growing it would require possibly adapting at the very least transporting water within itself over increased distances. Continual growth might require adaptation to bigger support structures like roots to adapt to wind or water scarcity. Each of these adaptations requires specialization and differentiation.
In biological networks, specialization of cells in heterogeneous organisms helps facilitate their adaptivity to environments (e.g., the famous example of the finch’s varying beaks in Figure 3.1 [70]). In the realm of synthetic biology, one analogue to specialization or variation is an attractor. Could it be that maintaining adaptivity during growth somehow forces cellular differentiation? If so, then during growth one might expect a homogeneous organism being forced to “chose” between cellular differentiation and increased adaptivity, or continued homogeneity and a lack of adaptivity. Informally I assert the hypothesis: **Cellular differentiation leads to more uniform adaptivity as the organism grows.**

In Chapter 2, I explored the dynamical property of robustness by varying the organism property of homogeneity and heterogeneity while organism size is static. Here, I focus on the dynamical property of adaptivity with respect to the organism property of homogeneity and heterogeneity during growth.

Note that robustness and adaptivity are related properties both in natural and synthetic biology. Biologically, an organism can either be robust to a stimulus or adapt and evolve. There has been research into how an organism can be robust while still evolving and adapting [3].

Related work has involved researching spread [4] and influence [26] in the dynamics

![Figure 3.1: Darwin Finches. Source: [70]](image)
of social networks. In particular, the standing ovation model [45] and similar consensus problems are of interest because the correlation in the dynamics of the organism is a collapse of difference or adaptivity.

### 3.2 Definitions

As already noted there has been research into the hamming distance between functions in an organism [32, 64]. However, here I use the simple count of different functions as a measure of homogeneity and heterogeneity of an organism. Therefore, a homogeneous organism would have 0 different functions, or formally $|\text{Im}(f)| = 1$, while a heterogeneous organism would have at least 1 different function or $|\text{Im}(f)| > 1$. Given these two distinct classes of organisms, the border case is of interest. Specifically, an organism with 1 different function, formally $|\text{Im}(f)| = 2$, where every cell determines its state using the same function except for one cell. I refer to this border case class of organism as **minimal heterogeneous**.

Given the earlier definition of an attractor, I can now define **adaptivity** as the number of attractors $\alpha(X)$ in the dynamics space of an organism. Attractor count is a measure of interest that has been extensively studied [31, 61, 18]. I use the term adaptivity due to its relationship to evolvability, defined by Aldana et al. to mean that an organism can acquire new functions and adapt to new environments [3]. Further, if attractor count is considered memory [29], then organisms with greater memory are more likely to be adaptive to new environments that require shifting memory.

### 3.3 Formal Hypothesis 2

Let $X \in \text{Hom}(n)$. Let $X[k]$ be the organism obtained by mutating $X$ so that one cell now uses function $k$. The minimally heterogeneous organism of size $n$ derived from a homogeneous
organism operating function $i$ at every cell is

\[ MinHet(n, i) = \{X^n_i[k] \mid k = 0, \ldots, 15, k \neq i \}. \]

Applying the definitions, Hypothesis 2 can be posed formally as follows: If $k$ is a class 2 homogeneous organism then $\forall n \geq 2, \exists i \geq 0$ such that:

\[ \text{mean}\{\alpha(X) \mid X \in MinHet(n + i, k)\} > \text{mean}\{\alpha(X) \mid X \in Hom(n + i)\}. \]

The implication of this hypothesis is that if adaptivity is a dynamical property that is selected for then growing homogeneous organism would experience evolutionary pressure towards undergoing cellular differentiation. This implication matches what is apparent in nature: organisms that are larger are also heterogeneous having complex specialized structures formed of differentiated cells.

### 3.4 Methodology

Results from Chapter 2 exhibited a pattern in the number of attractors in homogeneous organisms at sizes $n = 2^i$ where $i \geq 1$. Consequently this experiment will exploit those initial results to explore the pattern in greater depth.

The methodology for testing this hypothesis will entail 3 components: programming, simulation to generate experimental data, and analysis. In this section I explain the program and its ex; data analysis and the related proof are covered in the results section.

#### 3.4.1 Program

In order to test hypothesis 2, the C program described in Chapter 2 that sequentially explores the organism state space to compute the number of attractors is extended here both to
explore larger organisms and to generate data files of dynamics graph with mutation edges that can be rendered using graphviz [15] and charted using gnuplot [41]. In order to explore larger organisms, the program is extended with functions for sampling the state space and estimating attractor counts based upon the sampled state space. Further, since the state space of larger organisms exceeds the memory limitations of C’s unsigned integer type, the program requires the GNU MP big number library [27].

Organisms are defined by the set of functions that determine cell state at each successive time step. The program takes this set of functions defining the organism and a number of initial states $r$ from which to sample the state space. Next, it computes $r$ random Boolean states of the organism of size $n$. From each of these $r$ random Boolean states, the program computes the successor states until it finds an attractor.

Estimating a lower bound of the number of attractors in the state space of an organism of size $n$ is accomplished in a trivial manner: the estimate is simply the number of discovered attractors $\alpha_e(n)$ from the sampled state space using the $r$ random Boolean start states: $\alpha(n) = \alpha_e(n)$. In this experiment $r = 1000$. This estimate serves as a lower bound of the number of attractors. As $r$ grows, $\alpha_e(n)$ approaches $\alpha(n)$.

For the scope of this experiment the lower bound estimate is sufficient. However, assuming that the sampled state is representative of the state space in its entirety the proportion: $\alpha(n) = \frac{2^n \alpha_e(n)}{r}$ can be used to estimate attractor count as well. A third estimate, Capture recapture, is possible if more than single iteration of selecting the random start states is completed. Future experiments will apply the later estimation method for more precise estimations as well as a modified capture recapture technique described in chapter 5.

### 3.4.2 Simulation

As seen in Chapter 2, the dynamics space of homogeneous XOR organisms exhibit a pattern of collapse in adaptivity at sizes that are a power of 2 (see Figures 3.6, 3.8, 3.10 and the chart
in Figure 3.2). Therefore, this experiment will focus on simulating organism sizes $n = 2^i$ where $i \geq 1$. The program fully explores the state space computing the number of attractors $\alpha$ for organism sizes $n \leq 16$ and samples the state space for organism sizes $n > 16$ computing the estimated number of attractors $\alpha_e$.

Sampling is used because complete exploration of the state space of larger organisms is computationally intractable and the results from larger organisms is necessary to demonstrate that the patterns seen continue at larger organism sizes. Using the lower bound estimation of $\alpha$ provides an experimental starting point for comparison of the dynamical property of number of attractors.

Next the program is repeated with each possible minimally heterogeneous organism at sizes $n = 2^i$ where $i \geq 1$. Since there are 16 functions, there are also 16 unique minimally heterogeneous organisms. As with the homogeneous XOR organisms, for sizes $n \leq 16$ the program computes attractor count $\alpha$ and for sizes $n > 16$ the program computes the lower bound estimation $\alpha_e$.

The data from the simulations is used to compare the dynamical property of number of attractors at sizes that are a power of 2.

3.5 Results

The results are broken into three parts: homogeneous organisms, minimally heterogeneous organisms, and a comparison of homogeneous to minimally heterogeneous.

3.5.1 Homogeneous Organisms

The number of attractors found from fully exploring homogeneous XOR organisms of sizes $n = 2, \ldots, 16$ is shown in Figure 3.2. For homogeneous XOR organisms sizes $n = 32, 64, 128, 256$ the lower bound estimation of the attractor count $\alpha_e = 1$ is derived from the $r = 1000$
random initial start states. In other words, only a single attractor was discovered when starting at 1000 randomly selected start states and computing the successor states until reaching an attractor.

In Figure 3.2 the homogeneous XOR organism attractor count $\alpha$ is seen to collapse to a single attractor at sizes $n = 2, 4, 8, 16$. Note, in Figure 3.2 the x-axis is organism size and the y-axis is attractor count $\alpha$ in log scale. Not shown in the figure, the attractor count for homogeneous organisms $n = 32, 64, 128, 256$ also collapses to a single attractor. Figures 3.3 and 3.4 show the homogeneous XOR organism dynamics at organism sizes that are not powers of 2 while Figures 3.6, 3.8, 3.10, and 3.12 show the dynamics of the homogeneous XOR organism when it collapses to a single attractor at organism sizes that are a power of 2.
Figure 3.3: XOR organism size 5. Figure 3.4: XOR organism size 6.

Table 3.1: Update rule 6 = XOR.

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>
<thead>
<tr>
<th>Rule</th>
<th>Boolean Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule 6=</td>
<td>((v_{i-1} \land \neg v_{i+1}) \lor (\neg v_{i-1} \land v_{i+1}))</td>
</tr>
</tbody>
</table>

3.5.2 Minimally Heterogeneous Organisms

Table 3.2 shows the attractor count from the simulation of minimally heterogeneous organisms at increasing sizes \( n = 2^i \) where \( i \geq 1 \). The table consists of the attractor counts from simulating minimally heterogeneous organisms that are labeled by the rule that the single differentiated cell applies. Each column corresponds to one of the minimally heterogeneous organisms and each row corresponds to an organism size that is a power of 2. Note, the table shows only minimally heterogeneous organisms that apply one of the following rules \{1, 3, 5, 7, 8, 10, 12, 14\} because these organisms exhibit an increasing number of attractors. Minimally heterogeneous organisms where the differentiated cell applies one of the following rules \{0, 2, 4, 9, 11, 13, 15\} are not shown in Table 3.2 because they did not circumvent the
collapse to a single attractor [7].

Finding small or specific attractors in the state space dynamics is a difficult problem. Finding singleton attractors that have an attractor length 1 is known to be NP-hard [71, 50, 1]. Another approach to finding attractors would be to construct or select an organism that is known to have the desired dynamics. In [49] algorithms for building networks with desirable dynamical properties are proposed. However, this experiment is exploring the state space of the minimally heterogeneous organisms to identify the attractors rather than selecting an organism with known attractors. Because the approach in this experiment is to sample the space and estimate the lower bound of the number of attractors there is the possibility that not all attractors are discovered. In the case of the homogeneous XOR organism, this possibility will be addressed in the proof that follows the experimental results [7].

Table 3.2 is broken into two parts. The top half, sizes $n = 2 \ldots 16$ show $\alpha$ derived from fully exploring the state space while the bottom half show $\alpha_e$ derived from sampling the state space from $r = 1000$ initial random start states. Enumerating the state space fully for organisms where size $n > 24$ becomes computationally intractable [7].

In Table 3.2 it is clear that as the minimally heterogeneous organisms double in size there is also an increase in the number of attractors. In other words, these organisms do not experience a collapse in their attractor count. Minimally heterogeneous XOR organisms where the differentiated cell applies one of the following rules \{1, 3, 5, 7, 8, 10, 12, 14\} have $\alpha > 1$ and $\alpha_e > 1$ for sizes $n = 2^i$ and $i > 2$.

### 3.5.3 Comparison

The experimental results and Figure 3.2 clearly show that the dynamics of homogeneous XOR organisms exhibit a collapse to a single attractor at organism sizes that are of the form $n = 2^i$. Figures 3.6, 3.8, 3.10, 3.12 show the components of the dynamics of homogeneous XOR organisms at sizes that are a power of 2. In contrast, Figures 3.7, 3.9, 3.11, 3.13 show
CHAPTER 3. HOW ADAPTIVITY DRIVES DIFFERENTIATION

Table 3.2: Table of minimally heterogeneous organism attractor count as size increases by powers of 2. Source [7]

<table>
<thead>
<tr>
<th>Size</th>
<th>Rule 1</th>
<th>Rule 3</th>
<th>Rule 5</th>
<th>Rule 7</th>
<th>Rule 8</th>
<th>Rule 10</th>
<th>Rule 12</th>
<th>Rule 14</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>32</td>
<td>≥530</td>
<td>≥793</td>
<td>≥809</td>
<td>≥505</td>
<td>≥527</td>
<td>≥804</td>
<td>≥806</td>
<td>≥505</td>
</tr>
<tr>
<td>64</td>
<td>≥998</td>
<td>≥1000</td>
<td>≥1000</td>
<td>≥1000</td>
<td>≥998</td>
<td>≥1000</td>
<td>≥1000</td>
<td>≥998</td>
</tr>
<tr>
<td>128</td>
<td>≥1000</td>
<td>≥1000</td>
<td>≥1000</td>
<td>≥1000</td>
<td>≥1000</td>
<td>≥1000</td>
<td>≥1000</td>
<td>≥1000</td>
</tr>
<tr>
<td>256</td>
<td>≥1000</td>
<td>≥1000</td>
<td>≥1000</td>
<td>≥1000</td>
<td>≥1000</td>
<td>≥1000</td>
<td>≥1000</td>
<td>≥1000</td>
</tr>
</tbody>
</table>

Figure 3.5: Homogeneous XOR vs Minimally Heterogeneous. [43]

the components of the dynamics of minimally heterogeneous rule 1 organisms at the same sizes. At organism sizes of the form \( n = 2^i \) where \( i > 2 \) minimally heterogeneous XOR organisms where the differentiated cell applies one of the following rules \{1, 3, 5, 7, 8, 10, 12, 14\} have dynamics that do not collapse to a single attractor.
Figure 3.6: XOR organism size 2. Source [7]  Figure 3.7: Minimally heterogeneous size 2 XOR+Rule 1. Source [7]

Figure 3.5 shows the $\alpha$ for the homogeneous organism for sizes $n = 2, \ldots, 33$ in contrast to the lower bound of expected number of attractors in minimally heterogeneous organisms. The number of attractors for homogeneous organisms of size $n = 21, \ldots, 36$ are taken from the integer sequence enumerated by Wolfram [43]. The number of states sampled is 1000 meaning that the lower bound is necessarily less than or equal to 1000. As a result the true lower bound for expected number of number attractors in the set of minimally heterogeneous XOR organisms applying rules $\{1, 3, 5, 7, 8, 10, 12, 14\}$ is likely much higher since each sampled state leads to its own unique attractor at larger organism sizes. Figure 3.5 shows that deviation from homogeneity can allow an organism to avoid low adaptivity as the organism grows.

These experimental results hold both for the dynamics of fully explored organisms where the size is $n < 20$ and for the sampled dynamics of organisms where size is $20 < n \leq 256$. Note, it is sufficient to have $\alpha_e > 1$ to imply that $\alpha > 1$, but it is not sufficient to have $\alpha_e = 1$ to imply that $\alpha = 1$. As stated earlier, the complexity of exhaustively searching the state space for attractors is not computationally possible for organisms of large size. However, I address the limitation of this experimental result and assumed pattern of attractor count collapsing to a single attractor for homogeneous XOR organisms in the proof that follows.
3.6 The Collapse of Adaptivity at Critical Sizes in Homogeneous Organisms

The following proof was originally published in the Journal of Computer Science and Systems Biology. [7]

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

Structure. We consider organisms whose cellular structure may be modeled as an undirected cyclic graph $C = (V, E)$ of size $n$, whose vertices are considered “cells”, and are enumerated $V = \{v_0, \ldots, v_{n-1}\}$. Each cell $v_i$ in $V$ is connected in cyclic order to two neighbors, so that $E = \{(v_i, v_{i+1 \pmod{n}}) | i = 0, \ldots, n-1\}$.
Microscopic cellular behavior within an organism is modeled by fixing a function $f: V \rightarrow \mathcal{F}$ that assigns to each cell $v \in V$, a function $f(v)$ from $\mathcal{F} = \{g: \{0,1\} \times \{0,1\} \rightarrow \{0,1\}\}$, the set of all binary Boolean functions; note that $|\mathcal{F}| = 2^{2^2} = 16$. 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 1.2 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 $b_0b_1b_2b_3$ 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 $|\text{Im}(f)| = 1$; 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 $s: V \times \mathbb{N} \rightarrow \{0,1\}$ 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:

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

for each $i = 0, \ldots, n-1$ and $t \geq 0$. Informally, the state of the organism’s 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 $s^+(t) = \{v \in V | s(v, t) = 1\}$. 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 $s^+(t) \subset V$.

### 3.6.2 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 p. 60. Next, in Section 3.6.3 (p. 61), we use the decompositions to prove significant results about the dynamics of infinite linear organisms.

We begin with the following definition.

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

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:

$$X = \vec{0}1_01_10_20_30_40_50_61_7\vec{0}.$$ 

Here $\vec{0}$ represents an abbreviation for the left-infinite sequence of 0s (for subscripts decreasing to $-\infty$), while $\vec{0}$ 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 \(\bar{0}\).

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 \((\mathbb{Z}/2\mathbb{Z})^\mathbb{Z}\) 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 \((\mathbb{Z}/2\mathbb{Z})^\mathbb{Z}\) defined as follows. Given two functions \(x, y\) in \((\mathbb{Z}/2\mathbb{Z})^\mathbb{Z}\), the value of \((x \oplus y) : \mathbb{Z} \to \mathbb{Z}/2\mathbb{Z}\) at integer \(i\) in \(\mathbb{Z}\), 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 3.3.

We use Table 3.3 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 $X = \overline{01010213040506170}$ then $\hat{S}X = \overline{01-10011203140607180}$.

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. 52-57), which follow.

**Definition 3.** Let $\hat{S} : (\mathbb{Z}/2\mathbb{Z})^\mathbb{Z} \to (\mathbb{Z}/2\mathbb{Z})^\mathbb{Z}$ be a unary operator defined such that for each function $x$ in $(\mathbb{Z}/2\mathbb{Z})^\mathbb{Z}$, the value of $\hat{S}x : \mathbb{Z} \to \mathbb{Z}/2\mathbb{Z}$ at $i$ in $\mathbb{Z}$ 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 $(\mathbb{Z}/2\mathbb{Z})^\mathbb{Z}$ 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 $(\mathbb{Z}/2\mathbb{Z})^\mathbb{Z}$, and all $i \in \mathbb{Z}$, $\hat{S}(x \oplus y)(i) = \hat{S}x(i) \oplus \hat{S}y(i)$.

**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 $\mathbb{Z}/2\mathbb{Z}$, 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 $\hat{S}$ can be distributed over summands. This idea shall be brought to fruition in Lemma 10.

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

For example, if $X = \vec{0}1_00_10_21_30_40_50_61_70$ and $Y = \vec{0}1_00_11_21_30_41_50_61_70$, then $x(i) = y(i + t)$ where $t = 1$ (for all $i$ in $\mathbb{Z}$), and hence $x$ and $y$ are said to be shift-related. On the other hand, if $Z = \vec{0}1_00_11_21_30_41_50_61_70$, 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 $\mathbb{Z}$. 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 $(\mathbb{Z}/2\mathbb{Z})^\mathbb{Z}$.

*Proof.* Consider functions $x, y, z \in (\mathbb{Z}/2\mathbb{Z})^\mathbb{Z}$. 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 $t_1 + t_2$. \hfill $\Box$

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: $\hat{S}X = \vec{0}1_10_10_01_12_03_14_05_61_70_8\vec{0}$ and $\hat{S}Y = \vec{0}1_00_11_21_30_41_50_61_70_81_9\vec{0}$ 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 $x, y \in (\mathbb{Z}/2\mathbb{Z})^\mathbb{Z}$ and $x \approx y$, then $\hat{S}x \approx \hat{S}y$. 
Proof. If \( x \approx y \), then by Definition 4, there exists \( t \in \mathbb{Z} \) such that \( x(i) = y(i + t) \) for all \( i \) in \( \mathbb{Z} \). By Definition 3, we know that \( \hat{S}(x)(i) = x(i - 1) \oplus x(i + 1) \) and \( \hat{S}(y)(i + t) = y(i - 1 + t) \oplus y(i + 1 + t) \). 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 \((\mathbb{Z}/2\mathbb{Z})^\mathbb{Z}\). Towards this, we introduce the next definition.

**Definition 5.** For each function \( x \in (\mathbb{Z}/2\mathbb{Z})^\mathbb{Z} \), let

\[
\bar{X} \overset{\text{def}}{=} \cdots x(-2) \cdot x(-1) \cdot x(0) \cdot x(1) \cdot x(2) \cdots
\]

be the associated bi-infinite binary (ordinary, non-indexed) string.

While Definition 1 reflects the fact that every function \( x \) in \((\mathbb{Z}/2\mathbb{Z})^\mathbb{Z}\) 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

\[
[\bar{X}] \subset (\mathbb{Z}/2\mathbb{Z})^\mathbb{Z}, \text{ wherein } [\bar{X}] \overset{\text{def}}{=} \{ x_t : \mathbb{Z} \rightarrow \mathbb{Z}/2\mathbb{Z} \mid t \in \mathbb{Z} \}, \text{ where } x_t(i) \overset{\text{def}}{=} x(t + i) \text{ for all } i \text{ in } \mathbb{Z}.
\]

**Lemma 6.** For any bi-infinite binary (ordinary, non-indexed) string \( \bar{X} \), the set \([\bar{X}] \subset (\mathbb{Z}/2\mathbb{Z})^\mathbb{Z}\) 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 \( y \in (\mathbb{Z}/2\mathbb{Z})^\mathbb{Z} \). Then by Definition 4, there exists \( t \) such that \( x_a(i) = y(i + t) \) for all \( i \) in \( \mathbb{Z} \), 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 \( F \subseteq (\mathbb{Z}/2\mathbb{Z})^\mathbb{Z} \) be the set of binary-valued functions on \( \mathbb{Z} \) having finite support; that is, \( x \in F \) iff \( x(i) = 0 \) for all but finitely many \( i \in \mathbb{Z} \).

For example, \( X = \vec{0}1_00_10_21_30_40_50_61_70 \) 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 \( (\mathbb{Z}/2\mathbb{Z})^\mathbb{Z} \setminus F \). 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 \( b, e : F \to \mathbb{Z} \) be defined as follows:

\[
\begin{align*}
  b(x) & \overset{\text{def}}{=} \begin{cases} 
  \min\{i \mid x(i) = 1\} & x \neq \vec{0} \\
  0 & x = \vec{0}
  \end{cases} \\
  e(x) & \overset{\text{def}}{=} \begin{cases} 
  \max\{i \mid x(i) = 1\} & x \neq \vec{0} \\
  -1 & x = \vec{0}
  \end{cases}
\end{align*}
\]

For each function \( x \) in \( F \), the **length** of \( |x| \overset{\text{def}}{=} e(x) - b(x) + 1 \) is taken to be the number of bits in the largest essentially non-zero subsegment of \( X \). Continuing the previous example \( X = \vec{0}1_00_10_21_30_40_50_61_70 \), 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 $\phi$ 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

$$X = X_0X_1 \cdots X_i \cdots X_{n-2}X_{n-1}.$$
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, $x \neq \bar{0}$, 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 $F \subset (\mathbb{Z}/2\mathbb{Z})^\mathbb{Z}$ 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 $e(\hat{S}x) = e(x) + 1$. \hfill \Box$

Given a function $x \in F$, we can decompose its string representation $X$ into $c$ disjoint component strings, where each of the components has $0^r$ as a prefix and suffix. Such a 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 $j = 1, \ldots, c$), and let $\mathcal{P}$ be a partition of the set $\{b(x) - r_1, \ldots, e(x) + r_c\} \subseteq \mathbb{Z}$

$$\mathcal{P} = \{(b_1, b_1 + 1, \ldots, e_1), (b_2, b_2 + 1, \ldots, e_2), \ldots, (b_c, b_c + 1, \ldots, e_c)\}$$

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

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

2. For $j = 1, \ldots, c$
   - $e_j \geq b_j + 2r_j$
   - $x(b_j + r_j) = x(e_j - r_j) = 1$
   - For all $i$ satisfying $b_j \leq i \leq b_j + r_j$ or $e_j - r_j < i \leq e_j$, $x(i) = 0$.

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

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

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

and take $W^j \overset{\text{def}}{=} 0^{r_j} \cdot \tilde{W}^j \cdot 0^{r_j}$. 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 \((b_1, W^1, g_1, W^2, g_2, \ldots, g_{c-1}, W^c)\) as an 
(r, c)-decomposition of \(x\).

To compute, for example, a \((1, 3)\) decomposition of our ongoing example 
\(X = \vec{0}101_001_021_30_40_50_61_7\bar{0}\), we need to provide a size 3 partition of the sequence 
\((-1, 0, 1, \ldots, 7, 8)\) into contiguous integer sequences.

If we take \(\mathcal{P} = \{(−1, 0, 1), (2, 3, 4), (6, 7, 8)\}\), 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 subsegments 
of indices (2, 3, 4) and (6, 7, 8). It follows that \((-1, \overline{0}10, 0, \overline{0}10, 1, \overline{0}10)\) 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 \(X = \vec{0}101_001_021_30_40_50_61_7\bar{0}\) and 
\(Y = \vec{0}11_1020_31_40_50_60_71_8\bar{0}\) introduced subsequent to Definition 4, we see that 
\((-1, \overline{0}10, 0, \overline{0}10, 1, \overline{0}10)\) is a \((1, 3)\)-decomposition of \(X\), while \((0, \overline{0}10, 0, \overline{0}10, 1, \overline{0}10)\) 
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 \(i \in \mathbb{Z}\). If \((b_1, W^1, g_1, W^2, g_2, \ldots, g_{c-1}, W^c)\) is an \((r, c)\)-decomposition 
of \(x\), then \((b_1 + t, W^1, g_1, W^2, g_2, \ldots, g_{c-1}, W^c)\) 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 
\((b_1, W^1, g_1, W^2, g_2, \ldots, g_{c-1}, W^c)\) is an \((r, c)\)-decomposition of \(x\). We refer to
the tuple \((W^1, g_1, W^2, g_2, \ldots, g_{c-1}, W^c)\) as an \((r, c)\)-decomposition of the \(\approx\)-equivalence class \([X]\).

Continuing our example, \((010, 0, 010, 1, 010)\) is a \((1, 3)\)-decomposition of \([X] = [Y]\). We note by definition each \((r, c)\)-decomposition
\((b_1, W^1, g_1, W^2, g_2, \ldots, g_{c-1}, W^c)\) of \(x \in F\) gives rise to a set of functions \(x_j : \mathbb{Z} \to \mathbb{Z}/2\mathbb{Z}\) (for \(j = 1, \ldots, c\)) where

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

satisfying the relation \(x = x_1 \oplus x_2 \oplus \ldots \oplus x_c\). 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 \((W^1, g_1, W^2, g_2, \ldots, g_{c-1}, W^c)\) 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

\[
(0^{r-t} \cdot S^t W^1 \cdot 0^{r-t}, g_1, 0^{r-t} \cdot S^t W^2 \cdot 0^{r-t}, g_2, \ldots, g_{c-1}, 0^{r-t} \cdot S^t W^c \cdot 0^{r-t})
\]

is an \((r - t, c)\)-decomposition of \([S^t X]\).

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

\[
W^j = 0^r \cdot x(b_j + r) \cdot x(b_j + r + 1) \cdots x(e_j - r - 1) \cdot x(e_j - r) \cdot 0^r.
\]
Viewing $W^j \hookrightarrow X$ 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

$$(b_1 - t, 0^{r-t} \cdot S^t \hat{W}^1 \cdot 0^{r-t}, g_1, 0^{r-t} \cdot S^t \hat{W}^2 \cdot 0^{r-t}, g_2, \ldots, g_{c-1}, 0^{r-t} \cdot S^t \hat{W}^c \cdot 0^{r-t})$$

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 $\hat{S}^t x$ and considering it as the basis of an $(r, c)$ decomposition of the $\approx$-equivalence class $[S^t X]$, 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 subsegments of $X$. This will be useful repeatedly in the arguments that follow.

### 3.6.3 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 3.6.4 (p. 65), we use this theorem to prove important results about the dynamics of finite cyclic organisms.

Formally stated:

**Theorem 11.** $\forall i \geq 2 \ S^{2^{i-1}}(10^{2^{i-1}-1}) = 10^{2^{i+1}-1} 1$.

Recalling $S : B \rightarrow B$ from Definition 10, we introduce the following named
The main result proved in this section (Proposition 18) is that for all $i \geq 2$, assertion $\phi_i$ is true. This proof shall proceed by induction, for which the next Lemma provides the base case.

**Lemma 12.** $\phi_2$ is true.

*Proof.* It suffices to show $S^1(10^31) = (10)^31$. Noting that $(01310)$ is a $(1,1)$-decomposition of $[10^31]$, by Lemma 10 we know $(S^1(10^31))$ is a $(0,1)$-decomposition of $[S^1(10^31)]$, and since $S^1(10^31) = 1010101 = (10)^31$, the assertion is proved. \(\square\)

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

*Proof.* Noting that $(00100)$ 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. \(\square\)

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

*Proof.* Since $(0(10)^{k-1}10)$ 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. \(\square\)

**Lemma 15.** If $\forall i > j \geq 2$, $\phi_j$ is true, then $S^{2i-1}(10^{2i-1}1) = 10^{2i+1-1}1$.

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

$$S^{2i-1-1}(10^{2i-1}1) = (10)^{2i-1}1.$$
By appealing to Lemma 14 we evaluate $S^1((10)^{2^i - 1}1) = 10^{2i+1-11}$, which completes the proof.

**Lemma 16.** If $\forall i < x$, $\phi_i$ is true, then $0 < k < x$ implies $S^{2k-3}(10^31) = (10)^{2k-11}$.

**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,

$$S^{2k-3}(10^31) = S^{2k-1-1} \circ S^{2k-2} \circ S^{2k-3} \cdots S^2 \circ S^1(10^2-1).$$

Repeated application of Lemma 15 yields

$$S^{21}(10^{2^2-1}1) = 10^{2^1-1}1$$

$$S^{22}(10^{2^3-1}1) = 10^{2^2-1}1$$

$$S^{23}(10^{2^4-1}1) = 10^{2^3-1}1$$

$$\cdots$$

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

It remains to compute $S^{2k-1-1}(10^{2k-11})$. Since $k < x$, we may assume the inductive hypothesis: $\phi_k$ is true. From this it follows that $S^{2k-1-1}(10^{2k-1-1}) = (10)^{2k-11}$. 

**Lemma 17.** If $\phi_i$ is true $\forall i < x$ then $\phi_x$ is true.

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

$$10^{2^x-1}1 = 10^{2^{x/2}} \cdot 0^{2^{x/2}-11} = 10^{2^{x-1}} \cdot 0^{2^x-11}$$
and thus
\[(0^{2^x-1}10^{2^x-1-1}, 1, 0^{2^x-1}110^{2^x-1-1})\]
is a \((2^x-1, 1, 2)\)-decomposition of \([10^{2^x-1}]\). So, by Lemma 10
\[(S^{2^x-1-1}(0^{2^x-1}10^{2^x-1-1}), 0, S^{2^x-1-1}(0^{2^x-1}10^{2^x-1-1}))\]
is a \((0, 2)\)-decomposition of \([S^{2^x-1-1}(10^{2^x-1})]\). Using Lemma 13, \(S^{2^x-1-1}(1)\) may be re-expressed as:
\[S^{2^x-1-1-2} \circ S^2(1) = S^{2^x-1-1-2}(10^31)\]
Appealing to Lemma 16 we determine that \(S^{2^x-1-1-2}(10^31) = (10)^{2^x-1-1}\). Thus, the \((0, 2)\)-decomposition of \([S^{2^x-1-1}(10^{2^x-1})]\) is in fact
\[(\overline{(10)^{2^x-1-1}}, 1, \overline{(10)^{2^x-1-1}})\].

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

The proof of Proposition 18 is now immediate.

**Proposition 18.** For all \(i \geq 2\), \(\phi_i\) is true.

**Proof.** The base case is given by Lemma 12, and the inductive step by Lemma 17.
Proof. Directly from Lemma 15 where \( i \geq 2 \), applying Proposition 18 shows that \( \phi_i \) is true for all \( i \geq 2 \). \( \square \)

3.6.4 Going from infinite to finite

Suppose now that instead of operating with infinite strings (functions on \( \mathbb{Z} \)), the operation is taking place on a cycle of \( n \) cells numbered 0, 1, \ldots, \( n - 1 \), 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 \( S(0^n) = 0^n \). \( \square \)

**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. \( \square \)

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

We are now ready to prove Theorem 1 (stated originally on p. 46): 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 \( S^{2i-1-1}(1) = (10)^{2i-1-1}1 \), a string of length \( 2^i - 1 \). Thus, at discrete time step \( 2^i - 1 \), the cells of the cycle are in state: \((10)^{2i-1-1}10\), implying that cells strictly alternate as 0, 1, 0, 1, \ldots 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 \( 0^n \) \( \Box \)

We are also ready to prove Theorem 2 (stated originally on p. 46): 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 initial state $X$ composed of any set of $X^i$, $X \rightarrow 0^n$.

Applying Lemma 13, we know $S^2(0^{n-1}1) = 10^31$. 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, the resulting state would be $0^n$ if the only two cells with value 1 collide at the same index. For this collision to occur the number of intervening zeros in Lemma 11: $10^{2^{i+1}-1}1$ must equal $n-1 \mod n$.

Therefore, in order for every state $0^n-11$ to lead to the state $0^n$ for a network of size $n$ the following must be true:

$$2^{i+1} - 1 \equiv n - 1 \mod n$$

$$2^{i+1} \equiv 0 \mod n$$

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

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

\[7\]

### 3.6.5 Summary

A single cell organism has only 1 attractor, and if $\alpha = 1$ then $\rho = 1$. Adaptivity or rather the advantage of increased adaptivity could drive growth. More “advanced” biological organisms have a greater ability to adapt to varied environments or inputs, while simpler
organisms must rely on their robustness. For a single cell organism growing by a single cell there are two possibilities: to grow by an additional cell that is the same and remaining homogeneous or to grow by cellular differentiation and becoming minimally heterogeneous. If the organism becomes a size $n = 2$ homogeneous organism, then when it grows by a single cell the same two types of growth are possible again. This pattern continues as long as the organism remains homogeneous. The experimental results for the limited set of synchronous linear cyclic Boolean organisms suggest that for a homogeneous XOR organism to grow and continue to increase its adaptivity it will need to undergo cellular differentiation and become heterogeneous.

Hypothesis 2 states:

- $\text{mean}\{\rho(X) \mid X \in \text{MinHet}(n+i,k)\} > \text{mean}\{\rho(X) \mid X \in \text{Hom}(n+i)\}$.

Hypothesis 2 suggests that selecting for the dynamical property of adaptivity will drive growth and morphogenesis by cellular differentiation. Proofs 3.6.4 and 3.6.4 verify that homogeneous XOR organisms undergo a collapse of adaptivity during growth when they reach sizes that are a proper power of 2. The experimental results shown in Table 3.2 show that there exists a set of minimally heterogeneous XOR organisms that have increasing number of attractors as the organism increases in size. This empirical increase in adaptivity of minimally heterogeneous XOR organisms coupled with the proofs showing a collapse in adaptivity for homogeneous XOR organisms at sizes that are powers of 2 confirms Hypothesis 2 holds for the limited set of homogeneous XOR organisms size $2, \ldots, 256$.

A formal proof that minimally heterogeneous XOR organisms have increasing adaptivity at increasing sizes that are powers of 2 would prove that Hypothesis 2 holds for all homogeneous XOR organisms. Further, it also remains to be shown that the generalized Hypothesis 2 applies to the other homogeneous organisms.
Chapter 4

How symmetry drives growth

4.1 Introduction

In Chapter 3, adaptivity was explored with the assumption that organisms will grow incrementally by a single cell (from size $n$ to $n + 1$). We saw too that adaptivity of the homogeneous XOR organism is adversely impacted as it doubles in sizes that are powers of 2. However, growth rate is a network property that can be expected to have its own relationship to dynamical properties. If growth rate impacts the dynamical properties, then natural selection might exert evolutionary pressure, making certain types of growth more likely than others.

To understand how growth rate relates to dynamical properties, in this Chapter I introduce the idea of how organism growth could preserve symmetry. Take an organism with obvious symmetry: for example, a starfish, which has clear rotational or radial symmetry. How can this starfish grow while preserving rotational symmetry? There is the possibility to growing by adding another arm. There is also the option of simply extending the length of each arm. Both of these modalities preserve rotational symmetry. This Chapter studies both of these symmetry preserving modalities of growth independently, and in comparison
CHAPTER 4. HOW SYMMETRY DRIVES GROWTH

I also explore how dynamical properties of symmetry can be preserved during growth [5], in order to understand which patterns of growth might be selected for in organisms. Previous work in symmetry of cellular automata has been explored by focusing on symmetry of the functions [33, 62], time symmetry referring to the successor function and forwards and backwards simulation in time [17], an organism’s dynamic state pattern symmetry [58], and symmetry of attractor basins [66].

The firing pattern of neurons have been widely studied [53, 28]. This firing pattern has an analogue synthetic biology: it is the periodicity of cell state in the dynamics of the organisms. Many computational models have been developed to study the significance of the firing rate of specific biological cells with respect to map like representations of space [23].

As biological organisms grow and develop, the cell growth and differentiation is responsible for both developing and preserving symmetry [5]. Here I examine cell dynamics to gain insight into patterns of growth and the impact of differentiation on varying forms of symmetry.

Future work will seek to understand what stimulus might produce changes on a cellular level that facilitate bilateral symmetry in the resulting dynamics of the organism. For example, how do cells of a tadpole know to differentiate in order to achieve bilateral symmetry where left and right limbs mirror each other instead of resulting in reduplication or rotational symmetry where the limbs are attached on the opposite sides of the body?

In Chapters 2 and 3, I take a perspective of the organism as a whole and observe the global dynamics. Since the focus in this Chapter is on symmetry of the organism, I will focus on the cell dynamics as they relate to the global organism dynamics. As before, I will focus only on attractor states since the organism spends the majority of its time in attractor states. Fixing our attention on any one cell of the organism as the organism orbits in a specific attractor, we see the individual cell flips on and off with a period that
divides the length of the attractor. Considering each cell over unbounded time, I compute the periodicity of its binary oscillations as the organism orbits in the attractor. These cell periodicities (or “colors”) may be different for different cells but the different values may be arranged symmetrically around the cyclic organism’s boundary. In this Chapter, I explore the rotational symmetry of cellular colors as a dynamical property in relation to organism growth [10].

Referring back to the example of a starfish and its rotational symmetry, experimentally I test two forms of growth that could preserve that rotational symmetry. The first method of growth is cell by cell elongation of each “arm” simultaneously as seen in the bottom of Figure 4.1. The second method of growth is the reduplication or growth of an entire new “arm” all at once as seen in the top of Figure 4.1. Taking inspiration from the biological organism, a starfish grows through elongation rather than adding another arm; therefore, informally the hypothesis being tested is that growth by segment elongation is more likely to preserve symmetry in dynamical properties than growth by segment reduplication.

4.2 Definitions

An individual cell is assigned a color that is determined by the periodicity of the cell’s state as the organism traverses an attractor. Note that periodicity of cell state is dependent upon

Figure 4.1: Starfish possible growth patterns.
Figure 4.2: The 4096 vertex dynamics graph of a size n=12 homogeneous organism in which all cells operate function 6 (XOR) Source: [10]

the attractor being traversed; as such, for every attractor, each cell can have a different periodicity and hence color. Because the periodicity has an upper bound of the length \( l \) of the attractor and a lower bound of 1 the possible colors for a cell will vary dependent upon \( l \) and its multiplicative factors. For each attractor, when each cell is colored according to its periodicity, the organism itself takes on a cyclic pattern of corresponding colors. This cyclic color sequence may or may not have rotational symmetry.

Using a size \( n = 12 \) homogeneous XOR organism’s dynamics as an example, Figure 4.2 shows the decomposition of the dynamics. The complete dynamics has 6 attractors of length 2 as in Figure 4.2 (left), 60 attractors of length 4 as in Figure 4.2 (center), and 4 attractors of length 1 as in Figure 4.2 (right). In Figure 4.2 the nodes are labeled with the Boolean values of the organism’s cells’ states and directed edges lead from a state \( X \) at time \( t \) to its successor state \( Y \) at time \( t + 1 \). Edges in black connect tributary states. Edges in blue connect attractor states. [10]

In Figure 4.3 on the left, the homogeneous XOR organism of size 12 orbits an attractor of length 4. In Figure 4.3 in the middle, each of the 12 cells’ states is expanded outwards towards infinity as the organism orbits in that attractor of length 4. In Figure 4.3 on the right, the periodicity of the cells of the homogeneous XOR organism of size 12 can be seen as the organism orbits an attractor of length 4, and corresponding colors are assigned to the
organism.

The pattern of colors arranged in a ring can result in symmetry. In particular, the coloring could be rotationally symmetric. In the case of coloring of an organism, rotational symmetry means that after a certain degree of rotation the coloring pattern repeats. The segment size $l$ is the number of cells that when an organism is rotated by $l$ results in the same coloring of the organism. Since rotation by segment size $l$ results in the same coloring, segment size divides organism size. The foldedness $k$ is the number of times segment size divides into organism size $k = n/l$. In an organism trivial rotational symmetry is defined as a segment size $l = 1$ and foldedness $k = n$ because it is the case where all cells have the same periodicity resulting in an organism where all cells have the same color. This thesis is only interested in non-trivial cases of symmetry.

An organism’s chromatic symmetry is defined to be non-trivial rotational symmetry in the coloring of the organism cells as it traverses an attractor. A rotationally symmetric organism has $k$ segments of size $l \geq 1$ where each segment has the same pattern of coloring. In Figure 4.3 (on the right), the coloring of the homogeneous XOR organism of size $n = 12$ as it orbits the attractor of length 4 shown in Figure 4.3 (on the left) exhibits chromatic symmetry with segment size $l = 6$ and foldedness $k = 2$. Therefore, the organism coloring can be rotated from any initial position by a segment of 6 cells to the right or left and the resulting organism coloring will be the same as the initial organism coloring [10].

Growth patterns have been extensively studied [2, 36]. Growth in Chapters 2 and 3 focused on incremental organism growth by a single cell from size $n$ to size $n + 1$. Growth by reduplication is growth by incrementally increasing the number of segments from $k$ to $k + 1$. This increase in segments means that the organism growth is from size $n$ to size $n + l$. Growth by elongation is growth by incrementally increasing the segment length from $l$ to $l + 1$. This increase in segment length means that every segment increases in length by 1 and in turn the organism growth is from size $n$ to size $n + k$. In Figure 4.4 the example of
a homogeneous XOR organism of size $n = 10$ in the orbit of an attractor of length 6 in its

dynamics graph elongation gives a starting point from which the two forms of growth can
be explored. Note, that in the example the chromatic symmetry has segment size $l = 5$ and
foldedness $k = 2$. Figure 4.5 represents growth that occurs by segment elongation from a
size $n = 10$ organism to a size $n = 12$ organism. This growth increases segment size $l = 5$ to
$l = 6$. Figure 4.6 represents growth that occurs by segment reduplication from a size $n = 10$
organism to a size $n = 15$ organism. This growth increases the foldedness $k = 2$ to $k = 3$.
Note that in Figure 4.6 the segment size remains $l = 5$ [10].

4.3 Formal Hypothesis 3

Given organism $X$, $|X| = n$. Let $Att(x) := \{A \mid A$ is an attractor of $X\}$. Given $l/n$, the
segment-$l$ organisms are defined as

$$Sym(X, l) = \{A \in Att(X) \mid X$ has $|X|/l$-fold symmetry in an attractor$$.
Applying the definitions, Hypothesis 3 can be posed formally as follows: Given $n \geq 2$, $l | n$, then:

$$\frac{|\text{Sym}(X_6^{n+\frac{n}{l}}, l + 1)|}{|\text{Att}(X_6^{n+\frac{n}{l}})|} \geq \frac{|\text{Sym}(X_6^{n+l}, l)|}{|\text{Att}(X_6^{n+l})|}.$$  

The implication of this hypothesis is that organisms growing and preserving symmetry have a growth rate that matches the number of segments in the rotational symmetry. This implication matches what is apparent in nature: organisms that have symmetry tend not to grow via reduplication one petal or one limb at a time; rather organisms tend to grow through elongation of their parts.
Figure 4.5: Coloring in orbit of attractor length 4 in dynamics of homogeneous XOR organism of size $n = 12$ Source: [10]

Figure 4.6: Coloring in orbit of attractor length 3 in dynamics of homogeneous XOR organism of size $n = 15$
4.4 Methodology

The methodology is again the same: developing a program, executing the program to simulate and generate experimental data, and finally analyzing the collected data. In this section I explain the program and its execution. Data analysis is covered in the results section.

Simulations for chromatic symmetry require greater computational resources than simply exploring the state space to identify attractors. Consequently, a preliminary selection is made to limit the computational space of the experiment. The selection is made using a shallow sample of the dynamics to select organisms and will constitute the focus of a deeper exploration of the dynamics of the selected organisms from the initial sampling.

Therefore, the selection simulation starts from each random state computing the successor states until discovering an attractor for which it computes: attractor length, periodicity of each cell across the attractor, and chromatic symmetry of the organism across that attractor. The goal of this simulation is to identify sets of organisms that have a greater probability to express non-trivial rotational symmetry across their attractors as the organism increases in size.

Figure 4.7 plots attractor length for homogeneous organisms as their size increases. The y-axis is attractor length in log scale while the x-axis is organism size. The homogeneous rule 6 organism is plotted separately to make clear which homogeneous organism the outlying points are associated with. From Figure 4.7 it is clear that only a subset of homogeneous organisms has increasing attractor lengths and is therefore more likely to exhibit non-trivial chromatic symmetry. Specifically, we see that the homogeneous rule 6 organism stands out with attractor lengths appearing to grow exponentially.

Figure 4.8 focuses on only the two homogeneous organisms (homogeneous rule 6 and rule 7 organisms) that have the greatest corresponding increase in attractor length as organism size increases as shown in Figure 4.7. Figure 4.8 shows the percentage of discovered attractors
that have nontrivial rotational symmetry for the subset of homogeneous organisms that has increasing attractor lengths. Relative frequency is the y-axis and organism size is the x-axis.

Figure 4.7: Attractor lengths of homogeneous organisms of sizes $n = 2 \ldots 50$ Source: [10]

The deeper simulations focus on homogeneous XOR organisms because they have been shown to have a non-zero likelihood to exhibit chromatic symmetry as their size increases from $n = 2, \ldots, 50$ in contrast to homogeneous organisms using different rules to determine their state. Recall that the deeper simulations use $r = 100,000$ random initial start states to sample the state space.

4.4.1 Program

The C program written to explore organism dynamics described in Chapters 2 and 3 is extended here to compute the chromatic symmetry of an organism across an attractor. Because the program computes chromatic symmetry of the organism across discovered attractors, data can be collected without full exploration of the state space of the organism. Note that
Figure 4.8: Estimated Percentage of Attractors with Nontrivial Rotational Symmetry
Source: [10]

regardless of whether the space is being sampled or fully explored chromatic symmetry of
an organism is not an estimation because it is computed across an individual attractor.

In this experiment the growth rate of an organism is increased from incrementing by 1
to increasing by either segment length or segment count. Consequently the program must
be able to explore dynamics of larger organisms resulting in the same obstacle and response
faced in the program for Chapter 3. For larger organisms fully exploring the state space
is computationally intractable. To address this obstacle, the program makes use of the
sampling technique developed for the program in Chapter 3 and requires the GNU MP big
number library [27].

Computational time is another limitation that accompanies the larger state space and
the additional calculations necessary to compute chromatic symmetry of the organism across
each discovered attractor. Therefore, the program is also extended to use distributed pro-
cessing to simulate multiple organisms in parallel via a custom written message passing
interface on a cluster of servers. The data from the distributed processes across the cluster is compressed and stored in a MySQL database for analysis.

The program begins by picking \( r = 100,000 \) random initial start states and computing the successor states repeatedly until an attractor state is discovered. Once an attractor is discovered, the program computes the periodicity of each organism cell across the attractor. The periodicity of each cell \( c_0, \ldots, c_{n-1} \) of the organism of size \( n \) for the attractor \( X \) is computed from iterating through each attractor state of attractor \( X \) starting at an initial attractor state of \( X \) at time \( t_0 \) for the length of the attractor \( L \) and storing the array of states \( \{s\{c_i, t_0\}, \ldots, s\{c_i, t_L\}\} \) for each cell \( c_i \) across the attractor cycle. Periodicity \( p \) is directly computed from each of these arrays such that for each attractor the organism of size \( n \) has an array of periodicity \( p_0, \ldots, p_n \) for the corresponding array of cell states at cell \( c_i \) for each of the \( c_0 \ldots c_n \) cells. Each array \( p_0 \ldots p_n \) corresponds to the chromatic symmetry of the organism across a single attractor. An organism with \( \alpha_e \) discovered attractors will have \( \alpha_e \) arrays of \( p_0 \ldots p_n \).

### 4.4.2 Simulation

Attractor length bounds periodicity; therefore, organisms with attractor lengths of 1 are restricted to the trivial case of chromatic symmetry. By sampling the dynamics of all the homogeneous organism from 1000 random start states for sizes \( n = 2 \ldots 50 \), preliminary data collection of attractor lengths is used to identify organisms whose coloring will be restricted to the trivial case of chromatic symmetry.

This simulation takes the set of organisms selected from the preliminary selection that exhibit both: discovered attractors have increasing length as the organism size increases and the percentage of discovered attractors with nontrivial rotational symmetry is non-zero. Because this simulation is focusing on a subset of organisms from the initial simulation greater computational time can be spent on each individual organism at every size. The second sim-
ulation selects $r = 100,000$ random initial start states for organism sizes $n = 2, \ldots, 50$ and computes: attractor length, periodicity of each cell across the attractor, chromatic symmetry across the attractor, and the segment length of the rotational symmetry for each attractor.

4.5 Results

The results are broken into three parts: selection, reduplication, elongation, and a comparison of elongation to reduplication.

4.5.1 Reduplication

Table 4.1 is divided into 3 sections. Here I focus on the first two sections: Precondition and Reduplication. Columns for the precondition are sequentially as follows: organism size $n$, segment length $l$, the number of attractors $\alpha_e$ discovered by sampling the state space from $r = 100,000$ random initial states, and percent of attractors discovered with rotational symmetry with segment length $l$. Columns for Reduplication are sequentially as follows: organism size after growth $n + l$ and percent of attractors discovered after growth with rotational symmetry with segment length $l$.

From the Table 4.1’s section Precondition and Reduplication sections it is clear that during growth by reduplication it is possible though rare for homogeneous XOR organisms to maintain the dynamical property of chromatic symmetry.

4.5.2 Elongation

In Table 4.1 is divided into 3 sections. Here I focus on the first and last section: Precondition and Elongation. Columns for the Precondition are sequentially as follows: organism size $n$, segment length $l$, the number of attractors $\alpha_e$ discovered by sampling the state space from $r = 100,000$ random initial states, and percent of attractors discovered with rotational symmetry
with segment length $l$. Columns for Elongation are sequentially as follows: organism size after growth $n + n/l$ and percent of attractors discovered after growth with rotational symmetry with segment length $l + 1$.

From the Table 4.1’s section Precondition and Elongation sections it is clear that during growth by elongation it is possible and not uncommon for homogeneous XOR organisms to maintain the dynamical property of chromatic symmetry.

4.5.3 Comparison

Here I compare the results of the probability of preserving symmetry for each growth pattern (elongation and re-duplication) from Table 4.1. The comparison is made directly between the columns for percent of attractors discovered with rotational symmetry with corresponding segment lengths for both the Reduplication and the Elongation sections. The percent of attractors discovered with rotational symmetry with corresponding segment length that are non-zero occurs more for the section Elongation than for the section Reduplication. Therefore, growth by elongation is more likely to preserve rotational chromatic symmetry.

To visualize the data from the table, Figure 4.9 compares chromatic symmetry after growth for organisms at the same size. Note that at each size multiple data points are possible because each segment has its own percentage. For example, the organism of size $n = 24$ has three different segment sizes, and in the case of elongation segment size $l = 12$ has the highest percent (22%) of rotational symmetry found out of the 3 possible growths. The alternative growths maintaining rotational symmetry for organisms of size $n = 24$ include segment sizes $l = 6$ and $l = 3$. Growth maintaining rotational symmetry for these segment lengths is 9% and 0% respectively. Figure 4.9 makes it apparent that growth through elongation has a greater likelihood to maintain chromatic symmetry than growth through reduplication. The non-zero percentages for growth by reduplication at sizes 15 and 26 indicate that although elongation may be more likely preserve to symmetry at certain sizes, growth by reduplication
Table 4.1: Results of testing growth by reduplication and by elongation. Source: [10]

<table>
<thead>
<tr>
<th>n</th>
<th>l</th>
<th>α_e</th>
<th>% Sym w Seg = l</th>
<th>n + l</th>
<th>% Sym w Seg = l</th>
<th>n + n/l</th>
<th>%Sym w Seg = l + 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>3</td>
<td>10</td>
<td>60</td>
<td>9</td>
<td>0</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>46</td>
<td>98</td>
<td>15</td>
<td>1.8</td>
<td>12</td>
<td>34</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>70</td>
<td>34</td>
<td>18</td>
<td>0</td>
<td>14</td>
<td>78</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>79</td>
<td>0.26</td>
<td>15</td>
<td>0</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>7</td>
<td>298</td>
<td>0.78</td>
<td>21</td>
<td>0</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>5</td>
<td>1112</td>
<td>0.019</td>
<td>20</td>
<td>17</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>9</td>
<td>4707</td>
<td>0.57</td>
<td>27</td>
<td>0</td>
<td>20</td>
<td>12</td>
</tr>
<tr>
<td>18</td>
<td>3</td>
<td>4707</td>
<td>0.0011</td>
<td>21</td>
<td>0</td>
<td>24</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>10</td>
<td>932</td>
<td>0.12</td>
<td>30</td>
<td>0</td>
<td>22</td>
<td>31</td>
</tr>
<tr>
<td>20</td>
<td>5</td>
<td>932</td>
<td>0.17</td>
<td>25</td>
<td>0</td>
<td>24</td>
<td>15</td>
</tr>
<tr>
<td>22</td>
<td>11</td>
<td>974</td>
<td>0.31</td>
<td>33</td>
<td>0</td>
<td>24</td>
<td>15</td>
</tr>
<tr>
<td>24</td>
<td>12</td>
<td>930</td>
<td>0.15</td>
<td>36</td>
<td>0</td>
<td>26</td>
<td>22</td>
</tr>
<tr>
<td>24</td>
<td>6</td>
<td>930</td>
<td>0.15</td>
<td>30</td>
<td>0</td>
<td>28</td>
<td>9</td>
</tr>
<tr>
<td>24</td>
<td>3</td>
<td>930</td>
<td>0.022</td>
<td>27</td>
<td>0</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>13</td>
<td>998</td>
<td>0.22</td>
<td>39</td>
<td>14</td>
<td>28</td>
<td>1.8</td>
</tr>
<tr>
<td>28</td>
<td>7</td>
<td>1000</td>
<td>0.09</td>
<td>35</td>
<td>0</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>28</td>
<td>14</td>
<td>1000</td>
<td>0.01</td>
<td>42</td>
<td>0</td>
<td>30</td>
<td>1.6</td>
</tr>
</tbody>
</table>

can preserve symmetry and may be the only way to preserve symmetry.

4.5.4 Summary

When considering growth in organisms there are different possibilities for the rate of growth. In Chapters 2 and 3, the growth rate is 1, meaning that organisms grow by a single additional cell. In this Chapter, the growth rate is by rotational segment size $l$ or by number of rotational segments $k$ where $k = n/l$. The experiment in this chapter assumes that organisms are growing in size and tests which of the growth rates an organism could take to preserve rotational chromatic symmetry.

Returning to the example of a starfish and its growth: does a starfish maintain its rotational symmetry by growing a new limb or by each limb growing in size concurrently?
In nature there are many examples of organisms growing by elongation, that is by increasing in size of their limbs or petals. Consequently, Hypothesis 3 states that:

\[
\frac{|\text{Sym}(X_6^{n+l}, l+1)|}{\text{Att}(X_6^{n+l}, l+1)} \geq \frac{|\text{Sym}(X_6^{n+l})|}{\text{Att}(X_6^{n+l})}
\]

The experimental results in Figure 4.9, with a couple of exceptions, support Hypothesis 3 for the limited set of synchronous linear cyclic Boolean homogeneous XOR organisms sizes 6, \ldots, 28 suggesting that growth by segment elongation would be selected over growth by segment reduplication to preserve rotational chromatic symmetry. The exceptions and limited set of organisms sampled indicate that further experiments are needed before Hypothesis 3 could be considered for the broader set of all organisms. However, empirically for the sampled subset of homogeneous XOR organisms, selection for preservation of the property of rotational chromatic symmetry will drive growth by elongation.
Chapter 5

Obstacles

5.1 Introduction

The experimental approach described in earlier chapters faces obstacles that accompany big data. As the size of a network grows linearly, both the state space and the number of possible organisms grow exponentially. Consequently, even at relatively small network sizes, exhaustive search of all possible organisms and their state space becomes computationally intractable. These limitations restrict the experiments to sampling techniques and subsets where the results are estimations instead of certainty.

The experimental restrictions arising from these computational limitations can be seen in the earlier chapters: In Chapter 2, the simulation is limited to the homogeneous organisms and sampling of the set of possible heterogeneous organisms. Further, the experiment is limited to sizes $n = 2, \ldots, 16$ in order to fully explore the state space of the simulated organisms. In Chapter 3, the organisms being simulated are further restricted to the subset of homogeneous XOR organisms and minimally heterogeneous organisms. Since the experiment includes sizes $n = 2, \ldots, 256$ (focusing on sizes that are powers of 2), the state space is fully explored for sizes $n \leq 20$ and sampled for sizes $n > 20$. In Chapter 4, the organisms...
being simulated initially are limited to the set of homogeneous organisms sizes \( n = 2, \ldots, 50 \) and the state space of the simulated organisms is initially sampled from 1000 random states. Next, the set of homogeneous XOR organisms of sizes \( n = 2, \ldots, 50 \) are sampled from 100000 random states. Finally, the set of minimally heterogeneous XOR organisms using NXOR at the differentiated cell are simulated for sizes \( n = 6, \ldots, 30 \) from attractor states of the homogeneous XOR organism that have non-trivial rotational chromatic symmetry.

An organism’s successor state \( S(X) \) is computed from a given organism’s state \( X \) by computing the next state of each cell of the organism given the current state of each cell’s neighboring cells. In order to find an attractor, the new state is compared to all previously discovered states to identify if a cycle has been encountered. If there is no match, the successor state is stored and the successor of the successor state is computed. This process is repeated until an attractor is found.

Identifying an attractor in the state space of an organism in the best case involves storing a single state, computing the successor state once, and making a single comparison. In the worst case, for an organism of size \( n \) there are \( 2^n \) states in the attractor and the above process stores \( 2^n - 1 \) states, computes \( 2^n \) successor states, and compares \( (2^n - 1 \times 2^n - 1) + 1 \) states. Clearly as the organism size \( n \) grows linearly, the number of states being stored and compared grows exponentially and becomes rapidly computationally limiting.

Computing organism color across an attractor and attractor robustness require that attractors be identified while organism robustness and attractor count for an organism require that all attractors be identified. Therefore the candidate approaches will address attractor computation, limited data resulting from state spaces that can not be fully explored, and techniques for exploring organism state space in parallel.

In the following sections I will elaborate on three approaches applied in the programs to mitigate big data obstacles. First, I will explore algorithms to increase efficiency in memory and search by computing, storing, and comparing the successor state at multiplicative or
exponential time increments instead of single increments. Second, I will explore sampling techniques to estimate properties of the dynamics and the organism. Third, I will use distributed processing to explore the state space and simulate sets of organism dynamics in parallel.

5.2 Trading time for space

An improvement to the trivial method of storing and comparing every state is instead to store every $2^i$th state. As such, the memory of states required in the worst case decreases for organisms of size $n$ from the exponential $2^n$ to the linear $\log_2 2^n$ or $n$.

Let $P$ be the number of states from an initial state $X$ to the first state in an attractor and let $L$ be the length of the attractor that $X$’s successor states lead to. Then the number of states stored is $\log P$ and the number of comparisons to discover the attractor is $O(P \log P)$.

Note, in the case that the path to the attractor (number of successor states before the first state in the attractor) is long and the attractor is short, if the first state in the attractor is just after a power of 2 then this could result in looping through the attractor until the next power of 2 is reached and the state in the attractor is stored for comparison. However, going by worst case, the improvement is significant from $O(2^{n^2})$ to $O(n \log n)$.

Code 5.1: Attractor computation storing states at powers of 2

```plaintext
// given a state S, size N, array of int functions of length N, return
// an array attractor name, attractor length
// store only every state that is a power of 2 — reduce comparisons,
// reduce memory storage
int attractor (UOID, state, name, length)
{
    found = 0;
    do{
        if (powerOf2(counter)) {
            append(path, state);
        }
        counter++;
        nextState = successorOrganismState(UOID, state);
        if (isInPath(path, nextState)) {
            found = 1;
        }
    } while (found == 0);
}
```
For XOR organisms, I developed an algorithm for simulating more than one successor step at a time with a goal of computing forward in powers of 2 successor steps at once. The basis for this algorithm is the pattern observed in the results for Chapter 2 and later proved in Chapter 3. The core component of this algorithm relies on the proof that organisms of size $n$ is a power of 2 in any state will collapse to a single attractor after a set number $C$ of successor computations.

The algorithm takes the initial start state and decomposes the start state into a set of states each with only a single cell in the state 1 where the remaining cells are in state 0. From Lemma 3 it is clear that the successor states can later be recombined or merged using the $\oplus$ or XOR operation resulting in the recomposed organism successor state. Next, it is known from Lemma 15 that each of the states resulting from the decomposed state has a known successor state at each power of 2 steps in the infinite case. Lemma 13 gives the base case of $S^2(1) = 10^31$ and the general case is given by Lemma 15 as: $S^{2^{k-2}}(10^{2^{k-1}-1}) = 10^{2^{k-1}-1}$.

Next, the algorithm breaks down the number of successor steps to compute forward into a series of powers of 2 starting at the largest power of 2 and continuing until the smallest remaining power of 2. Note, a simple way to think of this is as the binary representation of the integer number of steps. For example 39 would be 101001. This would mean that first compute the successor state after $2^5$ steps, next compute that state’s successor after $2^3$ steps, and finally compute the immediate successor after $2^0$ steps. In other words, instead of computing the successor state in 39 steps the algorithm computes successor state in 3 steps.
(not including the decomposition and recomposition of composite states along the way).
Therefore the new algorithm operates in $\log_2 C$ steps each of which is a power of 2 instead of $C$ steps of size 1 that the sequential successor algorithm uses.

Code 5.2 takes as its input the following three values: an index value $index$ for the cell that has state 1 in an organism state where all but that cell have the state 0, the number of steps $K$ to compute forward, and the organism size $n$. The output of this function is the organism state at the closest power of 2 steps less than $K$.

Code 5.3 takes as its input an organism state $organismState$, a number of steps $sSteps$, and organism size $n$. This function recursively calls itself to compute the state by decomposing the number of steps into powers of 2. The organism state is also decomposed into its composite parts of organism states that are made up of all cells in 0 state except for one cell with the state 1. The composite organism states are used as inputs to Code 5.2 to compute the organism state at each successive power of 2 that add up to $sSteps$. The results are merged and this is repeated until a single organism state is returned.

**Code 5.2: Compute successor in steps of powers of 2**

```c
int *computeSuccessorP2(int index, int K, int N)
{
    // note the K being passed should be the exact K from Lemma 12!
    // computes organismState using lemma 12 and returns that state using index to keep track of where the 1’s will be
    int 2K = power(2, K);  // 2^K
    int DIV = divide(2K, 2);  // divide by 2 since we are moving from our center index equally
    int RI = (DIV + index) % N;  // right index shifted to the right of our index by DIV mod n to wrap around our organism
    int LI = (index - DIV) % N;  // left index shifted to the left of index by DIV mod n to wrap around our organism
    organismState[RI] = 1;
    organismState[LI] = 1;
    return organismState;
}
```

**Code 5.3: Compute successor in steps of powers of 2**

```c
int *computeSuccessor(organismState, sSteps, N)
{
    // takes an organism state and the number of successor steps to compute forward
```
5.2.1 Limitations

Storing and comparing every $2^i$th state improves the memory required for identifying attractors and has the potential to improve the number of operations (or time) required to identify attractors. However, this improvement does resolve the exponential increase in state space and consequently the number of successor operations that need to be computed to identify
an attractor or to explore the state space fully. The algorithm that computes successor operations in powers of 2 does address how many successor operations are needed to identify an attractor. However, this algorithm does not change the number of successor operations that are required to explore the state space fully and it is restricted to only being applicable for homogeneous XOR organisms. As organisms grow in size linearly their state space grows exponentially. Further, if attractor length grows exponentially then computing the organism color across the attractor will still be computationally intractable regardless of successor operations in powers of $2^i$. Though these limitations pose restrictions for future work and experiments, the approaches taken suggest that remediation for the computational and memory limitations to related big data obstacles is possible.

## 5.3 Trading accuracy for time

Sequential enumeration of the complete state space for larger organisms is computationally intractable in the experimental approach. I address this obstacle through random sampling of the state space to produce estimations of the properties of the dynamics.

Random sampling is accomplished by picking a random state between 0 and $(2^n) - 1$ instead of sequentially starting at each subsequent state in the state space. Each random state’s successors are then simulated until an attractor is discovered. Attractors are determined, in the same manner as with sequential starting states, by computing the successor states from the initial state until a previously seen state is seen again.

The random sampling function in the program takes as its input a number of random states $d$ to generate. Next, the function computes the maximal state size based upon the organism size $n$. The random states are generated using the GNU Multiple Precision Arithmetic Library function call `mpz_urandomm`. The function `mpz_urandomm` requires a seed and the maximum that in this case is $2^n$ and returns a value less than the maximum [27].
Next, each computed random state is used to discover an attractor. The program keeps track of the number of random initial start states that lead to an attractor and the value is stored in the database table for each attractor. In other words, each unique attractor $U\text{AID}$ in the attractor table has a cumulative value $R\text{DC}$ representing the total number of random initial start states that when simulated forward lead to that $U\text{AID}$.

Experimentally, the program primarily computes the following properties of an organism: $\alpha$ as the number of attractors and $\rho$ as the organism robustness. Therefore, when randomly sampling the state space, there must be a correlating estimation $\alpha_e$ and $\rho_e$. However, the program also collects data regarding the length of each attractor $L$, the number of random initial start states $R\text{DC}$ that lead to each attractor, and the total number of random initial start states generated for an organism $T\text{RDC}$. Sampling can occur sequentially as well, and the program keeps track of the total number of initial sequential states.

The sequential sampling function in the program takes a number of states $d$ to generate as its input. These states are generating incrementally starting from the last sequential state explored $S\text{DC}$, which is stored in the database. The value $S\text{DC}$ is a cumulative value starting from 0 adding $d$, where the total possible states in the state space $2^N - 1 = P\text{SC}$ and $S\text{DC} \leq P\text{SC}$. Next, each sequential state is used to discover an attractor. The program keeps track of the number of initial start states that lead to an attractor and the value is stored in the database table for each attractor. In other words, each unique attractor $U\text{AID}$ in the attractor table has a cumulative value $S\text{DC}$ representing the total number of sequential start states that when simulated forward lead to that $U\text{AID}$.

From the above described sampling approaches I apply the following two estimations for the size of the basin of attraction for an attractor: sequential sampling,

$$E\text{BS} = (S\text{DC}/P\text{SC})(2^n),$$
and random sampling,

\[ EBS = \frac{RDC}{TRDC} \left( \frac{2}{n} \right) \].

Certainty \( C \) for sequential sampling is computed as \( C = \frac{PSC}{2^n} \) and for random sampling \( C = \frac{TRDC}{2^n} \).

For capture recapture, the program randomly samples a number of states \( d \), twice during 2 separate intervals. The set of attractors identified from the first set of \( d \) starting states are labeled as Capture 1 \( C_1 \). The set of attractors identified from the second set of \( d \) starting states are labeled as Capture 2 \( C_2 \). And, the set of attractors seen in both Capture 1 and Capture 2 are labeled as Recapture \( R \). There are a variety of capture recapture models that are commonly used for population estimation. Of these, I started with the Lincoln-Petersen method [56]. The Lincoln-Petersen method is derived from the proportion of recaptured entities in the second iteration of capture; \( R/C_2 \), should be equal to the proportion captured in the first iteration divided by the total population, \( C_1/N \). From this proportion the estimated total population \( \alpha_e \) can be solved for: \( (C_1 \times C_2)/R = \alpha_e \) [44, 48, 34].

Capture Recapture has limitations that occur as the size of the state space and the number of attractors increase to such a degree that the sampled Capture 1 \( C_1 \) and Capture 2 \( C_2 \) have no overlap. If \( R = 0 \) in the Lincoln-Petersen method then solving for \( n \) gives: \( \alpha_e = (C_1 \times (C_2))/0 \). A further limitation of Capture Recapture is the assumption that the size and distribution of entities are the same. As seen in earlier figures not all basins of the attraction are the same. An example of this would be if trying to capture animals from a diverse set of animals. In a mixed population the larger or slower animal might be easier to mark or capture and a smaller or faster animal might be harder to mark or capture. The resulting captures will contain a greater number of the slower or larger animals. This imbalance will skew the proportion and in turn the estimated population size \( \alpha_e \).

Therefore, I developed a capture recapture technique that takes into account the varying
attractor sizes to estimate attractor counts of differing sized attractors and the total number of attractors in an organisms dynamics space. I use attractor length $L$, which is computed and stored in the database attractor table for each unique attractor $UAID$ of each unique organism $UOID$ as a way to distinguish attractor type.

I propose the two techniques to below to be used to estimate the number of attractors $\alpha_e$. In future experiments the techniques will be compared to identify how well they perform in relation to existing estimation techniques including the Lincoln-Petersen method and in relation to known population results to measure their accuracy. This experiment can be conducted using the existing program and multiple intervals. The experiment should also explore the minimal sample size necessary for the results of each technique to fall into the desired confidence interval.

### 5.3.1 Technique 1

The first technique is derived from the relationship of the number of sampled states to the total state space. It estimates number of attractors using a single interval and cumulative discovered basin of attraction size. This technique assumes that the average basin of attraction size in the sample space is the same as the average basin of attraction size for the entirety of the state space. Let $d$ be the number of random start states sampled from a single interval, $d_i$ be the number of random states that are in the basin of attraction for attractor $i$, $k$ be the number of distinct attractors discovered from the $d$ random start states, and $A_i$ be the basin of attraction for attractor $i$. Given the total number of states in the state space is $2^n$ for organisms of size $n$ and the assumption that the space sampled is representative of the entire state space, $\alpha_e(\sum_{i=1}^{k} A_i) \approx 2^n$. Solving for $\alpha_e$,

$$\alpha_e \approx \frac{2^n}{\sum_{i=1}^{k} A_i}$$
Next I use confidence intervals to determine a lower bound with 95% confidence for the number of attractors of type $i$ as:

$$2^n \left( \frac{d_i}{d} \right) \left[ 1 - 1.96 \sqrt{\frac{d_i d(1-d_i)}{d^2}} \right]$$

and upper bound:

$$2^n \left( \frac{d_i}{d} \right) \left[ 1 + 1.96 \sqrt{\frac{d_i d(1-d_i)}{d^2}} \right].$$

The total number of attractors of types $i = 1, \ldots, k$, assuming that these are the only types of attractors in the state space, can be bounded by summing across all attractors $1 ldots, k$ such that the lower bound is:

$$\sum_{i=1}^{k} \left( 2^n \left( \frac{d_i}{d} \right) \left[ 1 - 1.96 \sqrt{\frac{d_i d(1-d_i)}{d^2}} \right] \right)$$

and upper bound:

$$\sum_{i=1}^{k} \left( 2^n \left( \frac{d_i}{d} \right) \left[ 1 + 1.96 \sqrt{\frac{d_i d(1-d_i)}{d^2}} \right] \right).$$

### 5.3.2 Technique 2

The second technique is derived from Lincoln-Petersen method but applied to each type of attractor where type is determined by attractor length $L$. The technique estimates $\alpha_e$ using two intervals of capture and recapture and the respective estimated basin of attraction sizes for each attractor type. Let $\sum_{i=1}^{k} A_i^1$ be the total number of interval 1, $\sum_{i=1}^{k} A_i^2$ be the total number of interval 2, and $\sum_{i=1}^{k} R_i$ be the number recaptured. Then

$$\frac{\sum_{i=1}^{k} A_i^1}{\alpha[\sum_{i=1}^{k} A_i^1 + \sum_{i=1}^{k} A_i^2]} = \frac{\sum_{i=1}^{k} R_i}{\sum_{i=1}^{k} A_i^2}.$$
and

\[
\alpha \left( \sum_{i=1}^{k} A_{1i}^1 + \sum_{i=1}^{k} A_{1i}^2 \right) < 2^n.
\]

### 5.3.3 Limitations

A major drawback to estimation techniques that rely on estimates by attractor type is that bigger sample sizes are necessary to ensure a recapture of each attractor type occurs. If no recapture for an attractor type is discovered then the estimation will be off. Because state space grows exponentially as organisms grow linearly, in many cases estimation techniques that sample in proportion to the state space only serve to provide estimations for organism sizes slightly larger than organisms that can be fully explored. Future work will entail identifying the relationship of sample size relative to state space or population size and looking for other techniques to improve accuracy while minimizing the required sample size. Future work would also involve an experiment measuring the accuracy of basin size estimation.

### 5.4 Parallel distributed processing

Serial simulation of organism states to explore the full state space by the program is computationally too slow to explore larger organisms and multiple organisms. Instead of simulating the state space dynamics of each organism one at a time, the processes can be distributed within a cluster such that each client simulates a single organism’s dynamics. However, even distributing a single organism’s dynamics to be computed by a single client can be too time intensive. The distributed processing that I implemented can also break up the state space to explore and distribute the simulation of a single organism’s state space between multiple cluster clients. In order to break the state space into parts that can be distributed as jobs to clients, there needs to be tracking of for states that have not been explored yet.
Consequently, the tracking and data storage of the computed state space are centralized in a MySQL database while the jobs are distributed via a message passing interface to client machines in server cluster environment.

Processing requests to the centralized server consist of an organism id and a number of states to process sequentially or randomly. In the case of the sequential request, the server queries the database to look up the last state explored for that organism id. Next the server breaks the requested processing steps up into equal sizes starting from the last sequential state explored. The resulting start and end states are then pushed into the client job pool such that each job has the organism id, start state, and end state. As the jobs are pushed to the client job pool, the server updates the database to reflect the sequential state that will be reached to avoid duplication of job requests. In the case of the random sampling request, the server splits the number of steps to process into equal sizes, which are then pushed into the client job pool such that each job has the organism id and a number of random states to start computing from. As the jobs are pushed to the client job pool, the server updates the database to reflect increases in the number of random states being sampled.

Cluster clients query the centralized server’s client job pool for available jobs. Sequential jobs consist of a start state, an end state, and the organism id. Random sampling jobs consist of a number of initial states to sample and an organism id. Clients then simulate forward from each initial state (sequential or random) until they find an attractor. Next they compute attractor length, attractor robustness, and chromatic symmetry. For each sequential state that lands in an attractor there is an incremented field and for each random state that lands in an attractor there is also an incremented field. The result is a series of entries for each attractor discovered sequentially or randomly consisting of that attractor’s name, length, robustness, chromatic symmetry, and number of initial states that lead to it. Finally the set of all entries is pushed back to the centralized server’s server job pool.

The server pulls the client entries from the server job pool as either sequential job results
or random job results. Next the server compresses and bulk inserts these entries according
to whether they are random sampling results or sequential results to a centralized MySQL
database that keeps track of the states explored.

5.4.1 Limitations

While the distributed processing alleviates some of the computational time, a bottleneck
occurs at the centralized database because there is a centralized server doing bulk inserts
derived from many cluster clients to write to a singular MySQL database. This bottleneck
can be clearly seen in Figure 5.1. However, this structure was selected to ensure unique values
for indexing and to avoid collisions in processing. Future work would be to decentralize the
database to improve access time for database read and write operations.
Pull sequential data from MySQL
- Create jobs from MySQL data
- Distribute jobs to clients
- Compress MySQL results from clients
- Push compressed data to MySQL

Cluster Clients ←→ Message Passing Interface Server ←→ MySQL Database

Figure 5.1: Distributed Processing in Cluster.
Chapter 6

Software

6.1 Introduction

Software development was an integral part of the experimental approach. The software to simulate organisms and their dynamics was written to test hypotheses described in this thesis. Programs were designed to simulate networks given minimal input parameters, store the resulting data in a database, and create output files that could be rendered using graphviz or gnuplot. The decision to develop new software instead of relying on existing software to conduct the experiments was to facilitate an easily customizable code base for the specific experiments being conducted and to be adaptable for different architecture requirements later on (for example, a distributed database). In Chapter 5 I describe some of the obstacles faced when simulating network dynamics and approaches that were taken to mitigate those obstacles. In this chapter I discuss the structure, libraries, and some example usage.

Note, that while the choice was not to use existing software to simulate the networks there are existing libraries that were taken advantage of to simplify the development process. The program is written in C with the intention to minimize both memory overhead and processing overhead in comparison to other high level languages that were considered (for
example, Java). The program takes advantage of the existing libraries to handle input, output, mathematics, and notably the GNU MP Bignum Library [27] to work with arbitrarily large numbers resulting from exponential growth in the state space.

6.2 Structure

By developing software libraries and using modular programming, the result is a series of programs and routines that can be easily swapped without needing to rebuild the entire code base. Each experiment is a singular command or set of instructions issued by a simple script. Figure 5.1 shows the infrastructural layout. The software structure will be expanded upon here.

6.2.1 Client Daemon

A client daemon runs on each client server in a server cluster environment. The client daemon uses rsync to maintain a copy of the client job pool from the server. In the job pool: open job files use a prepended $o$ and have name format:

$$o.$CLIENT.task.parameter1.parameter2...parameterN$$

processed jobs use a prepended $p$ and have the format:

$$p.o.$CLIENT.task.parameter1.parameter2...parameterN$$

and completed jobs use a prepended $c$ and have the format:

$$c.o.$CLIENT.task.parameter1.parameter2...parameterN.$$
Each client $CLIENT selects an open job that has the matching prefix $CLIENT to execute. The job name itself is the command to execute including the input parameters. After selecting a single open job, the daemon executes the command task with the designated input parameters and updates the open job file from the pool to a processed job. Next the daemon waits for the task process to complete before marking the job file completed. Finally the client daemon rsyncs the job pool with the server. The data from a job completed by client $CLIENT is output to csv files locally on that client.

6.2.2 Server Daemon

A server daemon runs on a single server connected to the cluster of client servers by a switch. MySQL runs locally on the same centralized server to further reduce latency. The server daemon keeps track of the total number of active client servers and their respective $CLIENT identifiers. The daemon loops through the following steps:

- looks for completed generate organism jobs and imports the corresponding org.csv files, and update the jobs as imported by prepending $i$ with the format:

  \[ i.c.o.$CLIENT.task.parameter1.parameter2...parameterN \]

- looks for completed generate attractor jobs and imports the corresponding att.csv files, and update the jobs as imported by prepending $i$ as with organism jobs.

- looks for completed generate attractor data jobs and imports the corresponding att-data.csv files, and update the jobs as imported by prepending $i$ as with organism jobs.

For each of these steps, the server daemon looks for all jobs of the same type (organism, attractor, or attractor data) that have completed. Next, rsync is called to pull all correlated output csv files for those jobs from the client servers where those jobs were executed. Finally,
the server daemon calls a script to compress and bulk insert the data from those csv files. For each type of job there is a related script for compression and bulk insert:

- importocsv.sh compresses and bulk imports organism exploration state results
- importsacsv.sh compresses and bulk imports sequential attractor results
- importracsv.sh compresses and bulk imports randomly sampled attractor results
- importadcsv.sh compresses and bulk imports attractor data results

Note, the scripts differentiate between sequential and random sampling. There are neither client nor server jobs related to organism data computation since this mainly requires reading from the database and with a singular database does not benefit from distributed processing.

### 6.2.3 Database

The choice of a MySQL database was made due to the requirement of having uniquely indexable fields without duplication and to take advantage of the open source platform. Having unique id’s is advantageous if the environment is further distributed to other servers or job submission is opened to the public. In this manner tasks can pick up from where they were left off without repeating previously executed simulations. The ideal result is a consistently growing shareable database containing dynamical properties of the organisms being studied.

The database structure is broken down into the following four tables: an organism table, attractor table, attractor data table, and organism data table. Table 6.1 shows the tables with respective list of tuples showing the table field and type. Note, in each of the tables, ID is an auto incremented integer value to make sure there exists a unique indexing for each entry in the table. Due to limitations on integer size some fields are stored as text representations of either binary, decimal, or hexadecimal values.
In the organism table, UOID is a unique organism ID. The leading character of every UOID is 1, and the complete string for an organism of size \( n \) is \( n + 1 \) characters long. Each character after the leading 1 of the UOID is a hexadecimal value that represents a successor function. The position of the character corresponds to the cell placement in the organism indexed from 0 to \( n - 1 \). In other words, an organism with \( UOID = 101 \) represents the organism of size \( n = 2 \) where cell \( c_0 \) uses Rule 0 and cell \( c_1 \) uses Rule 1 from Table 1.3. SIZE is stored as an integer value corresponding to the organism size \( n \) and PSC is the number of states explored in the state space sequentially (\( PSC \leq 2^n \)).

In the attractor table, OID is the organism ID and stored in the same way as a UOID. However, in the attractor table each OID is not necessarily unique since each organism may have more than 1 attractor. UAID is a unique attractor ID composed of the binary representation of the attractor state with the smallest canonical value in that attractor cycle. Each UAID is not necessarily unique since the attractor name may be the same between two organisms. However, for a given organism each attractor UAID will be unique. As a result, the combination of OID and UAID may be used to index a row in the attractor table. LENGTH is the length of the attractor or number of attractor states in an attractor cycle. SDC is the number of sequential states that lead to the attractor as a result of repeated successor computation. RDC is the number of random start states that lead to the attractor as a result of repeated successor computation. EBS is the estimated basin of attraction size of the attractor. C is the certainty of the estimated basin of attraction size based upon the total number of states in the state space that have been explored sequentially: \( C = PSC/2^n \). COLORS is a hyphen delimited string of integers where each integer is the periodicity of the correspondingly indexed cell in the organism across the attractor. In other words, if COLORS = 1 − 2 − 1 then the organism of size \( n = 3 \) has cells \( C_0 \) with periodicity 1, \( C_1 \) with periodicity 2, and \( C_2 \) with periodicity 1 as the organism traverses the attractor.

In the attractor data table, OID and UAID are the same as in the attractor table.
SNO is an integer value representing a number of successor states from the attractor state that is the UAID. BM is an integer value representing the index value of a cell of the organism. Using the SNO and BM values for a given organism OID and attractor UAID pair, each possible mutation can be uniquely referenced. Each of these mutations results in a mutation edge leading to a UAID. DSTAID is the UAID reached from the mutation edge resulting from the mutation of cell $C_{BM}$’s state of the attractor state that is a result of computing the successor state SNO times from the current UAID. In other words, of $UAID = DSTAID$ the mutation edge corresponding to the tuple $OID, UAID, SNO, BM$ is robust. Otherwise, if $UAID \neq DSTAID$ then the mutation edge is not robust.

In the organism data table, OID is a unique organism ID just as UOID is in the organism table. Each OID here corresponds to a single row in the organism data table. ROBUSTNESS is the organism robustness $\rho$ computed by querying the attractor data table for OID and returning the average of all the attractors’ robustness. ALPHA is the number of attractors $\alpha$ computed by counting UAID in the attractor table when querying for OID. TRDC is the total number of random start states that have been traced into attractors computed by summing RDC when querying the attractor table for OID. C is the certainty of any estimations made in this table including $\alpha$ and $\rho$ and is computed as $C = \frac{PSC}{2^n}$.

Limitations

Limitations result from the current structure. Specifically, processing or computation that requires reading from the database can not be distributed. In other words, the organism data table can not be distributed to the clients in the cluster for computation. Reads and writes to the singular database instance result in a bottleneck. Bulk compression and insertion helps mitigate the congestion, but ultimately does not resolve the underlying problem.
Table 6.1: Database tables and fields and their types

<table>
<thead>
<tr>
<th>Table Name</th>
<th>(Field, Type)</th>
</tr>
</thead>
<tbody>
<tr>
<td>organism table</td>
<td>(ID, Integer), (UOID, Text),</td>
</tr>
<tr>
<td></td>
<td>(SIZE, Integer), (PSC, Text)</td>
</tr>
<tr>
<td>attractor table</td>
<td>(ID, Integer), (OID, Text),</td>
</tr>
<tr>
<td></td>
<td>(UAID, Text), (LENGTH, Text),</td>
</tr>
<tr>
<td></td>
<td>(SDC, Text), (RDC, Text),</td>
</tr>
<tr>
<td></td>
<td>(EBS, Text), (C, Float),</td>
</tr>
<tr>
<td></td>
<td>(COLORS, Text)</td>
</tr>
<tr>
<td>attractor data table</td>
<td>(ID, Integer), (OID, Text),</td>
</tr>
<tr>
<td></td>
<td>(UAID, Text), (SNO, Text),</td>
</tr>
<tr>
<td></td>
<td>(BM, Integer), (DSTAID, Text)</td>
</tr>
<tr>
<td>organism data table</td>
<td>(ID, Integer), (OID, Text),</td>
</tr>
<tr>
<td></td>
<td>(ROBUSTNESS, Float),</td>
</tr>
<tr>
<td></td>
<td>(ALPHA, Text), (TRDC, Text),</td>
</tr>
<tr>
<td></td>
<td>(C, Float)</td>
</tr>
</tbody>
</table>

6.3 Libraries

The required functions are grouped in header files so that the calls can exist independent of their implementation. Header files are broken down into the following groupings:

- libdb contains functions for basic database functionality including read and write
- libgdb contains functions for simulating organisms and computing dynamical properties
- libsym contains functions for computing color and chromatic symmetry
- libpdb contains functions for printing and outputting data

For each of the header files a corresponding C implementation exists that is compiled into an object file, which can be accessed by C programs. Current implementation makes use of the MySQL database and structure described above.
CHAPTER 6. SOFTWARE

Limitations

The print library only outputs data from the database and this does not include tributaries. I developed an independent program for generating full state space and mutation phase space of a single organism for organism size \( n \leq 20 \). However, this program only outputs to dot files without writing to the database. The limitation of creating images for larger organisms exists due to the number of nodes in the graph at larger organism sizes coupled with the memory limitations of programs like graphviz. Modifications to graphviz were not explored because the images were not accessible visually at that density and increasing image size would not be practical.

6.4 Usage

Usage of the software is broken down into 4 main programs corresponding to each table being written to: generateOrganism, generateAttractors, generateAttractorData, and generateOrganismData.

The program generateOrganism takes as primary input a mode of operation (o, s, sz, h, hr, rbs). Organism mode o requires additional input of the UOID and creates a single organism entry of UOID in the organism table initializing PSC. Sequential mode s requires additional input of the UOID and a number of organisms to generate. This mode sequentially generates the requested number of organisms by incrementing from the given UOID and writing the results to the organism table. Size mode sz requires additional input of organism size and number of organisms to generate. Organisms are generated sequentially from the last UOID of that size and written to the organism table. Homogeneous mode h requires input of a size, and writes the UOID for every possible homogeneous organisms of that size to the organism table. Homogeneous ranged mode hr requires a starting organism size and ending organism size, repeating mode h for each of the sizes in the range specified.
Finally random by size mode \textit{rbs} requires input of organism size, number of organisms to generate, and a seed for the randomization function. This mode randomly generates the requested number \emph{UOID} of the specified size and these are written to the organism table.

The program \textit{generateAttractors} takes as primary input a mode of operation (\textit{sbs}, \textit{cbid}, \textit{pbid}, \textit{rbid}, \textit{rbs}). Sequential by size mode \textit{sbs} requires additional input of organism size. Complete by organism id mode \textit{cbid} requires additional input of \emph{UOID}. Partial by organism id mode \textit{pbid} requires additional input of \emph{UOID}, a number of states to simulate, and an initial state to start incrementing sequential states to simulate from. The \textit{sbs}, \textit{cbid}, and \textit{pbid} modes will update the processed state counter \emph{PSC} in the organism table and each attractor discovered from each initial state will have it’s sequential state counter \emph{SDC} incremented. Random by organism id mode \textit{rbid} requires additional input of \emph{UOID}, a number of random start states to simulate from, and a seed for the randomization function. Random by size mode \textit{rbs} requires additional input of organism size \(n\), number of random start states to simulate from, and a seed for the randomization function. The \textit{rbid} and \textit{rbs} modes will increment the random state counter \emph{RDC} for each attractor discovered from a random initial state. All \textit{generateAttractors} modes generate a new row in the attractor table for each newly discovered attractor reached from an initial start state with \emph{OID}, \emph{UAID}, \emph{LENGTH}, \emph{SDC}, \emph{RDC}, \emph{EBS}, \emph{C}, and \emph{COLORS}. Otherwise if a pre-existing attractor is encountered, then \emph{SDC} is incremented or \emph{RDC} is incremented depending on whether it was reached from a random start state or sequential start state and \emph{EBS} and \emph{C} are updated to reflect the change as well.

The program \textit{generateAttractorData} takes as primary input a mode of operation (\textit{uaid}, \textit{uoid}, \textit{s}). \emph{UAID} mode requires additional input of the \emph{UAID} and \emph{UOID} pair and the length of attractor \emph{UAID}. This mode iterates through the attractor states for the specified number of states in the attractor. For each state identified by a significant node offset and for each bit mutation of that attractor state the program computes and writes the corresponding
destination attractor ID to the attractor data table. *UOID* mode requires additional input of the *UOID*. In this mode, the program queries the attractor table for the *UAID*s and attractor *LENGTH*s associated with that *UOID*. Next, the program repeatedly executes *UAID* mode with the *UOID* and each resulting *UAID* and *LENGTH* from the query. Size mode *s* requires additional input of the organism size. This mode queries the organism table for all *UOID* matching the specified size and then repeatedly executes *UOID* mode with the resulting *UOID*s from the query.

The program **generateOrganismData** takes as primary input a mode of operation (*uaid*, *uoid*, *s*). *UAID* mode requires additional input of the *UAID* and *UOID* pair. The program queries the attractor data table for all rows matching the *UAID* and *UOID* pair computing the total number of attractors, the robustness of the organism, and the total random start states. Next, a query to organism table for the number of sequential states explored matching the *UOID* organism is used to compute the certainty. Finally, these values are written to the organism data table. *UOID* mode requires additional input of the *UOID*. This mode repeats the *UAID* mode for all attractors matching the *UOID* in the attractor table. Size mode *s* requires additional input of an organism size and repeats the *UOID* mode for all *UOID* matching the specified size in the organism table.

Each experiment was written as a script that invoked the calls to the above described programs. For example, the experiment described in Chapter 2 would be executed with the sequence of commands in Table 6.2. These commands can easily be contained or generated and then executed by a script. Alternatively, a simple c program can directly make function calls using the libraries described in Section 6.3.

The source code described above is publicly available through a repository hosted at "https://sourceforge.net/p/rbndb/code/ci/master/tree/DB/". Breaking up the simulation functionality and experimental steps into functions that are accessible through libraries via header files facilitates ease of expanding or modifying the code base for future experiments.
Table 6.2: Program usage for hypothesis 1 experiment

<table>
<thead>
<tr>
<th>To Generate Organisms</th>
<th>To Generate Attractors</th>
</tr>
</thead>
<tbody>
<tr>
<td>generateOrganism hr 2 16</td>
<td>generateAttractors sbs 2</td>
</tr>
<tr>
<td>generateOrganism rbs 2 1000 1</td>
<td>generateAttractors sbs 3</td>
</tr>
<tr>
<td>generateOrganism rbs 3 1000 1</td>
<td>generateAttractors sbs 4</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>generateOrganism rbs 16 1000 1</td>
<td>generateAttractors sbs 16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To Generate Attractor Data</th>
<th>To Generate Organism Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>generateAttractorData s 2</td>
<td>generateOrganismData s 2</td>
</tr>
<tr>
<td>generateAttractorData s 3</td>
<td>generateOrganismData s 3</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>generateAttractorData s 16</td>
<td>generateOrganismData s 16</td>
</tr>
</tbody>
</table>

The repository contains some of the scripts for the experiments, the libraries and header files, the client and server daemons for distributed processing, and other experiments’ programs.
Chapter 7

Future Work

The research and experiments in the preceding chapters relied on software that serves as a basis for continued experimentation. For each of the experiments, I describe future research here. In some cases initial steps and preliminary data has been collected. Data, programs, and scripts developed for these experiments are publicly accessible from the repository: “https://sourceforge.net/p/rbndb/”.

Expanding upon Chapter 2 entails formally defining the shift of the organism property of homogeneity towards increased heterogeneity that occurs as a result of growth or mutation. A formal definition would provide a clear basis for identifying the potential benefits or repercussions of continued cellular differentiation. In chapter 3, I define minimally heterogeneous [7] as the first step of this transition from homogeneity to heterogeneity. The next experiment should explore the relationship of homogeneous organisms to this specific subset of heterogeneous networks by repeating the experiment from Chapter 2 with minimally heterogeneous organisms instead of randomly sampled heterogeneous organisms. The experiment should be further expanded to explore how a progressive shift towards increased organism heterogeneity impacts the dynamical properties of adaptivity and robustness. Additionally, future work connecting the results from Chapter 2 and Chapter 3 should formalize
the relationship of robustness to adaptivity.

Expanding upon Chapter 3 entails formally proving that the number of attractors increase exponentially during growth in minimally heterogeneous XOR organisms. This proof combined with proofs 3.6.4 and 3.6.4 confirming the collapse of adaptivity in homogeneous XOR organisms at organism sizes that are powers of 2 would formally prove Hypothesis 2 for homogeneous XOR organisms.

Expanding on Chapter 4 entails expanding the experiment testing rotational chromatic symmetry to explore bilateral symmetry resulting from mutation. This experiment should simulate all possible mutations. This next experiment should also explore a greater range of organism sizes and organism types besides the homogeneous XOR organism.

Expanding on Chapter 5 entails identifying how well the proposed modified capture re-capture estimation technique performs in relation to existing estimation techniques including the Lincoln-Petersen method. The experiment should perform estimations on simulated organisms with known population results to measure accuracy of the estimations. Another experiment should identify the ideal sample size relative to state space or population size and look for other techniques to improve accuracy or minimize the required sample size. Future work should also involve an experiment measuring the accuracy of basin size estimation.

Future work for Chapter 6 should explore other database structures that can alleviate the bottleneck of centralized storage. This structural change should include software changes improving indexing and enabling distributed processing in a dynamically changing distributed environment instead of just within a specified cluster environment.

7.1 Growth

The focus of Chapter 4 is growth rates that facilitate the persistence of the dynamical property of rotational chromatic symmetry. Growth rate is just one network property that
can be explored in relation to dynamical properties. Rotational symmetry is just one form of symmetry that can be explored in relationship to network properties. Future work should continue to explore relationships between network properties and dynamical properties.

Here, I briefly explore another form of chromatic symmetry in the synthetic organisms that is commonly observed in biological organisms: bilateral symmetry. Because the organisms are simulations and neighboring cells are the only form of input, the organisms are a closed system. In the following experiment, I simulate outside information to the organisms using cellular mutation of the rule being applied at a particular cell and explore the resulting dynamics [10].

### 7.1.1 Bilateral Symmetry

Chapter 4 focuses on growth rates that preserve rotational symmetry in homogeneous XOR organisms. However, rotational symmetry is one form of symmetry and many biological organisms, like the starfish, exhibit bilateral symmetry. Further the growth considered hereto in this Chapter only varies by rate and not by cell type. In this section I begin to examine morphogenesis and bilateral symmetry arising from cellular differentiation, however more work is still required.

Considering how organisms form bilateral symmetry in nature, I draw inspiration for the following experiment from “Steps to an Ecology of Mind” where Bateson writes: “An unfertilized frog’s egg is radially symmetrical, with animal and vegetal poles but no differentiation of its equatorial radii. Such an egg develops into a bilaterally symmetrical embryo, but how does it select one meridian to be the plane of bilateral symmetry of that embryo? The answer is known—that, in fact, the frog’s egg receives information from the outside.” [5]

In other words the development of bilateral symmetry from an organism that has rotational symmetry requires external information or stimulus.

External information or stimulus can be simulated with a mutation or cellular differenti-
ation at a single point in the organism. This mutation serves as a change in the information that internally simulates the effect of external forces on information communicated between cells.

Consequently, to test the assertion about biological organisms, here I rephrase the assertion for synthetic organisms: “Cellular differentiation leads to bilateral symmetry: If we start in an attractor with nontrivial rotational symmetry segment size in the dynamics of a homogeneous function 6 (XOR) organism and one of its’ cells is mutated to function 9 (NXOR), then the attractor state will lead to an attractor that is bilaterally symmetric in the dynamics of the mutated organism.” [10]

Continuing the experimental methodology, this experiment re-uses the existing program to simulate and collect data. The experiment required extending the program that computes rotational symmetry to also test for bilateral symmetry.

The experiment simulates homogeneous XOR organisms and minimally heterogeneous organisms derived from a single cell mutation of the homogeneous XOR organism. In this experiment, the single cell mutation is from XOR to NXOR but there are 15 other possible mutations that could be simulated. Future work would continue the experiment by simulating all possible mutations, which is computationally intensive.

### 7.1.2 Preliminary Results

“In Table 7.1, we compare the percentage of single-cell mutations in each attractor which lead to a bilaterally symmetric attractor in settings where nontrivial rotational symmetry is both present and absent. Column 1 is organism size where attractors with nontrivial rotational symmetry segment sizes have been discovered by sampling; column 2 is the number of attractors with nontrivial chromatic rotational symmetry; column 3 is the percent of bilaterally symmetric attractors reached from attractor states in column 3 by mutating a single cell from function 6 (XOR) to function 9 (NXOR) in the original organism; column 4
Table 7.1: Results of testing bilateral symmetry resulting from single cell mutation. Source: [10]

<table>
<thead>
<tr>
<th>org size</th>
<th>#</th>
<th>% become bilateral</th>
<th>#</th>
<th>% become bilateral</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>6</td>
<td>100</td>
<td>4</td>
<td>100</td>
</tr>
<tr>
<td>10</td>
<td>45</td>
<td>90</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>12</td>
<td>42</td>
<td>100</td>
<td>28</td>
<td>100</td>
</tr>
<tr>
<td>14</td>
<td>231</td>
<td>100</td>
<td>67</td>
<td>100</td>
</tr>
<tr>
<td>15</td>
<td>20</td>
<td>47</td>
<td>20</td>
<td>33</td>
</tr>
<tr>
<td>18</td>
<td>1000</td>
<td>94</td>
<td>1000</td>
<td>89</td>
</tr>
<tr>
<td>20</td>
<td>606</td>
<td>100</td>
<td>606</td>
<td>95</td>
</tr>
<tr>
<td>22</td>
<td>893</td>
<td>100</td>
<td>893</td>
<td>100</td>
</tr>
<tr>
<td>24</td>
<td>100</td>
<td>100</td>
<td>1000</td>
<td>100</td>
</tr>
<tr>
<td>26</td>
<td>654</td>
<td>100</td>
<td>654</td>
<td>100</td>
</tr>
<tr>
<td>28</td>
<td>270</td>
<td>100</td>
<td>270</td>
<td>100</td>
</tr>
<tr>
<td>30</td>
<td>998</td>
<td>100</td>
<td>446</td>
<td>90</td>
</tr>
</tbody>
</table>

is the number of attractors without nontrivial chromatic rotational symmetry; column 5 is the percent of bilaterally symmetric attractors reached from the attractor states in column 4 by mutating a single cell from function 6 (XOR) to function 9 (NXOR) in the original organism.” [10]

“Table 7.1 shows that there is a high probability after cellular mutation an attractor state will lead to a bilaterally symmetric attractor in the dynamics of the mutated organism. However, the probability of the attractor state leading to a bilaterally symmetric attractor is higher if the initial attractor state has a nontrivial chromatic rotational symmetry.” [10]

Future work would expand upon the mutations, organism sizes, and attractor states tested.
Chapter 8

Conclusion

This thesis studies the impact of natural selection for dynamical properties on cellular differentiation and organism growth. More specifically, I take an experimental approach to explore the relationship between the network properties of homogeneity, heterogeneity, and growth rate and the dynamical properties of robustness, adaptivity, and chromatic symmetry. I propose and test the following three hypotheses: 

Hypothesis 1: Cellular differentiation increases the expected robustness in an organism’s dynamics. 
Hypothesis 2: Cellular differentiation leads to more uniform adaptivity as the organism grows. 
Hypothesis 3: Growth by segment elongation is more likely to preserve symmetry in dynamical properties than growth by segment reduplication.

In the experimental approach first I develop software described in Chapter 6 to simulate networks for each experiment. Next I execute the program and collect the data from the simulations. Finally I analyze the results.
CHAPTER 8. CONCLUSION

8.1 Results

8.1.1 Hypothesis 1

The experiment for Hypothesis 1 simulates homogeneous organisms of sizes 2, \ldots, 16 and compares their dynamical properties to the dynamical properties averaged across simulation of 1000 random heterogeneous organisms of the sizes 2, \ldots, 16. The comparison focuses on homogeneous organisms that have a nontrivial number of attractors (e.g. $\alpha \geq 1$) and tests whether this subset of homogeneous organisms or randomly sampled heterogeneous organisms have greater robustness on average.

If $X$ is an organism, then $|X|$ denotes the number of cells in $X$. The organism of size $n$ where all cells operate according to function $i$ is denoted $X^n_i$. The set of all homogeneous organisms of size $n$ is denoted

$$Hom(n) = \{X^n_i \mid i = 0, \ldots, 15\}.$$  

The set of all heterogeneous organism of size $n$ is denoted

$$Het(n) = \{X \mid X \text{ is a heterogeneous organism of size } n\}.$$  

*Formal Hypothesis 1 is as follows*

$$\text{mean}\{\rho(X) \mid X \in Het(n)\} > \text{mean}\{\rho(X) \mid X \in Hom(n)\}.$$  

The results from Chapter 2 show two possibilities for homogeneous organisms that grow in size while remaining homogeneous: either an organism will have high robustness and a bounded number of attractors or it will have low robustness but an unbounded number
of attractors. In contrast, heterogeneous organisms have both high expected robustness and an unbounded expected number of attractors. Further, in comparison to the Class 2 homogeneous organisms that exhibit unbounded number of attractors during growth, a randomly selected heterogeneous organism of the same size will, with probability tending towards 1, have greater robustness and typically have increasing numbers of attractors as the heterogeneous organism grows in size. In other words, the results support Hypothesis 1 when comparing Class 2 homogeneous organisms to randomly sampled heterogeneous organisms [9].

8.1.2 Hypothesis 2

The experiment for Hypothesis 2 simulates homogeneous and minimally heterogeneous organisms for sizes 2, . . . , 256 at sizes which are proper powers of 2. The resulting adaptivity of the homogeneous organisms is compared to the resulting adaptivity of the minimally heterogeneous organisms. The comparison tests whether organisms with a growth rate of 1 will experience an increase in adaptivity if during the growth they undergo cellular differentiation.

Let $X \in \text{Hom}(n)$. Let $X[k]$ be the organism obtained by mutating $X$ so that one cell now uses function $k$. The minimally heterogeneous organism of size $n$ derived from a homogeneous organism operating function $i$ at every cell is

$$
\text{MinHet}(n, i) = \{X^n_i[k] \mid k = 0, \ldots, 15. k \neq i\}.
$$

Applying the definitions, formal Hypothesis 2 is as follows: If $k$ is a class 2 homogeneous organism then $\forall n \geq 2, \exists i \geq 0$ such that:

$$
\text{mean}\{\alpha(X) \mid X \in \text{MinHet}(n + i, k)\} > \text{mean}\{\alpha(X) \mid X \in \text{Hom}(n + i)\}.
$$
The experimental results from Chapter 3 show that the dynamics of homogeneous XOR organisms exhibit a collapse to a single attractor at organism sizes which are of the form $n = 2^i$. At organism sizes of the form $n = 2^i$ where $i > 2$ minimally heterogeneous XOR organisms where the differentiated cell applies one of the following rules $\{1, 3, 5, 7, 8, 10, 12, 14\}$ have dynamics which do not collapse to a single attractor. These empirical results support Hypothesis 2 for the limited subset of homogeneous XOR organisms. In other words, if homogeneous XOR organisms were to undergo cellular differentiation during growth they could maintain increasing adaptivity instead of a collapse in adaptivity.

The results from Chapter 3 include formal proofs for Theorem 1 (stated originally on p. 46) and Theorem 2 (stated originally on p. 46): 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. If regardless of initial state $X$ the homogeneous XOR organism always ends up in the same attractor, then the number of cells in the organism is a power of 2. The two proofs show that the organism will have only 1 attractor, which is length one and consists of the state where all cells have a value of 0, if and only if the number of cells in the organism is a proper power of 2 [7].

8.1.3 Hypothesis 3

The experiment for Hypothesis 3 simulates the dynamics of homogeneous XOR organisms at sizes $n = 2, \ldots, 50$ from 100,000 random start states. The program computes: attractor length, periodicity of each cell across the attractor, chromatic symmetry across the attractor, and the segment length of the rotational symmetry for each attractor discovered from a random start state. Next, the program compares each resulting nontrivial chromatic symmetry of an organism of size $n$ across an attractor with all possible chromatic symmetries of the organism at size $n+l$ (growth by segment length or reduplication) and at size $n+n/l$ (growth
by segment count or elongation). In other words, the experiment compares impact of growth rate of an organism on chromatic symmetry by modifying the growth rate to a growth rate that could maintain the chromatic symmetry of the organism across its attractors.

Given organism $X$, $|X| = n$. Let $\text{Att}(x) := \{A \mid A \text{ is an attractor of } X\}$. Given $l/n$, the segment-$l$ organisms are defined as

$$\text{Sym}(X, l) = \{A \in \text{Att}(X) \mid X \text{ has } \frac{|X|}{l}-\text{fold symmetry in an attractor}\}.$$ 

Applying the definitions, **formal Hypothesis 3 is as follows**: Given $n \geq 2$, $l|n$, then:

$$\frac{|\text{Sym}(X_0^{n+\frac{r}{i}}, l + 1)|}{|\text{Att}(X_0^{n+\frac{r}{i}})|} \geq \frac{|\text{Sym}(X_0^{n+i}, l)|}{|\text{Att}(X_0^{n+i})|}. $$

The experimental results from Chapter 4 show that the percent of attractors discovered with rotational symmetry with corresponding segment length that are non-zero occurs more for the section Elongation than for the section Reduplication for the sampled subset of homogeneous XOR organisms. These empirical results support Hypothesis 3 that growth by elongation is more likely to preserve rotational chromatic symmetry [10].

### 8.1.4 Analysis

The experimental results shown in Table 2.11 support Hypothesis 1. This suggests that selection for the dynamical property of robustness during growth will drive the cellular differentiation and morphogenesis. The experimental results shown in Table 3.2 coupled with the proof that adaptivity of homogeneous XOR organisms collapse at size $n = 2^i$ support Hypothesis 2. This suggests that selection for the dynamical property of adaptivity will drive growth and morphogenesis by cellular differentiation. The experimental results
as shown in Figure 4.9 support Hypothesis 3 and suggest that selection for the dynamical property of rotational chromatic symmetry could drive the selection of growth by elongation. Consequently, these combined experimental results from Chapters 2, 3, and 4 demonstrate that morphogenesis and growth can be driven by selection of dynamical properties!

8.2 Software

The experimental approach faces obstacles related to big data. The main obstacles faced from exponentially growing state space and data from simulating organisms can be categorized as: number of computations, memory, and search (reading and writing). In Chapter 5 I describe the following three approaches to addressing the obstacles of memory and computational limits: algorithms, sampling, and distributed processing. Search (reading and writing) is left for future work and possibly an infrastructure modification. The three candidate approaches to mitigate memory intensive and computationally intensive tasks are implemented in the software described in Chapter 6. The modular approach using libraries and scripts facilitates ease of switching between approaches, creating additional approaches and functionality, or changing the underlying infrastructure.

Customized algorithms are used to minimize memory usage and the number of computations. Sampling is used to minimize the data required and in turn the memory used and the number of computations. Distributed processing is used to break up the computations into parts and distribute them to many processors. Each processor then only computes its subtask while an aggregator collates the results.

8.2.1 Trading time for space

In Chapter 5 I describe two algorithms developed to decrease both memory and computational requirements. The first algorithm improves upon the trivial method of storing and
CHAPTER 8. CONCLUSION

comparing every state by instead storing every 2\textsuperscript{i}th state. This decreases the memory re-
quired in the worst case for organisms of size \( n \) from the exponential 2\(^n\) to the linear \( \log_2 2^n \)
or \( n \). The second algorithm decreases computational steps for homogeneous XOR organisms
by simulating more than one successor step at a time computing forward in powers of 2 suc-
cessor steps at once. The algorithm applies the proof in Chapter 3 to decompose a starting
organism state into \( \bar{0}1 \) states, for each state compute forward a power of 2 successor steps
applying the general case of Lemma 15, and finally recomposing the organism’s successor
state from the resulting computed states.

Note in the case of the second algorithm, while it enables computing forward by a power
of 2 successor steps, the algorithm is limited to homogeneous XOR organisms and does not
compute intermediary states. The lack of intermediary states makes identifying an attractor
more challenging unless the attractor length is already known. Though, if the attractor data
is already known then there is no need to compute or simulate it.

8.2.2 Trading accuracy for time

Estimation techniques described in Chapter 5 offer a trade off of accuracy for computational
time. Since fully exploring the state space of larger organisms is computationally intractable,
this trade of accuracy for time enables the experimental approach to still provide data for
larger organisms. Random sampling of the state space is generally for the estimations of
the organism and attractor properties. However, estimations can also be conducted using
data from incomplete sequential exploration data. Estimating basin size \( EBS \) as a result of
partial sequential exploration is given by \( EBS = (SDC/PSC)(2^n) \), and estimation of basin
size as a result of random sampling is given by \( EBS = (RDC/TRDC)(2^n) \). Certainty \( C \)
for sequential explored state space is computed as \( C = (PSC/2^n) \) and for random sampling
\( C = (TRDC/2^n) \). Estimating attractor count for randomly sampled organisms can be done
using either of two proposed techniques. The first technique uses a single interval of random
CHAPTER 8. CONCLUSION

sampling. Let $A_i$ be the basin of attraction for attractor $i$. Given the total number of states in the state space is $2^n$ for organisms of size $n$ and the assumption that the space sampled is representative of the entire state space, $\alpha_e(\sum_{i=1}^{k} A_i) \approx 2^n$. Solving for $\alpha_e$,

$$\alpha_e \approx 2^n / \sum_{i=1}^{k} A_i.$$

The second technique uses two intervals of random sampling in the same fashion as the Lincoln-Petersen method. Let $\sum_{i=1}^{k} A_1^i$ be the total number of interval 1, $\sum_{i=1}^{k} A_2^i$ be the total number of interval 2, and $\sum_{i=1}^{k} R_i$ be the number recaptured. Then

$$\frac{\sum_{i=1}^{k} A_1^i}{\alpha[\sum_{i=1}^{k} A_1^i + \sum_{i=1}^{k} A_2^i]} = \frac{\sum_{i=1}^{k} R_i}{\sum_{i=1}^{k} A_2^i}$$

and

$$\alpha[\sum_{i=1}^{k} A_1^i + \sum_{i=1}^{k} A_2^i] < 2^n.$$

Estimation using the Lincoln-Petersen method involves two intervals of random sampling where the first interval results in a capture 1 $C_1$ and the second interval results in a capture 2 $C_2$. These estimation of the number of attractors in the state space is computed as $\alpha_e = (C_1 \times C_2)/R$. Comparing the accuracy of each of the described estimation techniques is a future experiment.

8.2.3 Parallel distributed processing

Computational limits can be addressed in part using distributed processing described in Chapter 6. In the software I develop to simulate organisms, distributed processing is accomplished through a message passing interface that breaks simulation tasks up and assigns the subtasks to clients as described in Figure 5.1. The results from each of the subtasks executed by the clients are then collated by the centralized server, compressed, and bulk inserted into
the MySQL database. The distributed approach addresses some of the associated computational and memory limits related to simulation; however, as organism sizes increase the state space grows exponentially and attractor discovery can be computationally intractable even for a single attractor. In the case where the subtask of identifying an attractor from a single starting state is computationally intractable, the distributed processing solution becomes moot. Further, using a centralized database results in a bottleneck with the database read/write operations being the limiter.

8.3 Applications

The combination of Hypotheses 1 and 2 suggest that growth for homogeneous organisms will reach a threshold beyond which cellular differentiation and heterogeneity offer more desirable dynamical properties. Further, it is apparent both in nature and the simulations that developing complex specialized structures through differentiated cells and cell networks provides increased robustness and adaptivity.

The application of the combined results offers insight into building networks with desired dynamical properties. This extends to any network whether physical, ecological, social, or computer. In computer networks, architecting a network that is adaptive to a variety of tasks and environments could be better accomplished with a heterogeneous set of nodes, services, or protocols. Similarly these results can be applied to neural networks and the need for greater number of classifications.

In computer networks, the decision of how to architect a network that is robust to intrusion or failure could be better accomplished with a heterogeneous set of nodes, services, or protocols. With farms genetic diversity is known to be more resistant to disaster. Homogeneous networks composed of a single genetic strain, a single protocol, or a single operating system are vulnerable to a single infection spreading easily to every node or cell in the net-
work causing it to collapse entirely. Heterogeneous networks with diverse genetic strains, multiple protocols, or various operating systems may have a portion of the nodes infected; however, the network as a whole will be more likely to persist.
Glossary

**attractor** the set of all states that are in the same cycle in the dynamics space of an organism. 10

**attractor length** the number of states in an attractor. 10

**attractor robustness** the number of robust mutation edges of the attractor divided by the total number of mutation edges of the attractor states. 17

**attractor state** an organism state that occurs in a cycle in the dynamics space of an organism. 10

**average attractor count** the average number of attractors for a set of organisms. 22

**average robustness** the average organism robustness for a set of organisms. 22

**basin of attraction** the set of tributaries leading to all attractor states of a single attractor. 10

**black box** a system observed in terms of inputs and outputs where the internal workings are not accessible nor known. 4

**Boolean** either 0 or 1. 6

**cell** a node in a network of cellular automata. 6

**cell state** the state of a cell during a discrete time interval. 6

**chromatic symmetry** non-trivial rotational symmetry in the coloring of the organism cells as it traverses an attractor. 73

**Class 1** the set of homogeneous organisms where $\alpha$ remains bounded by some constant $b$ as the organism grows. 23

**Class 2** the set of homogeneous organisms where for all constants $b$, there exists a size $n_b$ at which $\alpha$ exceeds $b$. 23

**color** the periodicity of a cell’s state as the organism traverses an attractor; color is assigned to an individual cell. 71
connected component an attractor and its basin of attraction are a connected component in the dynamics of an organism. 10

deterministic resulting in the same state given the same initial state. 6

dynamics a directed graph $S = (2^V, D)$ whose vertex set consists of all possible states of the organism (i.e. the power set of $V$), and whose edge set $D$ includes every ordered pair $(X, Y)$ for which $s^+(t) = X \implies s^+(t + 1) = Y$. 9

elongation organism growth by incrementally increasing the segment length from $l$ to $l + 1$. 73

foldedness $k$, the number of times segment size $l$ divides into organism size $n$ $k = \frac{n}{l}$. 73

Garden of Eden state a state in the dynamics of the organism that cannot be reached from any other state by a directed edge. 9

heterogeneous not every cell in an organism has the same function assigned to it. 8

homogeneous every cell in an organism has the same function assigned to it. 8

linear cyclic an organism is linear cyclic when each of its cells is directly connected only to their two immediately neighboring cells such that the cells form a ring where an initial cell is connected to the next cell and the last cell. 6

mutation the perturbation of a random cell state in the attractor state; in other words, a bit flip of a single bit in the organism state. 15

mutation edge an edge connecting an originating attractor state to one of its resulting mutation states. 15

mutation state the organism state that results from a mutation of a single random cell state in an attractor state. 15

number of attractors the total number of attractors in the dynamics space of an organism, also referred to as $\alpha$. 18

organism robustness the average attractor robustness of all the attractors in an organism's dynamics space; also referred to as $\rho$. 18

organism state the ordered set of its cell states during a discrete time interval. 9

organisms a network of cellular automata where each node of the network will be a cell of the organism. 6
reduplication organism growth by incrementally increasing the number of segments from \( k \) to \( k + 1 \). 73

robust a mutation edge that leads back to the originating attractor by a path of successor states. 16

rotational symmetry rotational symmetry of an organism’s coloring; in other words, after a certain amount or segment of rotation of the organism cell coloring, the coloring is the same. 73

segment size the number of cells which when rotated by result in the same coloring of an organism. 73

successor \( Y \) is a successor of \( X \) if an organism state \( X \) is connected to organism state \( Y \) by a directed edge from \( X \) to \( Y \) in the dynamics of the organism. 9

synchronous every cellular state is instantaneously determined at each time interval. 7

tributary the set of states composing a path of tributary states leading up to but not including the first encountered attractor state when computing successor states starting from a Garden of Eden state. 10

tributary state a state that is not in an attractor. 10

trivial rotational symmetry a segment size \( l = 1 \) and foldedness \( k = n \); in other words, the case where all cells have the same periodicity resulting in an organism where all cells have the same color. 73
Acronyms

ALPHA $\alpha$, also number of attractors. 105, 106

BM bit mutated. 105, 106

C certainty. 104–106

COLORS cell colors of an organism across an attractor. 104, 106

DSTAID destination attractor ID. 105, 106

EBS estimated basin size. 104, 106

LENGTH attractor length. 104, 106

OID organism ID. 104–106

PSC processed state count, also total sequential state count. 104, 106

RDC random dart count, also random state count. 104, 106

ROBUSTNESS $\rho$, also organism robustness. 105, 106

SDC sequential dart count, also sequential state count. 104, 106

SIZE $n$, also organism size. 104, 106

SNO significant node offset. 105, 106

TRDC total random dart count, also total random state count. 105, 106

UAID unique attractor ID. 104, 106

UOID unique organism ID. 104, 106
Bibliography


