SYMMETRY AND UNIVERSAL CRITICAL PROPERTIES OF A CLASS OF DISCRETE STATE NETWORKS

A Dissertation

Presented to

the Faculty of the Department of Physics

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

By

Shabnam Hossein

May 2015

SYMMETRY AND UNIVERSAL CRITICAL PROPERTIES OF A CLASS OF DISCRETE STATE NETWORKS

Shabnam Hossein

APPROVED:

Dr. Kevin E. Bassler, Chairman
Dept. of Physics

Dr. Rene Bellwied
Dept. of Physics

Dr. George Reiter
Dept. of Physics

Dr. Ricardo B. R. Azevedo
Dept. of Biology and Biochemistry

Dean, College of Natural Sciences and Mathematics

Acknowledgements

I express my deepest gratitude to my advisor, Dr. Kevin Bassler. This dissertation would not have been possible without his guidance and support. He is not only a great teacher and mentor, but also an extraordinary human being. His perseverance, passion and hard working attitude in both research and life have always been a role model for me.

I want to thank my committee members, Dr. Ricardo Azevedo, Dr. George Reiter, and Dr. Rene Bellwied for their patience and guidance. I should also thank my former committee member, Dr. Bambi Hu for his help and encouragement.

I would also like to thank all group members especially Dr. Florian Greil, Dr. Charo del Genio, Matthew Reichl, Bo li, Amy Nyberg, and Dr. Jimmy Trevino for their friendly supports and great discussions.

This dissertation is dedicated to my parents whose sacrifices and patience are beyond my comprehension.

SYMMETRY AND UNIVERSAL CRITICAL PROPERTIES OF A CLASS OF DISCRETE STATE NETWORKS

An Abstract of a Dissertation
Presented to
the Faculty of the Department of Physics
University of Houston

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

By
Shabnam Hossein
May 2015

Abstract

In this dissertation, we study the statistical mechanics of Boolean networks as a simple model in class of heterogeneous complex systems. Boolean networks are used as generic models of complex systems of many interacting units, such as gene and protein interaction systems, neural networks and economical systems. They are particularly good examples of complex systems to study because they are relatively simple, yet have a nontrivial dynamical phase transition. We investigate the statistical mechanics of how this model behaves and dynamical properties of random Boolean networks at criticality.

First we study the dynamics of critical random Boolean networks and find what the symmetry of the dynamics is. We propose a symmetry group, the *canalization preserving group*, that describes the symmetry observed. The orbits of this symmetry group consist of Boolean functions that have the same canalization values. Canalization is a form of robustness in which a subset of the input values control the behavior of a node regardless of the remaining inputs. We show that the same symmetry governs critical random discrete multi-state networks dynamics with higher number of output values for each node. We investigate the criticality of random multi-state networks, and show that the same canalization preserving symmetry governs the critical multi-state networks dynamics as well.

We also study a particular dynamical property of critical random Boolean networks: their attractor length distribution. Using a known result that nodes relevant to the dynamics on attractors at criticality can be divided into separate components, we analyze the structure of these relevant components and how their dynamics combine to find the distribution of attractor lengths. This is accomplished by mapping the problem to the enumeration of binary Lyndon words. Using analytical arguments we show that the attractor length distribution becomes scale-free in the large network limit with a decay described by a critical exponent of 1. The universal nature of this behavior is demonstrated by comparison to that of the evolved critical state achieved through the playing of an adaptive game that selects for diversity of node's behavior, and that of the attractor length distribution of critical multi-state networks.

Contents

List of Figures

Chapter 1

Introduction

Complex systems theory is an interdisciplinary field that deals with systems consisting of many interacting parts. It is agreed among researchers that complex systems share some distinct features: many (often) heterogeneous parts, non-linear interactions, and organization and collective behavior without an external force [69, 70, 36, 23]. These systems are common in different fields and everyday life. Physical systems such as crystals, magnets, superconductors, non-linear fluids, granular fluids are not only classical condensed matter systems but also examples of complex systems [5]. Ecosystems and biosphere [91, 55], urban societies [15, 18], economics and markets [66, 67] and flocking or schooling behavior in birds [100, 31] are just some other examples of the systems that show complexity in their behavior.

A common approach in modeling complex systems that has gained interest among the physics community is to envision them as networks. Some of the complex systems can intuitively be considered as networks, and some can be approximated as networks

[73, 30, 95, 4, 1]. A network consists of *nodes* or *vertices* that are the units of the system and the interactions between them is modeled by the *links* or *edges* of the network. Since nodes can represent many different type of entities, the abstraction of complex networks allow one to compare the fundamental and statistical properties of many originally diverse systems without the detailed knowledge of the parts of the systems; a standard method in statistical physics. Moreover, there are many useful tools in a physicist's toolbox that can be used to quantitatively describe the behavior of complex networks. These include concepts and methods from non-linear dynamics and statistical physics. Non-linear dynamics gives a framework that can be used to study the non-linear interactions of the system [96]. The physical concepts of phase transition, critical phenomena and scaling that have been borrowed from statistical physics are among the useful tools for describing and analyzing complex systems theories [94, 42, 93, 1].

A particularly important tool that can be used is symmetry. The concept of symmetry has played a pivotal role in the advancement of twentieth-century physics. Accounting for symmetry can greatly simplify analysis and provide powerful insights into the nature and behavior of physical systems. Group theory is the mathematical framework that translates symmetry into suitable mathematical relations. Symmetry and the theory of groups have proven useful in many areas of physics. In particle physics, symmetries have become an indispensable tool of theory formation. It was the classification of hadrons through representations of symmetry group $SU(3)$ that suggested common constituents for these particles and led to the Standard Model [45].

Likewise, in traditional condensed matter physics, symmetry is important for determining the behavior and characteristics of systems that range from the structure of lattices to universal aspects of critical phenomena [57]. Complex systems often differ from more traditional condensed matter systems because they consist of heterogeneous components [22], which makes their analysis and achieving any sort of general understanding of their behavior difficult. In the present work, one of the main questions that we try to answer is: Can symmetry in heterogeneous complex systems be exploited to simplify their analysis and obtain fundamental insights into their dynamics?

In mathematical terms a network is represented by a graph. A graph is an ordered pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that consists of a set $\mathcal{V}$ of vertices and a set $\mathcal{E}$ of edges, where $\mathcal{E}$ can be a set of ordered or unordered pairs of vertices [98]. Ordered pairs of vertices correspond to *directed graphs*, and unordered pairs of vertices correspond to *undirected graphs*.

When modeling a system with a network, in addition to the structure, or topology, of the network, which is the way the nodes are arranged and interact with each other, e.g. direction of the links, link weights, ..., one may define network dynamics [74]. The notion of network dynamics can refer to *dynamics on networks* or *dynamics of networks*. In the first case nodes are individual dynamical systems that are connected through static links. In other words, different dynamical processes take place on a network with a fixed topology [8]. Dynamics of the networks refer to the case where topology of the network evolves with time. Networks are *adaptive*, or *co-evolutionary*, only if there topology and states evolve in an interdependent way

[49, 46, 48, 82]. In this work we focus on the dynamics on the networks.

Dynamical processes on networks can be divided to *continuous* and *discrete* dynamics. Reaction-diffusion models are examples of systems that have continuous dynamics. Continuous dynamical systems are typically modeled using differential equations. Cellular automata and Boolean networks are examples of systems that have discrete dynamics. Difference equations can be employed to model these systems with discrete dynamics [75].

Boolean dynamics is arguably the simplest nontrivial discrete dynamics on a network. Despite their simplicity, such networks, Boolean networks, show nontrivial dynamical behavior. Hence, making Boolean networks ideal paradigmatic examples of complex systems. They can be used to explore the fundamental dynamical properties of complex systems and to develop the tools needed to study these systems.

Boolean networks allow only for two possible states for each node of the network. The reason that they are called Boolean is after George Boole who worked on an algebra in which the values of the variables are the truth values true and false, usually denoted by 1 and 0 respectively [24]. Boolean networks were first introduced by Stuart Kauffman in 1969 [58] as a simple model for genetic regulatory systems of biological cellular organisms. Each gene was assumed to be represented by a Boolean node. An expressed gene that is transcribed is given a value 1, and the gene that is not transcribed is given a value of 0. The links of the networks represented regulatory mechanisms. Since then Boolean networks have been used to successfully model essential features of many other systems in diverse fields including the wild-type gene expression patterns of the *Drosophila melanogaster* segment polarity genes [2], the regulatory sequences of models of *Arabidopsis thaliana* flowers [81], and the cell-cycle network of yeast (*S. cerevisiae*) [62]. They have also been used extensively to model other biological systems, such as neural networks [26, 104], as well as a wide range of physical and social systems [3, 105, 25], including switching circuits [53] and social groups within which consensus emerges through peer interactions [43].

In their original variant, also known as the N-K model, Boolean networks consist of N nodes with binary output states, 0 or 1, that are connected by a directed graph of uniform in-degree K describing their interactions. A network state is a N-tuple of the network's nodes output states. A particular directed graph $\mathcal{G}$ together with a set of update functions $\mathcal{F} = \{f_1, \ldots, f_N\}$ defines a *realization of a directed network*, $(\mathcal{G}, \mathcal{F})$, where each of the N nodes receives K in-links. Thus, every node has exactly K inward pointing links, or in-links. The links can be assigned to nodes differently, but since each link is an in-link on one side and an out-link on the other side, the total number of in-links and out-links is the same. Therefore, the distribution of the in-links and out-links can be different, but their mean value must be the same. Also, the number of in-links of each nodes is fixed to the same value, K. This results in a binomial distribution for the out-links with the mean K which becomes equivalent to a Poisson distribution in the large network limit. Hence, the topology of the N-K model is defined by its number of nodes, N, and its number of in-links, K. If the topology of the network does not change during the dynamical evolution of the network, the model is called *quenched*, whereas in *annealed* model the inputs of the nodes are reset at every time step [32].

The Boolean network dynamics describes how the output states of the nodes change with time. In the classical model of Boolean networks, the output state of the nodes are updated synchronously. However, there are other possible updating schemes such as *deterministic asynchronous* update where the updating order of the nodes is defined once per realization and then stays fixed, and *asynchronous stochastic* update where at each time step the node to be updated is picked randomly [41, 44]. In this work, we consider only the synchronous update scheme.

The output state of the nodes are updated synchronously and periodically, according to the input each node receives from the nodes it is connected to by its in-links, as follows. The state of node i at time t, $s_i(t)$, is the output state of a fixed Boolean function, f_i, assigned to node i,

$$s_i(t) \quad = \quad f_i(s_{i_1}(t-1), s_{i_2}(t-1), \dots , s_{i_K}(t-1)), \tag{1.1}$$

where $s_{i_1}(t-1), s_{i_2}(t-1), \dots , s_{i_K}(t-1)$ are the states of the K nodes connected to node i by its in-links, at time $t-1$. Note that the Boolean function f_i, in general, varies for different nodes, which, along with the differences in their respective connections in the directed graph, makes Boolean networks examples of heterogeneous complex systems.

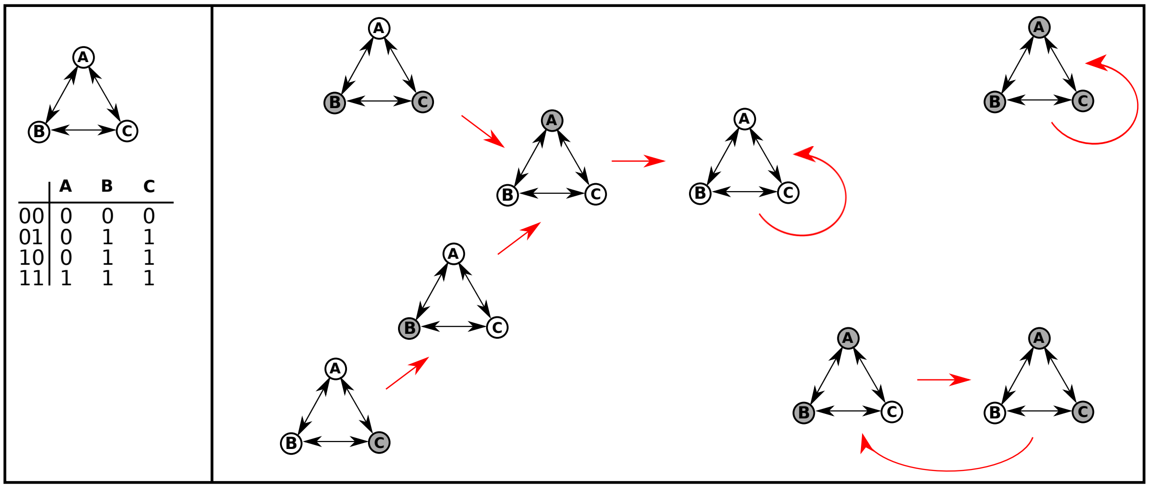
In *random* Boolean networks (RBN), which is the variant of the model mostly considered in this dissertation, both the Boolean functions and the in-links are chosen randomly. Each output function, f_i, is assigned randomly from the set of all possible Boolean functions with K inputs. Since there are 2^K possible input sets for a Boolean function with K inputs, and the output value of the function for each of those input sets can have two values, the total number of possible Boolean functions with K

input 1	input 2	output
0	0	1
0	1	1
1	0	0
1	1	0

Figure 1.1: Example of a Boolean function with $K = 2$ inputs. A Boolean function specifies the output value for each possible set of input values. For $K = 2$ there are $2^2 = 4$ possible sets of input values. A Boolean function is fully defined, if output values are fixed for all the possible input sets.

inputs is 2^{2^K}. Figure 1.1 shows example of a Boolean function with two inputs.

The dynamics of the network can be tuned by varying the choice of the sets of Boolean functions $\mathcal{F}\{f_1, f_2, \cdots, f_N\}$ for each fixed K. Depending on the distribution of their Boolean functions, ensemble-averaged random Boolean networks have two distinct phases of dynamical behavior: *chaotic* and *frozen* [33]. It should be mentioned that all the properties of the Boolean networks that we consider here are derived by averaging over ensembles of networks. This is because we are investigating common properties of the ensemble of networks and not the characteristic of a specified network. One of the features that distinguishes these two phases is the distribution of network attractor lengths [12]. The sequence of a Boolean network states of a network of size N is eventually periodic, and the network dynamics will always eventually settle, in finite time, onto a periodic attractor. This directly follows from the facts that (i) the overall number of possible network states, 2^N, is finite and (ii) the dynamics of the state of the nodes are deterministic. The set of states which end on an attractor including the attractor's states is called the *basin of attraction.* The states that are not on the attractor itself but are in the basin, are called *transient*



	A	B	C
00	0	0	0
01	0	1	1
10	0	1	1
11	1	1	1

Figure 1.2: Attractors and basins of attraction of a small Boolean network. A Boolean network (left panel, top) with 3 nodes, $N=3$, and two in-links per node, $K=2$, is updated by the functions A, B, and C which are shown (left panel, bottom). Each node receives 2 in-links from the other 2 nodes of the network. The black double arrow lines between the nodes describe a pair of oppositely oriented directed links. The corresponding network state space is shown (right panel). The size of the network state space is $2^N=2^3=8$, and each of the network states are shown. Grey nodes have the output value of 0 and white ones have the output value of 1. The red arrows show how the network states are updated. There are three attractors, two of which are fixed points (attractors of length 1). The largest basin of attraction includes 5 states, 4 of which are transients and the attractor length is 1. The largest attractor length is 2 and the size of its basin of attraction is 2 as well.

states. Figure 1.2 illustrates these concepts for a small Boolean network.

In the chaotic phase, the attractor distribution is sharply peaked around an average period that grows exponentially with the network size. In the frozen phase, on the other hand, the distribution of attractor lengths is independent of the network size. At the *critical* point between the two dynamical phases, the attractor length distribution obeys a power law distribution. It should be noted that this is a generalization of the concept of phase transition from equilibrium systems to non-equilibrium systems.

It is also possible to distinguish between the different dynamical phases of random Boolean networks by their sensitivity to perturbations. The sensitivity to perturbations can be measured by the *normalized Hamming distance* [50]. Two copies of a network realization are updated for two slightly different initial states. The normalized Hamming distance between the two networks is defined as the number of nodes that are in a different state divided by the total number of nodes, N. Therefore, the normalized Hamming distance between two copies that are perturbed away from each other, after t updates, h_t, is

$$h_t = \frac{1}{N}\sum_{i=1}^{N}(s_i^1(t) - s_i^2(t))^2. \tag{1.2}$$

If h_t is small, the probability that one input of a node differs in the two copies can be neglected, and the change of h_t during one time step is given by

$$h_{t+1} = \lambda h_t \tag{1.3}$$

where λ is called *sensitivity* [89] which is a measure of sensitivity to perturbations of network states. In principle, in order to see the effects of perturbation of a network's state, the initial number of nodes states perturbed is not important. However, in practice, the networks sensitivity to perturbations can be studied by perturbation of one node's state and studying how the perturbation propagates in the network with time.

The frozen phase is characterized by small sensitivity, $\lambda < 1$. In this case, a perturbation of a network's state, achieved by flipping a node's value, propagates to less than one other node on average per time step. In this phase, the output states of the vast majority of nodes become frozen, after some transient time, and a perturbation of a network's state will eventually die out. The number and lengths of the attractors of the network are not influenced by these frozen nodes and as a result the attractors are short in the frozen phase.

On the other hand, in the chaotic phase the sensitivity is large, $\lambda > 1$, and a perturbation of a network's state, achieved by flipping a node's value, spreads on average to more than one node per time step. In this case, even after long times the majority of the nodes are not frozen and still their output states change. Thus, attractor lengths can be very long, incorporating a large fraction of the whole state space.

On the boundary between these two phases, where $\lambda = 1$, the dynamics is *critical*. In this case, a perturbation in a network's state neither grows nor shrinks on average. Critical networks have always been of special interest because of their robustness and evolvability, features considered fundamental to biological networks [102]. They are also of particular interest in applications, since, for example, they can store and transfer more information than frozen networks with static outputs or chaotic networks with random outputs [77]. Figure 1.3 is shows how normalized hamming distance and sensitivity behaves in different phases.

As mentioned above, critical networks have a broad, power law distribution of attractor lengths [12, 20]. Power law distributions are scale free. Scale free behavior is characteristic of condensed matter systems at continuous phase transitions, at which there are no characteristic lengths or time scales [108]. As a result, unlike away from criticality, only features that do not involve characteristic scales, such as the symmetry properties and the dimension of the system, can be relevant to the scaling behavior of the system. That is, at criticality, many of the details of the system become irrelevant to its behavior. Because of this, systems that have the same symmetry and dimension, but might differ in other ways, will display the same essential behavior. This phenomenon is known as universality. Systems or models that have the same behavior at the critical point are said to be in the same universality class. To quantify critical behavior the scaling properties of the system are studied. The scaling properties are quantified by critical exponents. A critical exponent is the exponent of a power law distribution observed at the continuous phase transition. For instance, the critical exponents at the liquid-gas critical point is independent of the chemical composition of the fluid. They are the same for both Nitrogen, N_2, and Methane, CH_4 [28]. A main goal of this work is to determine if universal behavior exists in heterogenous complex systems and if so to find a universality class of models.

Critical networks can be constructed by biasing the output values of the functions towards homogeneous outputs. Biasing the output values of a function means independently choosing each of the output values of the function to be 1 (or 0) with probability p, or 0 (or 1) with probability $1 - p$. If $p = 1/2$, then all the different functions are equally likely to be constructed, but if $p \neq 1/2$, there is a bias in favor of constructing functions whose outputs are more homogeneous. Critical random Boolean networks can be constructed by randomly choosing Boolean functions from those constructed with a critical bias, p_{crit}. The critical bias p_{crit} is the bias value

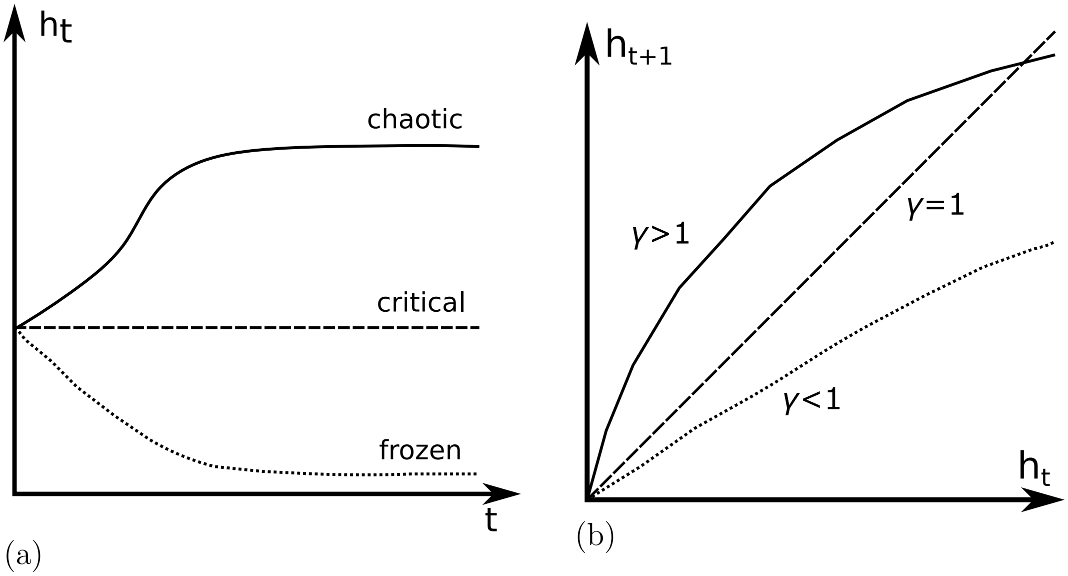


Figure 1.3: Schematic plots of the time evolution of normalized Hamming distance and sensitivity. Panel (a) shows schematically how the normalized Hamming distance evolves with time for different dynamical phases. Panel (b) shows sensitivity for different phases.

that satisfies the equation [32]

$$1 = 2Kp_{crit}(1 - p_{crit}). \tag{1.4}$$

This criticality equation was derived by Derrida and Pomeau [32]. In order to derive this equation, they introduced the annealed approximation method where the in-links and the Boolean function of each node are reassigned at every time step. This neglects possible correlations between nodes. They then considered the evolution of the distance between two configurations of the same network, starting from two random initial conditions. The annealed approximation of the evolution of this distance has been shown to be exact in the limit of large system sizes even when there are correlations between nodes [54]. Setting the distance to be 1 at the boundary between the frozen and chaotic phases results in Eq. 1.4 for the criticality

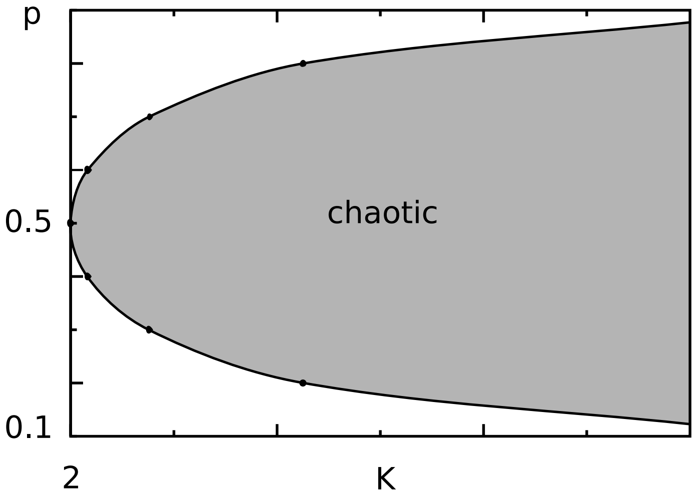


Figure 1.4: Phase diagram of random Boolean networks. K is the average in-degree, and p is the bias of Boolean functions. The grey area corresponds to the chaotic phase, while the white area show the frozen phase. The line separating two phases is the critical line given by Eq. 1.4.

condition. From this equation one can determine the critical bias, p_{crit} for a given K, or the critical average in-degree for a given bias. Figure 1.4 shows the phase diagram of random Boolean networks and the critical line corresponding to Eq. 1.4.

Another order parameter that can be used to determine the phase transition between the two dynamical phases and hence deriving the criticality condition is the size of the stable core, introduced by Flyvbjerg [38, 37]. The stable core is the fraction of nodes that evolve to constant states, and their states do not change after some transient time. In the chaotic phase, in the limit of an infinite system, the size of the stable core tends to a value less than 1, and it tends to 1 in the frozen phase. The critical point is where this sharp transition happens. Both approaches predict the same location for the critical line.

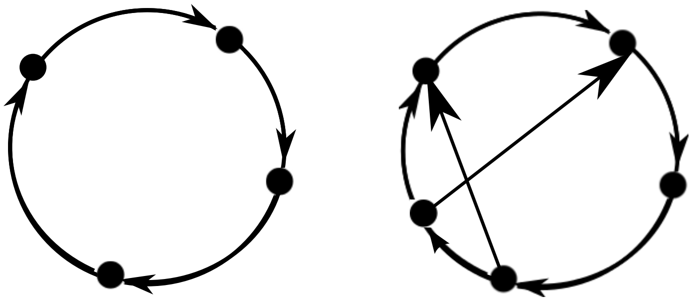


Figure 1.5: A simple loop vs. a complex loop. The left loop is an example of a simple loop with no additional links. Each node only receives an input from the preceding node on the loop. The right loop is a complex loop with additional links.

Not all nodes of the network contribute similarly to its dynamics [60]. Some nodes have fixed outputs at all time steps, after the transient time, regardless of their inputs, and are called frozen nodes. Non-frozen nodes can be either irrelevant or relevant to the dynamics: A relevant node influences at least one other relevant node, and is independent of the behavior of the irrelevant ones; while the behavior of the irrelevant nodes are entirely determined by the relevant nodes. The concept of relevant nodes was introduced by Flyvbjerg and Kjær [39]. They realized that it was only those nodes that fully determine the network dynamics. Moreover, the set of relevant nodes can be partitioned into components that are disjoint subgraphs of the network, the *relevant components*, and the dynamics of each relevant component is independent of the others. [60]

It has been shown that in critical networks, in the limit of large network sizes, the number of relevant nodes scales as $N^{1/3}$ where N is the number of nodes [68]. A finite number of relevant nodes have two relevant inputs, inputs from relevant nodes, and a vanishing fraction of them have more than two relevant inputs [68]. The relevant nodes naturally form themselves into groups called components that are independent of each other [60]. It also follows that all relevant components except a finite number of them form simple loops without additional links, also known as simple components. In the limit of large network sizes, it is only the largest loop that is not simple, i.e. it has additional cross-links [60]. Figure 1.5 shows an example of a simple loop and a complex loop.

In this dissertation, we seek the answers to the questions mentioned earlier. We want to find the role symmetry plays in the dynamics of heterogenous complex systems, to determine if there exists a universality class of heterogenous complex systems, and, if so, to characterize its characteristic universal scaling behavior. For this we study the statistical mechanics of Boolean networks and a generalization of them as prototypical examples of heterogenous complex systems. Our interest is the mathematical physics of their critical properties. We first investigate the symmetry of critical random Boolean networks dynamics. Then we study a specific critical property of random Boolean networks that scales, namely attractor length distribution, and find what its universal critical exponent is. In order to determine if the critical behavior of random Boolean networks falls into a universality class, we study the critical behavior of discrete multi-state networks [107], which are a generalization of random Boolean networks, and compare their results with that of Boolean networks.

In chapter 2, we study a manifestation of the symmetry of critical random Boolean networks dynamics. Critical random Boolean networks with two and three inputs per node are studied separately. Such networks with a uniform distribution of Boolean functions are the most widely studied examples of Boolean networks. We investigate the frequency of Boolean functions of the active nodes of the network. In order to find the active nodes of the network, we implement a pruning algorithm that allows us to calculate the frequency of different function types on the active nodes of the networks. This is done both analytically and numerically. The analysis of different function types reveals a symmetry of the critical random Boolean networks dynamics. There are classes of functions that occur with the same frequency.

In chapter 3, we investigate the nature of the symmetry observed in chapter 2. We find the characteristics of the functions that are in the same class, and propose a symmetry group that corresponds to this symmetry. The newly proposed symmetry group preserves canalization values of the functions. Canalization values show how robust a function is to perturbations of its input values. The canalization preserving symmetry group for any in-degree K will be described, and the smallest group for $K=2$ will be found.

In chapter 4, we generalize random Boolean networks to discrete multi-state networks with more than two possible output values per node. We look at the criticality condition of discrete multi-state networks and introduce a new method to describe the phase space of discrete multi-state networks. We calculate the frequency of multi-state functions of the active nodes of the network with a variant of the algorithm used in chapter 2, and show that canalization preserving symmetry group is the symmetry group that can be used to describe the symmetry of critical discrete multi-state networks. This suggests the existence of a universality class.

In chapter 5, we explicitly show that a universality class exists by studying a particular scale invariant property of critical random Boolean networks and random multi-state networks, and showing that its scaling behavior is the same for a range of models. The attractor length distribution of critical random Boolean networks is studied which obeys a power law distribution. We show that this scale-free behavior is not exclusive to critical random Boolean networks dynamics, and we can see it in other dynamics of random Boolean networks too such as an evolutionary game played on the nodes that selects for diversity [78]. Discrete multi-state networks show scale free power law distributions at the critical point as well. More importantly, we show that the critical exponent of this distribution is 1 for all of these cases in the limit of large network sizes. This confirms the existence of a universality class of heterogenous complex systems. Furthermore, we present a plausibility argument to explain this universal behavior.

Finally, in chapter 6 we present a short summary of the results in the thesis.

Chapter 2

Function classes in critical random Boolean network dynamics

2.1 Introduction

In this chapter we investigate the structure of the critical dynamics of random Boolean networks. The dynamics of each node in a random Boolean network is controlled by an independent, random chosen Boolean function. Specifically, we will show both numerically and analytically that nodes contribute differently to the dynamics, depending on which Boolean function they have. Furthermore, we will show that there exist classes, or sets of Boolean functions that cause the nodes whose dynamics they control to contribute equivalently to the critical dynamics when averaged over a random Boolean network ensemble.

As defined in chapter 1, nodes relevant to the dynamics of random Boolean networks have been studied previously. For instance, the scaling behavior of the number of nonfrozen, and relevant nodes of critical random Boolean networks have been analytically studied [61, 68]. It was shown in previous work that the number of nonfrozen nodes for any critical network with the in-degree greater than one, scales as $N^{2/3}$, where N is the total number of nodes in the network, while the number of nonfrozen nodes with two nonfrozen inputs scales as $N^{1/3}$ [61]. Furthermore, the mean number of relevant nodes increases as $N^{1/3}$, with only a finite number of relevant nodes having two relevant inputs, and a vanishing fraction of nodes having more than three of them [68].

We will further this investigation of relevant nodes by analyzing the distribution of functions associated with such nodes, along with the canalization properties of these functions. As was discussed in chapter 1, since relevant nodes determine the dynamics of critical random Boolean networks, this investigation helps to understand the symmetry and dynamical properties of these networks. We will show that the distribution of the Boolean functions depends on how canalizing the functions are. *Canalization* [101] has its roots in the study of evolutionary biology and is thought to be a crucial feature that is helpful in explaining the stability of prototype expression in biological systems. The concept was formed based on experimental observations that certain developmental traits appear to control the expression of other traits. Canalization suppresses the changes in phenotype expression that require development to deviate from the canalized pathway. It has been used extensively as a form of robustness in various network models. In the study of random Boolean networks, canalization has to do with some inputs controlling the output regardless of the remaining inputs [84]. It refers to the robustness of functions under perturbations of their inputs; for instance, the extreme case of a constant function would be said to be *highly* canalizing. Since relevant nodes control the dynamics of critical networks, the question of how canalizing the relevant nodes of a critical Boolean network are, is indeed an important one.

In this chapter, we analytically and numerically derive the distribution of update functions on the nonfrozen core of critical random Boolean networks for two cases of networks with $K=2$, and $K=3$ inputs per node. We use a variant of the method that was first introduced by Flyvbjerg [38] and later developed by Kaufman et al. [61] to find the nonfrozen nodes: starting from the nodes with constant functions, the frozen core of the network can be determined iteratively. By setting up the corresponding differential equations for this process, the average number of nonfrozen nodes with a particular update function is found. Then, considering the effects of fluctuations, the probability distribution of the update functions on the nonfrozen core is derived.

The reason that we consider the nonfrozen or active nodes of the networks rather than the relevant nodes is that, as It will be shown below (Fig. 2.4), the relative frequency of functions of the relevant nodes with respect to each other is the same as the relative behavior of functions of the active nodes. Therefore, instead of considering relevant nodes, we can simply focus on the properties of functions of the active nodes. This simplifies the calculations and reduces the simulations times.

We discover that classes of functions exist that have identical dynamical behavior. It turns out that the functions that are in the same class have the same canalization properties, hence creating canalization classes. We will first consider critical Boolean networks with $K=2$ inputs per node, and then investigate the case of $K=3$ inputs per node.

2.2 Critical random Boolean networks with in-degree two

2.2.1 Finding the active nodes

As stated in chapter 1, heterogeneous composition is often a feature of complex systems that distinguishes them from more traditional condensed matter systems. This heterogeneity makes them complicated, but, as we will now see, also allows for novel methods of analysis that can provide a fundamental understanding of their behavior. In particular, one can look for symmetry among the heterogeneous components of a complex system to find the nature of its dynamics.

In Boolean networks, one aspect of their heterogeneity is that nodes can behave differently for identical sets of inputs. The responses of the nodes are dictated by their output functions, which are heterogeneously distributed. Moreover, various Boolean functions are generally utilized with different frequencies, and nodes with different Boolean functions can play different roles in the dynamics of a network. To look for a manifestation of symmetry in the dynamics in the critical state of random Boolean networks, we will consider the distribution of Boolean functions utilized by the nodes.

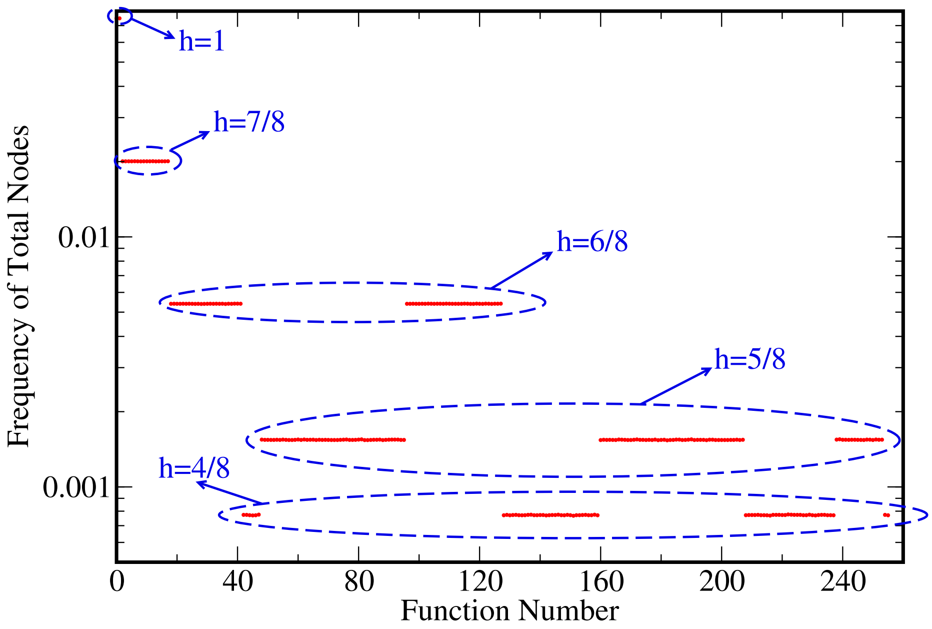


Figure 2.1: Ensemble-averaged frequency of the Boolean functions used by all nodes of critical random Boolean networks with in-degree $K = 3$. Functions that have the same frequency have the same homogeneity value. Five different homogeneity values h are possible, as shown. The frequencies are normalized such that the sum of the frequencies of all of the functions is unity.

We have simulated ensembles of networks constructed to be critical, by randomly forming the function of each node with critical bias while maintaining the parity between 0 and 1. This parity is achieved obtained by randomly choosing to bias the output of each node's function toward a homogeneous output of either 0 or 1, with equal probability. This form of bias toward homogeneity of the output functions is used, for simplicity, to maintain an overall symmetry between 0s and 1s in all aspects of the networks' structure and dynamics. Also, all networks that we consider here have simple directed graphs describing the interactions of the nodes; that is, the nodes have no self-interactions and receive at most one input from any other node.

Figures 2.1 shows numerical results for networks of 10^5 nodes each with $K=3$ inputs, averaged over an ensemble of 10^6 networks. There are 256 possible Boolean functions with 3 inputs, each of which occurs with a certain frequency. In this figure, the functions are numbered from 0 to 255, in a manner that groups related functions together [64].

The distribution of Boolean functions among all nodes of the network is shown in Fig. 2.1. Five different frequencies are possible due to the internal homogeneity of each function. Internal homogeneity, which is a measure of how homogeneous the output of the function is, is defined as the fraction of the 2^K outputs of a function that are 0 or 1, whichever is larger [103]. For Boolean functions with 3 inputs, there are $2^3=8$ possible input sets, and the homogeneity of functions can take five values: 1, $^7\!/_8$, $^6\!/_8$, $^5\!/_8$, and $^4\!/_8$. The five different classes of functions occur according to the binomial distribution of 1s and 0s and the imposed symmetry between them. If the homogeneity of a function is h, then its frequency is

$$\mathrm{Freq}(h) \;=\; \frac{1}{2}\binom{8}{8h}\left[p_c^{8h}(1-p_c)^{8-8h}+p_c^{8-8h}(1-p_c)^{8h}\right].$$

We have shown the results of Boolean functions with 3 inputs, and not 2 inputs here because the higher number of different possible frequencies for networks with $K=3$ makes it easier to see the class structure. There are 3 different frequencies possible due to the internal homogeneity of each function for networks with $K=2$. The corresponding homogeneity values of the functions are 1, $^1\!/_2$, and $^3\!/_4$.

Not all the nodes, however, have the same importance for the dynamics of the network. Some nodes never change their output states, either because they have completely homogeneous output functions, or because they receive input from functions with homogeneous output functions. These frozen nodes are irrelevent to the dynamics of the network [14, 90, 60, 87]. This suggests that in order to find symmetry in the dynamics of the network, analysis should focus on the behavior of the non-frozen, *active* nodes of the network and on their functions.

In order to find the active nodes of the network, which were defined in chapter 1, all the frozen nodes of the network must be found and eliminated. Nodes can become frozen either by getting inputs from constant or previously frozen nodes or by forming self-freezing loops. Self-freezing loops are loops of nodes with canalizing functions that can fix each other on their canalizing value, and therefore form frozen nodes [83]. It will be discussed later that if we develop critical random networks by biasing the outputs toward homogeneity, self-freezing loops will not form. Therefore, we only consider the first mechanism of frozen nodes formation. Thus, the frozen nodes in a given network realization are found by first finding the nodes with constant functions and then find the nodes that become frozen by receiving inputs from them. These newly frozen nodes themselves can freeze some more nodes, and the process repeats iteratively until the entire frozen core of the network is determined.

We consider an ensemble of networks of size N which are biased toward homogeneity by the critical bias, which is $p_{crit} = 1/2$ for $K = 2$. The strategy to describe this process analytically is to find the effects of a randomly picked frozen node at each time step of the process on other nodes. To do so, we employ the stochastic *pruning method* used in [61]. Here we use a variant of this stochastic pruning model. Unlike the original variant of the model, which tracked only the number of active nodes through the pruning process, we track the independent progress of nodes with various output functions. This is essential for our purposes because at the end, we want to determine the distribution of output functions of the active nodes, and not merely the number of active nodes.

First, we set up 16 urns, numbered from 0 to 15, one for each of the $K = 2$ Boolean functions. We call them *original* urns. Original urn i contains all the functions numbered i. We also have an urn called the *frozen* urn, F, where the frozen nodes are put. The nodes with constant functions are obviously frozen because constant functions give constant outputs at all time steps. Therefore, the initial number of nodes in urn F is equal to the initial number of nodes with constant functions. The initial number of nodes in other original urns is the same as the initial number of nodes with each of those function types.

Using these urns, the pruning process is defined as follows: At each step of this process, we remove a node from the frozen container F and compute its average effect on the other nodes. Based on the canalization properties of each of the function types, the nodes that receive input from this frozen input will either become frozen and go to container F or will remain unfrozen but effectively become a function with one free input. If such a function receives another frozen input, it has no other choice than to freeze. After considering all the effects of the chosen frozen node, it is eliminated from the frozen urn. The process ends when F is empty or all the nodes are frozen, and then whatever is left in the rest of the urns are the active nodes.

To address the question of how canalizing the active nodes of the network are, we need to find how many of the active nodes functions of the network are *originally* from each of the function types. In order to do so, we keep track of the initial type of each of the nodes as they move from urn to urn. Hence, we should introduce *intermediate* urns: $\{iK_1;\ i{=}1\ \ldots\ 15\}$. iK_1 is the urn corresponding to the $K=1$ functions that were initially in i urn with function type i, and received a frozen input but did not become frozen. Initially, all these intermediate urns are empty.

After the process is finished, the nodes that are left in the urns (intermediate and original) represent the nonfrozen or active nodes of the network. To determine the frequency of active nodes with each of the function types, we sum the number of nodes in the corresponding original and intermediate urns of a given function type. For example, the nodes left in urn 4 plus the nodes left in urn $4K_1$ represent the active nodes of the network that originally had number 4 functions. The difference between the nodes left in these two urns is that active nodes in the original urn 4 are still $K=2$ Boolean functions, whereas, the active nodes in $4K_1$ urn have been effectively converted to $K=1$ Boolean functions by receiving a frozen input. Functions in $4K_1$ urn have one active input, while functions in urn 4 have two. The numbering of the functions is based on converting the binary output of the functions to decimal numbers and is shown in Fig. 3.1 for $K=2$ Boolean functions.

The exact probability of each of these transitions from one urn to the other is easily computed if we consider the geometric representation of functions. In addition to defining a Boolean function by its output values, discussed in chapter 1, Boolean functions have a geometric representation as well. The set of Boolean functions with K inputs maps one-to-one onto the set of configurations of the K-dimensional hypercubes with vertices that have binary states [84]. According to the geometric

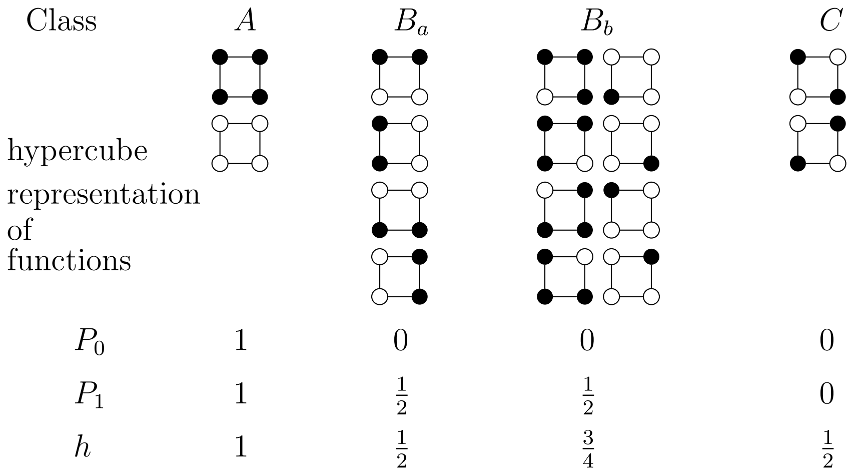

Class
A
B_a
B_b
C
hypercube representation of functions
P_0 1 0 0 0
P_1 1 $\frac{1}{2}$ $\frac{1}{2}$ 0
h 1 $\frac{1}{2}$ $\frac{3}{4}$ $\frac{1}{2}$

Figure 2.2: Hypercube representation of $K=2$ Boolean functions. P_0 is the probability that the function is canalizing by zero inputs, i.e. it is constant. P_1 is the probability that the function is canalizing by one input (or by one input value). We can define the total canalization value of a function as $P=\Sigma P_k$. In this definition, functions in class A ($P=2$) are more canalizing than class B_a and B_b ($P=\frac{1}{2}$), and these two classes are more canalizing than functions in class C ($P=0$).

representation, Boolean functions with two inputs are represented by squares, i.e. , two dimensional hypercubes. Figure 2.2 illustrates the hypercube representation of the 16 $K=2$ functions. The functions are categorized based on their canalization values and homogeneity. Function categories of functions are labeled as function classes A, B_a, B_b, C. These different classes correspond to the group orbits of a symmetric group, called Zyklenzeiger group, where hypercube rotations and parity are the operators of the group [10]. Geometrically, it means that Different functions that belong to the same class have the geometric property that they can be rotated into each other by symmetry operations on the K-dimensional hypercube plus parity. This symmetry group is important in an evolutionary game played by nodes of random Boolean networks which selects for diversity of the nodes behavior [78].

We use this evolutionary game through this dissertation as an example of a different dynamics of random Boolean networks and compare its result with that of random Boolean networks dynamics. Because of this, we use the Zyklenzeiger categorization in most of the figures of this chapter.

The internal homogeneity and the $P_k (k=0,\ 1)$ values for each of the classes are also shown in Fig. 2.2. Internal homogeneity, which is a measure of how homogeneous the output of the function is, is defined as the fraction of the 2K outputs of a function that are 0 or 1, whichever is larger [103]. For a Boolean function with K inputs, P_k $(k = 0,\ 1,\ \ldots,\ K-1)$ is a measure of how canalizing the function is. It is defined as the probability that the output of the strategy is determined if k randomly chosen inputs are fixed, but the other inputs are allowed to vary. In the hypercube representation, P_k is the fraction of $K-k$ dimensional hypersurfaces of the hypercube that are homogeneously colored. For example, for $K=2$ functions, P_0 (which is the probability that a function is canalizing by zero inputs, or in other words, the probability that it is constant) is the fraction of homogeneously colored $K-k=2-0=2$ dimensional surfaces. P_1, which is the probability that a function is canalizing by one input, is the fraction of homogeneously colored $K-k=2-1=1$ dimensional surfaces (sides of the square) to the total number of one dimensional surfaces (sides).

The probability that a $K = 2$ function becomes frozen by receiving a frozen input is the probability that the function is canalizing by one input or by that specific input value (P_1). In hypercube representation, this is given by the fraction of homogeneously colored sides of the square. The fraction of non-homogeneously colored sides represents the probability that the function does not freeze and instead effectively converts to a $K = 1$ function and moves to the appropriate intermediate urn. Figure 2.3 shows these transition probabilities for each of the classes.

Neglecting the effects of fluctuations for now, we can write the equations that describe the average change of the number of nodes in each of the urns. The equations for a sample function, function number 10, are:

$$\begin{aligned}
\Delta N_{10} &= -2\frac{N_{10}}{N} \\
\Delta N_{10K_1} &= \frac{N_{10}}{N} - \frac{N_{10K_1}}{N} \\
\Delta N_F &= -1 + \frac{N_{10}}{N} + \frac{N_{10K_1}}{N} + \dots \\
\Delta N &= -1
\end{aligned}$$

The average number of nodes in urn i at a given time step is shown by N_i, and the total number of nodes is shown by N. To clarify these equations a bit more, let us calculate how the average number of nodes changes in each urn. A number 10 function type, which is a $K = 2$ Boolean function, chooses the randomly selected frozen node from container F with probability $2(\frac{1}{N})$ ($\frac{1}{N}$ is the probability of choosing the frozen node among all possible N nodes of the network at that time, as an input.). Assuming that at the moment there are N_{10} nodes in urn 10, $2(\frac{N_{10}}{N})$ nodes leave the urn as a result of this input choice. As depicted in Fig. 2.3, the probability that choosing a frozen input freezes a function in B_a class, which is the probability that the function is canalizing by one input, is $\frac{1}{2}$. In terms of the Zyklenzeiger group classification, number 10 function type is in class B_a. Therefore, $\frac{1}{2} \cdot 2(\frac{N_{10}}{N})$ of the

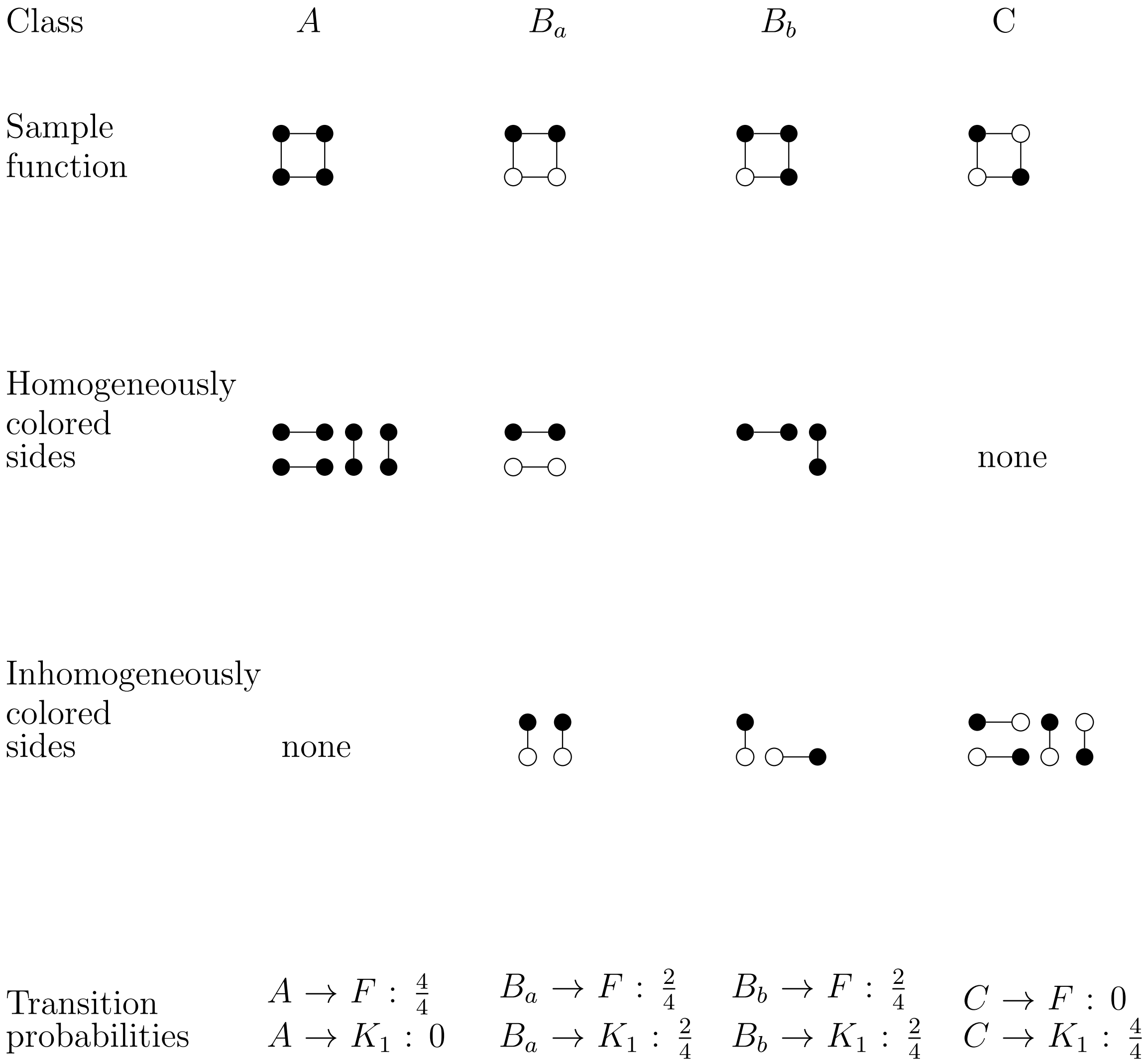

Class
A
B_a
B_b
C
Sample function
Homogeneously colored sides
none
Inhomogeneously colored sides
none
Transition probabilities
$A \to F : \frac{4}{4}$
$A \to K_1 : 0$
$B_a \to F : \frac{2}{4}$
$B_a \to K_1 : \frac{2}{4}$
$B_b \to F : \frac{2}{4}$
$B_b \to K_1 : \frac{2}{4}$
$C \to F : 0$
$C \to K_1 : \frac{4}{4}$

Figure 2.3: The transition probabilities of the pruning process. The functions that have the same canalization values have the same transition probabilities.

nodes that have chosen the frozen input will become frozen, and are transferred to the frozen container. The other half, that are not frozen, but have converted to effective $K = 1$ functions will be transferred to the $10K_1$ urn. The same logic is applicable for the other function types as well. Also, we note that at each time step, a frozen node is eliminated from the network, therefore, we have $\Delta N = -1$.

Dividing all the above equations by $\Delta N = -1$, in the limit of large N, we get the set of deterministic differential equations for the average number of nodes in each of the urns:

$$\begin{aligned} \frac{dN_{10}}{dN} &= \frac{N_{10}}{N} \\ \frac{dN_{10K_1}}{dN} &= \frac{N_{10K_1}}{N} - \frac{N_{10}}{N} \\ \frac{dN_F}{dN} &= \frac{N_F}{N} + \dots \end{aligned} \tag{2.1}$$

These differential equations are written for all the 16 Boolean function types, as is shown in appendix A. It gives a set of first order ordinary differential equations that can be easily solved by the standard methods. Figure 2.4 shows the ratio of the number of Boolean functions occurring on nodes that are active to the number at which they occur on all nodes. Investigating this ratio, rather than the frequency of the Boolean functions on active nodes, allows the effect of the initial conditions for each fixed bias to be eliminated, and the functions to be properly compared. It also shows the results of solving Eqns. 2.1. Both results are averaged over an ensemble of critical networks of size $N = 100\,000$. One way to create a critical network is to bias the outputs of the update functions toward homogeneity when defining each function:

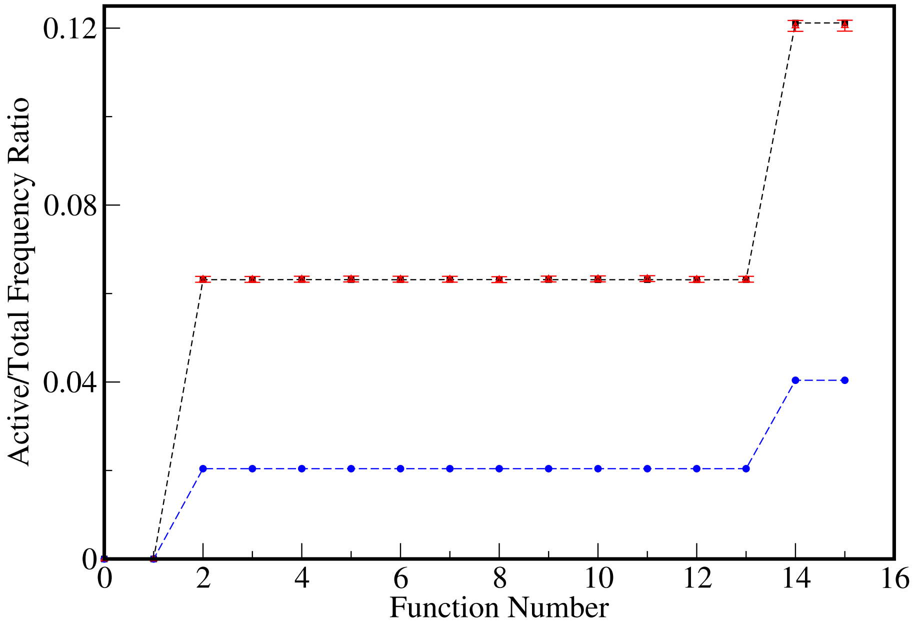


Figure 2.4: Comparing direct simulations with deterministic equations results. The black circles and the dashed line passing through them is the result of direct simulations of active nodes of the network, and the blue circles show the results of calculating Eq. 2.1. The red triangles show the results for the relevant nodes of the network. The error bars are 2 times standard deviation over network realizations. The procedure of calculating the active to total frequency ratio for each of the function numbers is as follows: The final number of each type of function is calculated for each method accordingly (direct simulations, solving differential equations, . . .). Then this number is divided by the initial number of that function type in the initial setting of the system, e.g. the initial number of each function type before pruning for the active nodes functions. Then these ratios are normalized by their sum. An ensemble of 5000 critical networks each of which have $N = 100\,000$ is considered. The functions are numbered from the most canalizing (function number 0) to the least canalizing (function number 15), which is not the numbering shown in Fig. 3.1 and used in the differential equations.

for any combination of inputs, 1 or 0 is assigned as an output with probability p, and the other output value with probability $1-p$. Critical random Boolean networks occur when the relation $2Kp_{crit}(1-p_{crit})=1$ is satisfied [32]. For the case of $K=2$ functions, this equation is solved when $p=\frac{1}{2}$ (i.e. when the update functions are chosen completely randomly).

In addition to comparing the results of direct simulations of active nodes, *real networks*, with the results of mean-filed approximation, Fig. 2.4 shows the results of direct simulations of relevant nodes. The number of relevant nodes functions is calculated by adding one pruning step further to the real networks simulations [61].The two direct simulation results are within each other's error bars, and this means that for our purposes instead of focusing on the relevant nodes, we can look at the behavior of the functions on the active nodes of the network. In this figure, the functions are ordered based on their canalization values, from the highest canalizing to the lowest canalizing. It clearly shows that there exist function classes that occur with the same frequency. For $K=2$ there are three function classes and the functions that are in the same class have the same canalization values. Moreover, it can be seen that the active nodes with more canalizing function types occur less frequently.

The mean-filed approximation captures the overall behavior correctly, but as it does not predict the total number of active nodes correctly [61], neither does it predict the frequency of each class accurately.

It should be noted that if one looks at the differential equations written for the change of the average number of nodes of each function type (see appendix A), without even solving the equations, it can be seen that the functions that have the same canalization values, also have the exact same equations. This confirms the numerical results of the existence of function classes. Therefore, in the rest of this chapter, instead of writing the equations for each function type separately, we write the equations for each function class, and we have an urn for each function class and not each function type. This will prove particularly useful as K increases, since as K gets larger the number function types grow exponentially as 2^{2^K}, hence the number of differential equations to study. Also, in order to compare the results of Boolean network dynamics with that of the evolutionary game mentioned earlier in this chapter, we consider the Zyklenzeiger function classes for writing the equations. For $K=2$, there are 4 Zyklenzeiger classes, labeled as A, B_a, B_b, and C. They are a subset of the 3 function classes we discovered for critical random Boolean networks dynamics, labeled as A, B, and C. Classes A, and C are the same in both cases, but class B divides into to classes in the Zyklenzeiger group categorization.

2.2.2 Potential problems

It is important to note that although we have obtained our numerical results here, through an implementation of the pruning process, one could simply naively look for the active nodes of a network by following the output states of the nodes on the network's attractors. Critical random Boolean networks can have multiple attractors [88, 13, 12, 21, 90]. The active nodes are the ones whose output values change over the course of some attractor. Hence in principle, one should examine the behavior of the nodes on all of the attractors to find the active ones. If this is done, an identical number of function classes should be detected through such a naive approach. On some attractors though, otherwise active nodes may become frozen through dynamical processes not associated with pruning [71], for example, by forming so-called self-freezing loops [83]. Therefore, the total number of active nodes found by using the naive approach may still be less than that found with a pruning process. Under particular conditions concerning the initial distribution of Boolean functions among the nodes, which are not met in the case we have considered here, self-freezing loops can be an important contributor to the frequency of active nodes. For all other initial function distributions, the contribution from self-freezing loops vanishes in the large network limit [68]. In this case, the difference between using a pruning process and using the naive approach is negligible, and the vast majority of the frozen nodes are found by the iterative pruning process.

Also, we mentioned that the pruning process continues until there are no frozen nodes left in the frozen urn. It has been recently argued [71] that this boundary condition might be too strict, and for K higher than 4 it does not find all the frozen nodes. Nevertheless, it has been shown [71] that the qualitative features of the found frozen core by the pruning process is the same as the actual frozen core. Therefore, this does not affect our conclusions regarding the canalizing properties of the active nodes for Ks higher than 4.

2.2.3 Effects of fluctuations

In section 2.2.1 we neglected the effects of fluctuations and instead focused on a mean-field description of the process of finding the nonfrozen nodes of the network.

However, the effects of fluctuations are clearly important as can be seen from the results in Fig. 2.4.

Since the number of nodes that are transferred from an urn to another one occurs with a known average rate, and each transfer is independent from another one at a different time step, the number of nodes that are transferred from an urn to another one at each time step has a Poisson distribution around that mean value. For example, the number of nodes that leave urn B_a, and choose a frozen node as input is Poisson distributed around the mean of $2(\frac{N_{B_a}}{N})$ with a variance $2(\frac{N_{B_a}}{N})$. The same is true for all the other urns. In this section we find the probability distribution of the number of active nodes functions in each function class using the Focker-Planck equations.

Since the fluctuations are inversely proportional to different powers of N, in the limit of large network sizes that we consider in order to make the calculations easier, a noise term will be added only to the mean-field equations of the urns with the largest fluctuations (smallest powers of $\frac{1}{N}$). This gives us the Langevin equations. By solving the deterministic differential equations, it can be shown that the variances of the urns B_aK_1, B_bK_1, and CK_1 are larger than the variances of all the other urns. As the total number of nodes left in the urns gets smaller during the pruning process, $(N/N^{ini} \to 0)$, the fluctuations in these three urns become larger relative to the other urns. However, we also need to consider the fluctuations in container F. Although the variance of F is small compared to the three mentioned urns, we have to consider the effects of fluctuations in this urn due to the small number of nodes in it when the process nears completion (ie. when the size of F reaches zero). The differential equation for F becomes:

$$\frac{dN_F}{dN} = \frac{N_F}{N} + \frac{N_C}{N} + \xi$$

where ξ is the noise term. Following the same argument as presented in [61], this noise term can be considered as Gaussian distributed noise with zero mean and unit variance, and the Fokker-Planck equation for the process will be:

$$\frac{\partial P(N_F, N)}{\partial N} = -\frac{\partial}{\partial N_F}(\frac{N_F}{N} + \frac{N_C}{N})P(N_F, N) + \frac{1}{2}\frac{\partial^2 P(N_F, N)}{\partial N_F^2} \tag{2.2}$$

where $P(N_F, N)$ is the probability that N_F nodes are left in urn F when N nodes are left in total. In order to derive this equation, note that, in general, if we have a stochastic differential equation of the form

$$dX_t = \mu(X_t, t)dt + dW_t\sqrt{2D(X_t, t)}$$

then the Fokker-Planck equation for the probability density $f(x, t)$ of the variable X_t is

$$\frac{\partial f(x, t)}{\partial t} = -\frac{\partial}{\partial x}(\mu(x, t)f(x, t)) + \frac{\partial^2}{\partial x^2}(D(x, t)f(x, t)))$$

where $\mu(X_t, t)$ is called the drift coefficient, $D(X_t, t)$ is the diffusion coefficient, and W_t is the Wiener process. In this case, the drift coefficient is $(\frac{N_F}{N} + \frac{N_C}{N})$, and the the diffusion coefficient is 1/2, which gives us the above Fokker-Planck equation for the pruning process.

By changing the variable to

$$\begin{aligned} x =& \frac{N_F}{\sqrt{N}} \\ y =& \frac{N}{(\frac{N^{ini}}{\beta})^{2/3}} \\ f(x,y) =& (\frac{N^{ini}}{\beta})^{1/3} P(N_F, N) \end{aligned}$$

the Fokker-Planck Eq. 2.2.3 will become

$$y\frac{\partial f}{\partial y} + f + (\frac{x}{2} + y^{3/2})\frac{\partial f}{\partial x} + \frac{1}{2}\frac{\partial^2 f}{\partial x^2} = 0.$$

This equation can not be solved analytically [61]. But instead the change of variables can be used to numerically tackle the problem. With this change of variables $W(N)$, which is introduced [61] as the probability that N nodes are left in total when the number of nodes in urn F is zero, in terms of $P(N_F, N)$, will be:

$$\begin{aligned} W(N) &= \int_0^\infty P(N_F, N)\, dN_F - \int_0^\infty P(N_F, N-1)\, dN_F \\ &= \frac{\partial}{\partial N}\int_0^\infty P(N_F, N)\, dN_F \\ &= (\frac{N^{ini}}{\beta})^{-2/3}\frac{\partial}{\partial y}\sqrt{y}\int_0^\infty f(x,y)\, dx \\ &= (\frac{N^{ini}}{\beta})^{-2/3} G(y) \end{aligned} \tag{2.3}$$

where $\beta = \frac{N_C^{ini}}{N} = \frac{N_A^{ini}}{N}$, and $G(y)$ is called the scaling function.

Since there is a one to one relation between the total number of active nodes and the number of active nodes in each of the urns, finding $W(N)$ results in finding the probability distribution of number of active nodes of all the other urns. For urn B_a, for instance, we will have:

$$W(N)\,dN = W(N_{B_a})\,dN_{B_a}$$

Having the probability distribution functions, we can calculate the average number of active nodes of each of the classes analytically. We need to add the appropriate intermediate and original urns to find how many active nodes were originally from each function class. The following solutions are derived for all of the classes (excluding class A since all of the functions in that class are frozen):

$$\begin{aligned}
\bar{N}_{B_a} &= \int N_{B_a} W(N_{B_a})\,dN_{B_a} = \int N_{B_a} W(N)\,dN \\
&= \int (N_{B_a} + N_{B_a K_1}) W(N)\,dN \\
&= \frac{N^{ini}_{B_a}}{N^{ini}} (\frac{N^{ini}}{\beta})^{2/3} \int yG(y)\,dy \\
\bar{N}_{B_b} &= \int N_{B_b} W(N_{B_b})\,dN_{B_b} = \int N_{B_b} W(N)\,dN \\
&= \int (N_{B_b} + N_{B_b K_1}) W(N)\,dN \\
&= \frac{N^{ini}_{B_b}}{N^{ini}} (\frac{N^{ini}}{\beta})^{2/3} \int yG(y)\,dy \\
\bar{N}_{C} &= \int N_{C} W(N_{C})\,dN_{C} = \int N_{C} W(N)\,dN \\
&= \int (N_{C} + N_{C K_1}) W(N)\,dN \\
&= 2\frac{N^{ini}_{C}}{N^{ini}} (\frac{N^{ini}}{\beta})^{2/3} \int yG(y)\,dy - (\frac{N^{ini}}{\beta})^{1/3} \int y^2 G(y)\,dy
\end{aligned}$$

The average values of the number of active nodes of each of the classes depend on some integrals involving $G(y)$. Therefore, instead of numerically guessing $G(y)$ first and then integrating it, used in [61], we first simulated the stochastic differential equations for large network sizes ($N = 100\,000$), and integrated over the $G(y)$ function

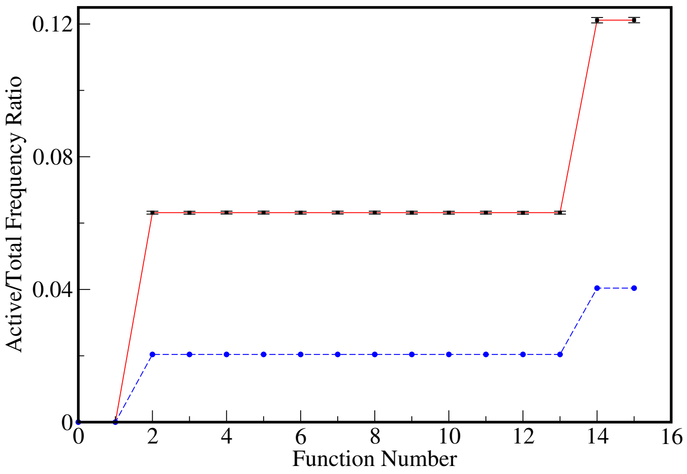


Figure 2.5: Comparison of direct simulations, mean-field results, and analytical calculations. The data in black squares are the results of direct simulations of the active nodes, and the blue circles and the line passing through them show the results of calculating Eq. 2.1. The red solid line shows the results of analytical calculations. Results are for an ensemble of 5000 networks with $N = 100\,000$. The functions are numbered from the most canalizing (function number 0) to the least canalizing (function number 15). The error bars are 2 times standard deviation over network realizations.

directly. We numerically integrated over this function that is derived from Eq. 2.2.3 to produce the values for of the various integrals. This method is expected to get more accurate results than numerically guessing $G(y)$ to find the desired integrations.

Figure 2.5 shows the results of simulating the real networks, calculating Eq. 2.1 and the analytical results. These results were computed using an ensemble consisting of 5000 critical networks with $N = 100\,000$ nodes. By increasing the size of the ensemble we get smaller error bars over the network ensemble, and by increasing the size of the networks, the analytical results get closer to the direct simulation results.

Two important points can be deduced from this graph:

i. The results derived by simulating the stochastic process, i.e., Langevin equations, are within the error bars of the real network results. This means that these two data sets are statistically similar, which confirms our method. It also indicates the importance of including the effects of fluctuations. There is an obvious discrepancy between the direct simulations and mean-field results.

ii. The frequency ratio of the active nodes functions to the initial number of those functions clearly shows that more canalizing functions appear less frequently in the active nodes of critical random Boolean networks. In fact, the only characteristic of the functions that determine their function class is their canalization values.

2.3 Critical random Boolean networks with in-degree three

There are $2^{2^3} = 256$ possible Boolean functions with $K = 3$ inputs. In order to find the function types of the active nodes, a set of differential equations for each of the function types can be written. If one does that it can be easily seen that the functions that have the same canalization values have exactly the same equations. Hence, based on our discussion in section 2.2.1, we write the differential equations for the change of the average number of nodes for Zyklenzeiger classes of functions, and show that the critical random Boolean network dynamics have a higher symmetry for $K = 3$ as well, and even some of the Zyklenzeiger classes behave similarly and converge in the light of symmetry that controls critical random Boolean networks dynamics.

According to Zyklenzeiger symmetry $K = 3$ functions can be classified in 14 different classes based on their canalization, internal homogeneity, and parity symmetry. Note that parity is necessary to distinguish classes for $K = 3$, but not for $K = 2$. Table 2.1 shows the classification of these functions with their corresponding P_k and internal homogeneity values. Figure 2.6 illustrates the hypercube representation of a sample function in each class. Each function is depicted by a cube (three dimension hypercube). The probability that a function is canalizing by one input, P_1, is the fraction of homogeneously colored surfaces, and the probability that it is canalizing by two inputs, P_2, is the fraction of homogeneously colored edges. Obviously, constant functions are represented as homogeneously colored cubes.

2.3.1 Finding the active nodes

Following the same method used for the case of $K = 2$, first we need to set up the appropriate urns to keep track of the original classes of the active nodes. We will have 14 original urns corresponding to the 14 Zyklenzeiger classes. For the case of $K = 3$, two intermediate steps are possible for a node to become frozen by receiving frozen inputs. Each of the $K = 3$ functions essentially becomes a $K = 2$ function provided that they do not become frozen by receiving a frozen input. These newly formed functions have another chance to receive a frozen input in the next step, and turn into a $K = 1$ function. The $K = 1$ functions have no other choice than to become

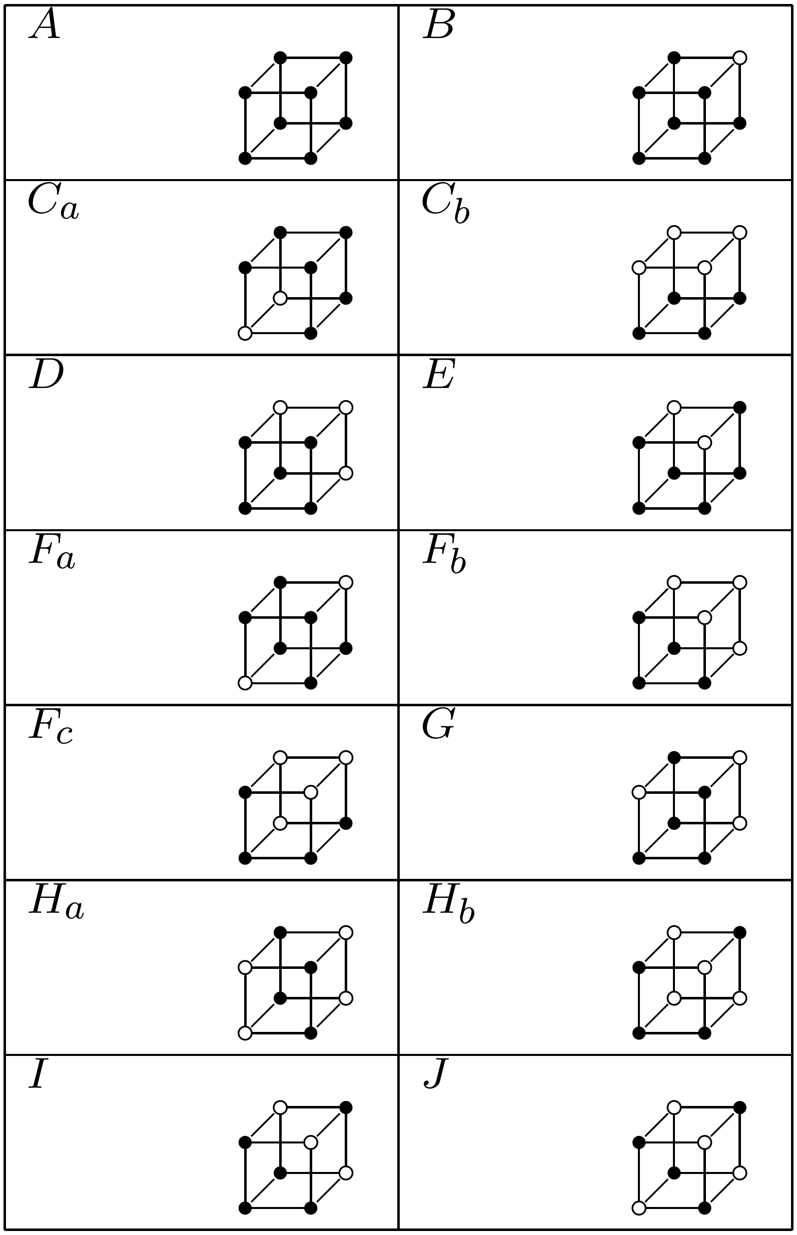


Figure 2.6: Geometric representation of a sample function from each of the 14 classes of the $K=3$ Zyklenzeiger group.

Class	A	B	C_a	C_b	D	E	F_a	F_b	F_c	G	H_a	H_b	I	J
Size	2	16	24	6	48	24	8	8	24	48	6	24	16	2
P_0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
P_1	1	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{6}$	0	0	0	0	0	0	0	0
P_2	1	$\frac{3}{4}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{7}{12}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{5}{12}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{4}$	0
h	1	$\frac{7}{8}$	$\frac{3}{4}$	$\frac{1}{2}$	$\frac{5}{8}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{5}{8}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{5}{8}$	$\frac{1}{2}$

Table 2.1: The fourteen class division of 256 K=3 functions.

frozen if they choose a frozen input. Therefore, in addition to the original urns and urn F, we have two types of intermediate urns. The first type corresponds to the functions that have received a frozen input but have not yet become frozen, and have turned into a $K=2$ function effectively. These urns will be of the form XB_a, XB_b, and XC, where X shows the initial class of the function. There will be 23 of these urns. The second type of intermediate urn corresponds to the functions having already tuned into one of $K=2$ functions in previous steps, which receive another frozen input. Those functions that still do not freeze turn into $K=1$ functions, and constitute these urns. They will be of the form XB_aK_1, XB_bK_1, and XCK_1, where X shows the initial class of theses functions. There will be 23 of these urns as well.

As before, the process of determining the frozen nodes of the network starts with transferring all the nodes with constant functions (class A) to the frozen urn F, since they are by definition frozen. Then at each time step of the pruning process, a frozen node is chosen randomly from F, and is thrown away after its effects on the rest of the network is determined. Depending on how many nonfrozen inputs are left for each of the functions, there is a probability that they choose the picked frozen node as an input, and based on their canalization properties they might freeze

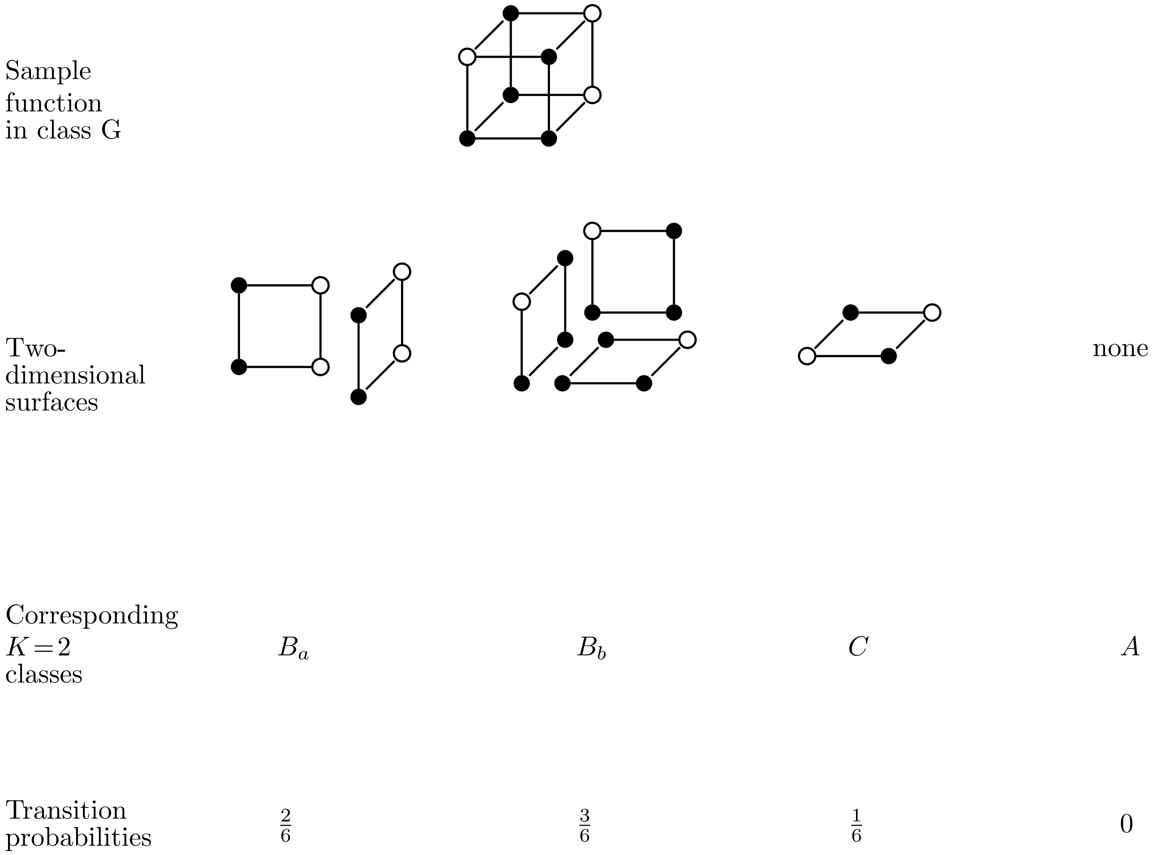


Figure 2.7: The transition probabilities of moving nodes with a sample function in class G between different urns. The functions that have the same canalization values have the same transition probabilities.

or turn into a lower K function. The probability of each of these transitions can be easily calculated by looking at the geometric representation of the functions. We will concentrate on nodes with functions originally from class G, as an example, and follow them step by step to see how they become frozen.

If a node is initially in class G, it has three nonfrozen inputs. Therefore, with probability $3(\frac{1}{N})$ it chooses the randomly picked frozen node. The probability that it becomes frozen is the probability that the function is canalizing by one input, which is the fraction of homogeneously colored surfaces of a function in class G. As depicted in Fig. 2.7, this probability is zero; hence, this class of functions does not become frozen by freezing only one of their inputs. When a $K=3$ function chooses a frozen input, it behaves effectively like a $K = 2$ function. In other words, the cube representing the $K = 3$ function is projected onto one of its surfaces, which represents one of the $K=2$ function classes. Recalling the square representation of $K = 2$ functions from the previous section, the probability of being transferred to any of $K=2$ function classes is simply the fraction of those type of function surfaces (squares) on the cube. In this case, as illustrated in Fig. 2.7, it is projected to the B_a class, and goes to GB_a urn with probability $2(\frac{1}{6})$. With probability $3(\frac{1}{6})$ it is projected to the B_b class, and is transferred to GB_b urn. Finally, with probability $\frac{1}{6}$ it goes to urn GC. If a node is initially in any of the GB_a, GB_b, or GC classes, by choosing a frozen node, with probability $2(\frac{1}{N})$, it effectively turns into a $K=1$ function. Thus, it is projected to one lower dimension, which is a side of a square. If there is any homogeneously colored edge, there is a probability that the node freezes; otherwise it goes to an urn of type GXK_1 (where X can be B_a, B_b, or C). Again by looking at the square representation of $K=2$ functions, it is obvious that $\frac{1}{2}$ of the edges of functions belonging to classes B_a, and B_b are homogeneously colored. Therefore, a node with GB_a function, for instance, has a 50% chance of freezing, and with probability $\frac{1}{2}$ it goes to urn GB_aK_1. Finally, a GB_aK_1 functions chooses the frozen node as input with probability $\frac{1}{N}$ and freezes.

We can write the following equations for the average change of the number of nodes in each of the urns described above:

$$\begin{aligned}
\Delta N_G &= -3\frac{N_G}{N} \\
\Delta N_{GB_a} &= \frac{2}{6}\cdot 3\frac{N_G}{N} - 2\frac{N_{GB_a}}{N} \\
\Delta N_{GB_b} &= \frac{3}{6}\cdot 3\frac{N_G}{N} - 2\frac{N_{GB_b}}{N} \\
\Delta N_{GC} &= \frac{1}{6}\cdot 3\frac{N_G}{N} - 2\frac{N_{GC}}{N} \\
\Delta N_{GB_aK_1} &= \frac{N_{GB_a}}{N} - \frac{N_{GB_aK_1}}{N} \\
\Delta N_{GB_bK_1} &= \frac{N_{GB_b}}{N} - \frac{N_{GB_bK_1}}{N} \\
\Delta N_{GCK_1} &= 2\frac{N_{GC}}{N} - \frac{N_{GCK_1}}{N}
\end{aligned}$$

The same argument is applicable to the rest of the function classes, and we can set up the same deterministic set of equations for the average change of the number of nodes in each of the urn. This would lead to the set of differential equations for the number of nodes in each of the urns. We will not write these 65 differential equations here, but will show the final results in a graph. Like before, we can simulate the deterministic equation set and find the number of active nodes of each of the function classes. At each step of the pruning process, a frozen node is picked randomly, its effect on other nodes is studied, and then it is eliminated from the network. This process continues until there is no frozen node left in urn F. In addition, as was explained before, we need to add the appropriate intermediate urns to find out how many of the active nodes are originally from which class. For example, after the pruning process is finished the sum $(N_G + N_{GB_a} + N_{GB_b} + N_{GC} + N_{GB_aK_1} + N_{GB_bK_1} + N_{GCK_1})$ gives the number of active nodes that originally had a function in class G.

Figure 2.8 shows the results of this deterministic process as compared with

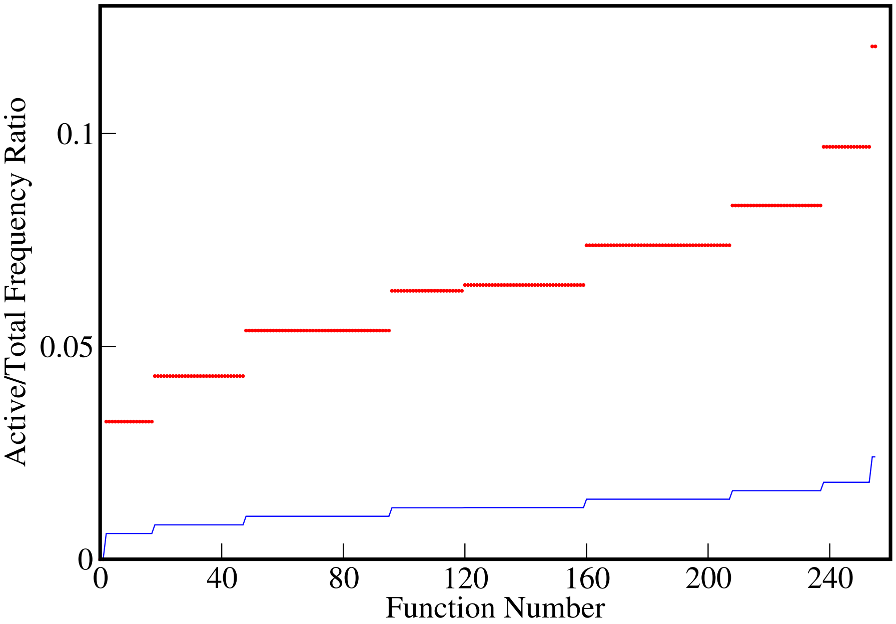


Figure 2.8: Comparison of the results of the direct simulations with mean-field calculations. The results of direct simulations of the active nodes is compared with that of mean-field calculations for an ensemble of 100 000 networks with $N = 100\,000$ nodes. The blue line shows the mean-field results, and the red dots are the results of direct simulations of active nodes. The statistical errors are small compared to the symbol sizes.

ensemble-averaged results of direct simulations of the active nodes. Not surprisingly, since we have not considered the effects of fluctuations yet, the results from the deterministic process do not compare well with the real network simulations. In the next section we will consider the effects of fluctuations in our analysis.

2.3.2 An alternative method of creating critical network ensembles

In the rest of this section, we will discuss an alternative way by which one can make an ensemble of $K = 3$ critical networks, which is not the method we used in this chapter. When a Boolean network is at the critical point, the damage caused by randomly flipping the value of one of its nodes spreads on average to one other node in the network. This can be quantified by sensitivity that is defined for small damage distances by the ratio of damage distance at time $t+1$ to the damage distance at time t. If sensitivity is one, the network is at the critical point. Moreover, sensitivity can be shown to be K times the probability that the output of a node changes when one of its inputs changes. This probability is different for different classes. By definition of P_2, we know that this probability is equal to $1 - P_2$ times the probability of being in each class function. Therefore, we have the following equation for the sensitivity (λ) of $K = 3$ networks:

$$\begin{aligned}\lambda &= 3 \cdot (\frac{1}{4}\frac{N_B^{ini}}{N_{ini}} + \frac{1}{3}\frac{N_{C_a}^{ini}}{N_{ini}} + \frac{1}{3}\frac{N_{C_b}^{ini}}{N_{ini}} + \frac{5}{12}\frac{N_D^{ini}}{N_{ini}} + \frac{1}{2}\frac{N_E^{ini}}{N_{ini}} + \frac{1}{2}\frac{N_{F_a}^{ini}}{N_{ini}} + \frac{1}{2}\frac{N_{F_b}^{ini}}{N_{ini}} \\ &\quad + \frac{1}{2}\frac{N_{F_c}^{ini}}{N_{ini}} + \frac{7}{12}\frac{N_G^{ini}}{N_{ini}} + \frac{2}{3}\frac{N_{H_a}^{ini}}{N_{ini}} + \frac{2}{3}\frac{N_{H_b}^{ini}}{N_{ini}} + \frac{3}{4}\frac{N_I^{ini}}{N_{ini}} + \frac{N_J^{ini}}{N_{ini}})\end{aligned}$$

By setting the sensitivity equal to one, we get the relation between the initial numbers of nodes in different classes that makes the network critical. In other words, if we choose the initial number of nodes in each of the classes in a way that this equation is satisfied, it is guaranteed that network will be in the critical phase.

However, computationally, we used another method to generate critical networks. This is done by biasing the output of the Boolean functions. If we put K equal to three in the criticality condition $2Kp_{\text{crit}}(1 - p_{\text{crit}}) = 1$, we can solve for the bias (p) that makes the network critical. In all the simulations of this paper, we have used the second method; we biased the outputs toward homogeneity with the appropriate value to generate the critical network used in our analysis.

2.3.3 Effects of fluctuations

In order to produce more accurate results we need to consider the effects of fluctuations. Therefore, the goal is to find the probability distribution of the number of active nodes in each of the initial function classes so as to get a better approximation for the average number of active nodes functions with different function types. We first compare the fluctuations of the different urns by solving the deterministic differential equations. The effects of fluctuations will be considered only in the urns with the largest fluctuations, which in this case are urn F, and urns of the form XYK_1. The reason is that the variance of choosing a frozen node is much larger in XYK_1 type functions, and that the number of nodes in urn F gets smaller as we approach the end of the pruning process. Adding appropriate noise terms to the deterministic differential equations we get the Langevin equations.

Following the same argument as in [68], a Gaussian noise term with zero mean and unit variance will be added to the deterministic differential equation of urn F. We get:

$$\frac{dN_F}{dN} = \alpha(\frac{N}{N^{ini}})^2 + \beta^{'}\frac{N}{N^{ini}} + \frac{N_F}{N} + \xi$$

where α and $\beta^{'}$ are constants depending on the initial number of nodes functions of some of the function classes, and are defined as:

$$\begin{aligned} \alpha &= \frac{1}{N^{ini}} \cdot (\frac{1}{2}N_D^{ni} + \frac{1}{2}N_G^{ini} + N_{F_a}^{ini} + N_{F_b}^{ini} + N_{F_c}^{ini} - 2n_J^{ini} - \frac{1}{2}N_I^{ini} - \frac{1}{2}N_B^{ini}) \\ \beta^{'} &= \frac{1}{N^{ini}} \cdot (3N_J^{ini} + \frac{3}{2}N_I^{ini} + \frac{1}{2}N_E^{ini} + \frac{1}{2}N_G^{ini} + N_{H_a}^{ini} + N_{Hb}^{ini}) \end{aligned}$$

Hence, the corresponding Fokker-Planck equation of this stochastic differential equation will be:

$$\frac{\partial P(N_F), N)}{\partial N} = -\frac{\partial}{\partial N_F}\left(\frac{N_F}{N} + \alpha(\frac{N}{N^{ini}})^2 + \beta^{'}\frac{N}{N^{ini}}\right)P(N_F, N) + \frac{1}{2}\frac{\partial^2 P(N_F, N)}{\partial N_F^2}$$

where $P(N_F, N)$ is the probability that N_F nodes are left in urn F when N nodes are left in total.

If we generate critical networks by biasing the outputs of the functions toward homogeneity ($p = 0.768$ for $K = 3$), then we have $\alpha = 0.0015$, and $\beta^{'} = 0.153$. This means that we can neglect the term $\alpha(\frac{N}{N^{ini}})^2$ compared to the term $\beta^{'}(\frac{N}{N^{ini}})$ in the large network size limit, $N_{ini} \rightarrow \infty$. This is because the number of active nodes is always smaller than the initial number of nodes, the term proportional to $(\frac{N}{N^{ini}})^2$ is much smaller than the term proportional to $(\frac{N}{N^{ini}})$, and can be neglected in the large N limit. It even gets smaller as the pruning process gets closer to its end. There is only one situation where we need to consider the $\alpha(\frac{N}{N^{ini}})^2$ term- namely, when $\beta^{'}$ is zero. This corresponds to the case where some of the function classes are initially empty. It has been shown that for a critical network of this type ($\beta^{'} = 0$), the pruning process that is used here does not reveal all the frozen nodes [61]. It neglects the nodes that become frozen by forming self-freezing loops, and not just by receiving frozen inputs. An additional step needs to be added to find the self-freezing loops, and eliminate them from the network. However, here by biasing the output of the functions to generate critical networks we do not consider networks for which $\beta^{'}$ is zero.

Neglecting the term proportional to $(\frac{N}{N^{ini}})^2$ we get the following Fokker-Planck equation:

$$-\frac{\partial P}{\partial N} = \frac{\partial}{\partial N_F}(\frac{N_F}{N} + \beta^{'}\frac{N}{N^{ini}})P + \frac{1}{2}\frac{\partial^2 P}{\partial N_F^2}$$

Using the same change of variables we used before:

$$\begin{aligned} x &= \frac{N_F}{\sqrt{N}} \\ y &= \frac{N}{(\frac{N^{ini}}{\beta^{'}})^{2/3}} \\ f(x,y) &= (\frac{N^{ini}}{\beta^{'}})^{1/3} P(N_F, N) \end{aligned}$$

the $W(N)$ function will be:

$$\begin{aligned} W(N) &= \frac{\partial}{\partial N}\int_0^\infty P(N_F, N)\, dN_F \\ &= (\frac{N^{ini}}{\beta'})^{-1/3}\frac{\partial}{\partial N}\sqrt{N}\int_0^\infty f(x,y)\, dx \\ &= (\frac{N^{ini}}{\beta'})^{-2/3}\frac{\partial}{\partial y}\sqrt{y}\int_0^\infty f(x,y)\, dx \\ &= (\frac{N^{ini}}{\beta'})^{-2/3}G(y) \end{aligned}$$

The final step to find the average number of active nodes of each of the 14 classes, is to find the $G(y)$ function. It can be seen that for these calculations we only need some integrals involving $G(y)$ ($\int yG(y)\, dy$, $\int y^2G(y)\, dy$, and $\int y^3G(y)\, dy$). Therefore, the plot of $(\frac{N^{ini}}{\beta'})^{2/3}W(N)$ versus $\frac{N}{(N^{ini}/\beta')}$ which shows $G(y)$ can be used to numerically calculate the desired integrals. These integrals recalculated numerically using the graph of $G(y)$ as shown in Fig. 2.9.

As an example, the following equation shows the average number of active nodes originated from class G:

$$\begin{aligned} \bar{N}_G &= \int N_GW(N_G)\, dNG = \int N_GW(N)\, dN \\ &= \int (N_G + N_{G_{B_a}} + N_{G_{B_b}} + N_{GC} + N_{GB_aK_1} + N_{GB_bK_1} + N_{GCK_1})W(N)\, dN \\ &= -\frac{1}{4}\frac{N_G^{ini}}{N^{ini}}\frac{1}{(\beta')^2}\int y^3G(y)\, dy - \frac{1}{2}\frac{N_G^{ini}}{(N^{ini})^2}(\frac{N^{ini}}{\beta'})^{4/3}\int y^2G(y)\, dy \\ &+ \frac{7}{4}\frac{N_G^{ini}}{N^{ini}}(\frac{N^{ini}}{\beta'})^{2/3}\int yG(y)\, dy \end{aligned}$$

where the appropriate intermediate urns are added to find the total number of active nodes, which were originally in class G. Note that we have also used of the fact that $W(N)\, dN = W(N_G)\, dN_G$. Figure 2.10 compares the results of the number of active nodes of different function classes derived by direct simulations, pruning equations,

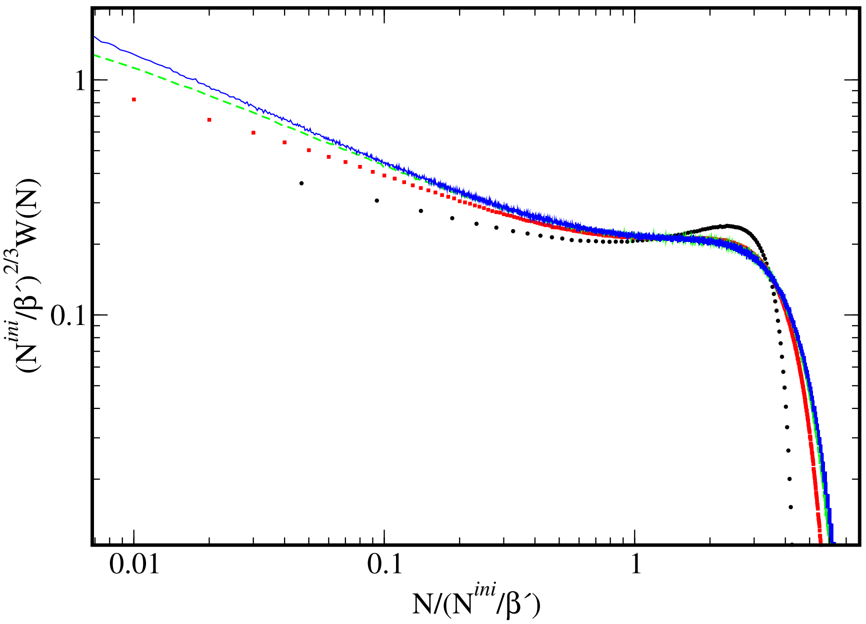


Figure 2.9: The plot of $G(y)$ function. The function G(y) is plotted for different network sizes. The black dots show the results for networks of size 100. The red squares are for networks of size 1000, the green dashed line show the results of networks of size 10 000 and the blue line is for networks of size 100 000.

and using the analytical results. It shows that:

i. The relative frequency of the relevant nodes of different classes is the same as that of the active nodes of the critical random Boolean networks. Therefore, we can focus on active nodes rather than relevant nodes to understand the canalization behavior of different function types of the network.

ii. The results derived by simulating the stochastic process, i.e. ,Langevin equations, are within the error bars of the real network results. This means that these two data sets are statistically similar, which confirms our method.

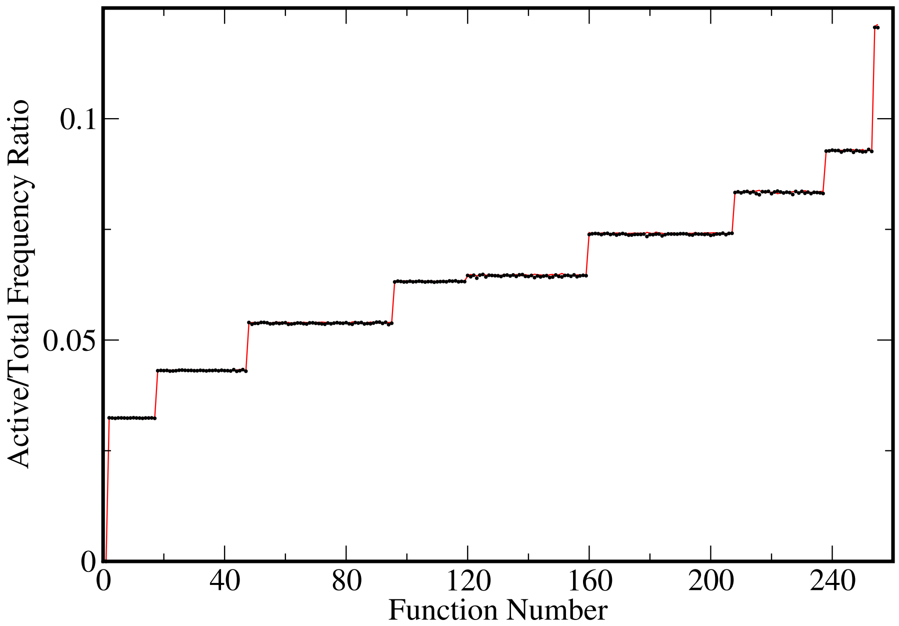


Figure 2.10: Comparison of the results of the direct simulations with analytical results. The results of direct simulations of the active nodes is compared with that of analytical calculations for an ensemble of 100 000 networks with $N = 100\,000$ nodes. The red line shows the analytical results, and the black dots are the results of direct simulations of active nodes. The statistical errors are small compared to the symbol sizes.

iii. The frequency ratio of the active nodes functions to the initial number of functions clearly shows that more canalizing functions appear less frequently in the active nodes of critical random Boolean networks. In fact, the only characteristic of the functions that determine their function class is their canalization values.

2.4 Summary

In this chapter we investigated the structure of the critical dynamics of random Boolean networks. We analytically derived the frequency of the different types of

Boolean functions among the active nodes of critical random Boolean networks since these nodes are relevant to the network dynamics. By setting up a set of differential equations that determines the active nodes through a pruning process, we can find the average number of active nodes functions. Then, considering the effects of fluctuations, the probability distribution of the number of active nodes is accurately calculated. We found that in critical networks the frequency ratio of active nodes functions by the total number of functions is inversely correlated with their canalization values, and there exist classes, or sets of Boolean functions that cause the nodes whose dynamics they control to contribute equivalently to the critical dynamics when averaged over an RBN ensemble. The existence of these function classes can be discovered both analytically and computationally. From analytical point of view, by writing the differential equations set up for different function types during the pruning process, it is obvious that function types that have the same canalization values have the same differential equations with the exact same coefficients.

The existence of function classes is a manifestation of the symmetry of critical random Boolean networks dynamics. This shows that not all different function types contribute differently to the random Boolean networks dynamics. The functions that are in the same class are said to be symmetric with respect to each other, and their common feature is their canalization values. In the next chapter we explore the nature of this symmetry in critical random Boolean networks dynamics, and try to describe it in group theory language.

It should be noted that we have done the calculations and simulations for networks with $K=2$, and $K=3$ critical random Boolean networks. But the methods used can be easily generalized to any in-degree K. In general, we can conclude that in critical random Boolean networks with any in-degree K there exist classes of functions on the active nodes of the network, and their dynamical behaviors are identical.

In principle, the methods developed in this chapter are useful more broadly for heterogeneous complex systems. In particular, in chapter 4 we show that they can be used for a broader class of heterogenous complex systems than Boolean networks, namely discrete multi-state networks.

Chapter 3

Symmetry of critical random Boolean networks dynamics

3.1 Introduction

Complex systems often differ from more traditional condensed matter systems because they consist of heterogeneous components [22], which makes their analysis and achieving any sort of general understanding of their behavior difficult. A question that naturally arises then is: Can symmetry in heterogeneous complex systems be exploited to simplify their analysis and obtain fundamental insights into their dynamics? In order to answer this question, it is essential to determine the role that symmetry has in controlling the behavior of these systems, and what it can tell us about their structure. It is also important to know whether symmetry can be used to distinguish systems with different dynamics.

In this chapter, we explore the role of symmetry in the dynamics of critical Boolean network models of complex systems. In particular, we consider symmetry in the behavior of the active nodes that we studied in chapter 2. This approach contrasts with studies of complex networks that have considered symmetry in the topological structure of the links, such as Ref. [65]. By analogy with other condensed matter systems, it is expected that the importance of symmetry will be most evident in critical networks, at the continuous transition, “at the edge of chaos” [59].

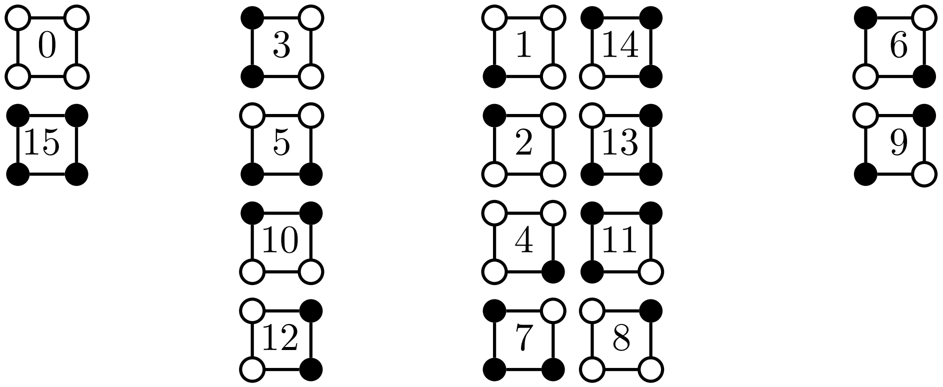
For this, we look more closely at the frequency distribution of the heterogeneous Boolean functions of the nodes that remain active on an attractor in critical networks studied in chapter 2. It was shown in chapter 2 that there are classes of functions, related by symmetry, that occur with the same frequency ratio. Here, we will determine the nature of this symmetry and show that it is related to preserving a form of network robustness known as *canalization*, which is of importance in developmental biology [101]. Canalization preserving symmetry is also compared and contrasted with that of the dynamics of an evolutionary game played by the nodes, that selects for diversity in their behavior [78] in order to show how different dynamics at the critical state correspond to different symmetries. Finally, we discuss how our results demonstrate the usefulness and power of using symmetry for characterizing complex network dynamics, and point to potential new approaches to the analysis of complex systems.

Please note that many of the results of this chapter are published in Ref. [56].

3.2 Nature of symmetry of the critical dynamics

Both the simulation and the analytic results in chapter 2 have shown that there are classes of functions that produce equivalent, symmetric behavior. Indeed, as we have seen in Eqns. 2.1, the equations for the urns that contain nodes with functions that are in the same class are identical, making the symmetry of the output functions explicit. Accounting for this symmetry allows for the calculations described in the previous section to become dramatically simplified. That is, it is not necessary to have separate urns for nodes with each different function or even function pairings. Instead, merely considering a collection of urns containing all nodes that have functions in the same class yields identical results. So what is the nature of the symmetry between functions in critical random Boolean networks that produces this identical behavior of functions in a class?

From the description of the pruning process in critical random Boolean networks in section 2.2.1 one can see that functions are in the same class, if they have the same probability of becoming frozen at each stage of the pruning process. In general, this is true when fixing up through $K-1$ inputs for functions with K inputs. That is, if for Boolean functions with K inputs the quantities P_k are defined for $k=0,1,\ldots,K-1$ as the probability that the output of the function is determined if k randomly chosen inputs are fixed, but the other inputs are allowed to vary, then functions with the same set of P_k values are in the same class. The function classes of the critical dynamics are uniquely determined by their set of P_k values. The number of such classes is 2, 3, 10, and 46 for functions of $K=1$, 2, 3, and 4 inputs, respectively.



P_0	1	0	0	0
P_1	1	$\frac{1}{2}$	$\frac{1}{2}$	0
h	1	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{1}{2}$

Figure 3.1: Ising hypercube representation of the 16 Boolean functions with in-degree $K = 2$. The decimal numbering of each function is given inside its corresponding Ising square representation. The canalization values, P_ks, and the internal homogeneity, h, of the functions are also shown. In a hypercube representation, P_k is the fraction of $K - k$ dimensional hypersurfaces of the hypercube that are homogeneously colored. In this case, since $K = 2$, P_0 is 1 if the function is a frozen function, represented by a homogeneously colored square, and is 0 otherwise. P_1 is the fraction of the homogeneously colored edges in the representation of the function. Functions with the same set of P_ks and h are grouped together.

The ability of a Boolean function to have its output become fully determined, so that it becomes frozen, when only a subset of the input values are specified, is a property known as canalization, introduced in chapter 2. Subsets of inputs that control the output of a Boolean function in this way are called canalizing inputs. Long recognized as a property of developmental biological systems [101], canalization is thought to be important because it allows greater evolutionary variation, hidden behind canalized traits, without potentially deleterious effects [102]. How canalizing a Boolean function is can be quantified by its set of P_k values. Figure 3.1 shows the canalization values of different functions with $K = 2$ inputs.

The symmetry of the critical state of random Boolean networks, then, is that which preserves canalization. Having the same set of P_k values defines an equivalence relation between Boolean functions with the same number of inputs, and a set of functions with the same set of P_k values form an equivalence class. The group that describes the canalization preserving symmetry has these equivalence classes. These equivalence classes are also orbits of that symmetry group; the functions in a class map into each other through symmetry transformations that are elements of the group.

For Boolean functions with $K = 2$ inputs, the smallest symmetry group we have identified that has the correct function classes has 48 symmetry operations. This group can be generated by three symmetry operations that we designate as R, N and P. These are shown in Fig. 3.2. R is counterclockwise rotation by 90 degrees in the geometrical representation of the functions, P is the parity operation that maps output 0 into output 1 and vice versa, and N is a *pruning* operation that maps functions onto one another that have different internal homogeneity but the the same canalization properties. The presentation of the group generated by the relations between these symmetry operations is

$$\{R^4 = N^2 = P^2 = I, (RN)^3 = I, RP = PR, NP = PN\},$$

where I is the identity operation. The structure of this group can be seen in the Cayley diagram shown in Fig. 3.3. In this diagram, nodes represent the elements of

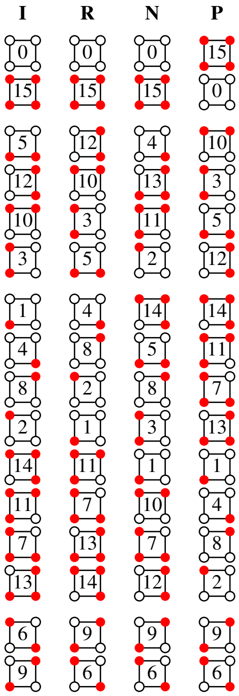


Figure 3.2: Generators of the minimal canalization preserving symmetry group for Boolean functions with $K=2$ inputs. The identity operation and the three operations that generate the group are indicated by their mapping of the set of 16 Boolean functions with $K = 2$ inputs onto itself. The first column depicts the unchanged Boolean functions, or equivalently the identity operation I. The other columns show the effect of the symmetry operations rotation R, pruning N, and parity P on each of the functions, respectively. Functions are listed in the first column consistent with the groupings in Fig. 1.1.

the group, and directed edges of the graph connecting them represent the effects of the generators of the group. That is, an edge, E, from element e_1 to element e_2 shows that $E \cdot e_1 = e_2$. In this case, edges with different colors correspond to the effects of the different generators of the group. If only rotations are considered, which are indicated by the red arrows, the diagram consists of 12 disconnected squares. Combining parity, shown by the black arrows, with rotation pairs the 12 squares into

6 disconnected cubes. Finally, the cubes become interconnected with the pruning operation, shown by blue arrows.

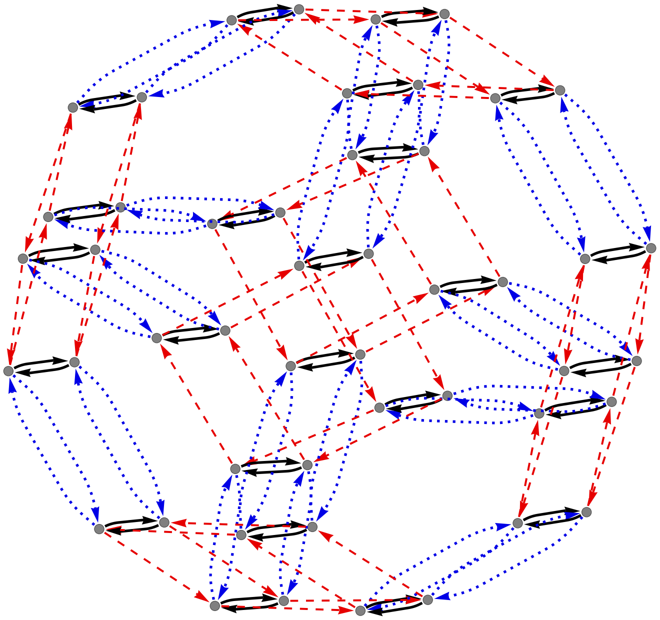


Figure 3.3: Cayley diagram of the minimal canalization preserving symmetry group for Boolean functions with $K = 2$ inputs. The elements of the group are the nodes of the graph, shown as gray dots. Directed edges of the graph indicate the effect of combining one of the group generators with an element. Arrows with different colors correspond to the effect of the 3 different generators of the group; rotation R, pruning N, and parity P operations are shown as red dashed, blue dotted, and black solid arrows, respectively.

Function classes, or orbits, do not, however, uniquely determine a symmetry group. Indeed, one can find more than one symmetry group for each value of in-degree K that preserves canalization. This is because the critical state may have additional symmetries, which will be reflected in the complete symmetry group of the dynamics. The actual symmetry of the critical dynamics of networks with $K = 2$ may, thus, be larger than the minimal group depicted in Fig. 3.2. Ideally, one would like to find the smallest group that describes all symmetries of the critical dynamics of networks for each value of K. This remains a difficult, open question. Nevertheless, the orbits manifested by a dynamics can be used to distinguish it from other dynamics.

3.3 Symmetry in other dynamics

The fact that the various Boolean functions behave symmetrically and can be grouped together into classes that characterize the symmetry is not restricted to critical random Boolean networks dynamics. The symmetries of other types of dynamics of Boolean networks can be manifested analogously, and a similar analysis to that described above can be done to determine the symmetry in those cases. Consider, for example, the adaptive dynamics mentioned in the introduction, in which the nodes compete in an evolutionary game that causes the network to self-organize to a critical state [78]. This game evolves the output functions used by the nodes through an extremal process [6, 80] that uses negative reinforcement [29] to select against majority behavior [110]. One step of the process, which occurs on a long time scale referred to as an epoch, starts with some random initial state of the nodes, and finds the dynamical attractor. It then targets a node for evolution by finding the one whose output is most often aligned with the output of the majority of nodes over the period of the attractor, and replaces its output function with a randomly chosen Boolean function. The process continues indefinitely, evolving the output function of one node each epoch. Over time, the set of output functions used by the nodes evolve through the process to promote diversity among their dynamics [10].

Through the evolutionary dynamics of the game it has been shown that a network will evolve to a critical state, evidenced by a power-law distribution of attractor periods [78]. The evolved state is a steady state, in the sense that ensemble-averaged properties are time-independent. For networks of nodes that each have K inputs, depending on the value of K, this occurs over a wide range of bias p with which the new Boolean functions are randomly chosen to replace those of the targeted nodes that lose the game [85]. Networks with $K=3$ inputs per node will self-organize to a critical steady state even when the new Boolean functions are unbiasedly chosen, that is, chosen with $p=1/2$. Note that, as described earlier, the game's evolutionary process will alter only the nodes' output functions; it does not explicitly alter the directed edges describing the regulatory interactions of the nodes. Nevertheless, the process does effectively alter the edges, making the process effectively an adaptive one that co-evolves both node and edge dynamics [47]. This is because some inputs of the output functions can be rendered effectively irrelevant through canalization. This makes their corresponding input edges effectively irrelevant, and thus, as the output functions evolve, so will the directed graph that describes the effective regulatory interactions [84].

An analysis along similar lines to that described above for finding symmetry in the critical state dynamics can also be done to find symmetry in the adaptive dynamics of the evolutionary game. In order to do so, however, the distribution of output functions of all of the nodes should be studied. This contrasts with our analysis of critical state dynamics of random Boolean networks, where the distribution of output functions of only a subset of the nodes, active nodes, was studied. This difference is due to the fact that the adaptive dynamics of the game directly affects the distribution of all the output functions, whereas critical dynamics involves only a subset of nodes. Indeed, if an analysis of the distribution of the output functions of nodes that are active on attractors, similar to what we have done for random Boolean networks constructed to be critical, is done on the networks evolved to a critical state by the game, then canalization preserving symmetry, not Zyklenzeiger symmetry, will be found.

For large networks of $K = 3$ inputs, it has previously been shown that the ensemble-averaged distribution of output functions used by all of the nodes in the evolved steady state of the game has 14 different classes, not the 10 found for critical state dynamics [84]. Thus, the symmetry of the adaptive dynamics of the evolutionary game differs from that of the critical state dynamics. In the case of the game, the symmetry corresponds to that of the K-dimensional Zyklenzeiger group, which consists of octahedral symmetry operations combined with the parity operation [84]. The 3-dimensional octahedral symmetry operations are 24 proper and 24 improper rotations of the Ising hypercube [51], which together with parity operator, P, make the Zyklenzeiger group a group of order 96 [52]. The orbits of this group correspond to the 14 classes of Boolean functions. The functions that are in the same class are mapped into each other by the elements of the Zyklenzeiger group. The hypercube representation of one representative function from each class is shown in Fig. 2.6 of chapter 2. All of the rest of the 256 functions can be obtained from these representative functions through the Zyklenzeiger operations. These 14 Zyklenzeiger classes are labeled from A to J, according to the canalization properties of the functions in the class. Additionally, those functions labeled C, F, and H, form multiple classes, designated with subscripts, to distinguish those with the same canalization properties but different internal homogeneity and parity symmetry [10]. The 10 function classes of critical network dynamics result from combining the Zyklenzeiger classes designated with subscripts. Thus, some of the canalization preserving classes split to make the Zyklenzeiger classes. This suggests that the actual canalization preserving symmetry is a higher symmetry than the Zyklenzeiger group, and therefore that the critical dynamics of Boolean networks have a higher symmetry than does the adaptive evolutionary game.

Still there are other dynamical symmetries for Boolean networks. In the case of large networks of $K = 2$ inputs, it has been shown that the 2-dimensional Zyklenzeiger group has 4 orbits that correspond to the 4 classes of output functions of the evolved steady state of the game [84]. This behavior is different from that of the critical state dynamics of random Boolean networks with 2 inputs that we saw earlier, which has 3 orbits, reflecting the distinct dynamics of these systems. Also, if the evolutionary game is played on finite size networks, the Zyklenzeiger symmetry is broken, and the evolutionary dynamics is controlled by the *input-inversion* subgroup of the Zyklenzeiger group [64]. This is because finite size effects force the nodes to co-operate differently, and hence change the underlying dynamics. This subgroup has 46 classes for functions with $K = 3$ inputs. Another example of a different dynamics on Boolean networks is threshold Boolean networks, for which the output values of nodes of the network change depending on the comparison of the sum of their input values and a threshold value. These networks can be used to model genetic regulatory and neural networks [86, 109]. Since threshold networks have a different dynamics from the evolutionary game and the random Boolean networks, one can expect to find a different symmetry that controls their dynamics, but the number of orbits that results from this dynamics is not yet known.

Here we see that different dynamics are controlled by distinct symmetries of the critical state, and as a result distinct symmetry groups. The characteristic features of the functions that divide them into classes give insight into the behavior and structure of the networks. For instance, the fact that canalization properties are the characteristic features that classify the functions of critical random Boolean networks tells us how the active nodes form a substructure of the network. Moreover, if one knows the symmetry of the critical state *a priori*, it makes the analytical description of the system much easier, because in such a description each class of functions can be reduced to a single entity. This indeed shows that using symmetry can be a powerful method for analysis of the behavior of critical Boolean networks.

3.4 Summary

In this chapter, we have shown that symmetry can be used to obtain fundamental insights into the dynamics of heterogeneous complex systems. These systems consist of components that can behave heterogeneously in the system's dynamics. When the dynamics has a symmetry, that symmetry can be manifested in the way that the heterogeneous components behave, causing classes of different components to behave in a similar, symmetric manner. By comparing the behavior of the various components, characteristic features of the symmetry can be discerned. Different types of dynamics can be manifested differently, allowing them to be distinguished. Furthermore, the symmetric behavior of classes of the components can be exploited to reduce the complexity of analytic models of their behavior.

To concretely demonstrate how symmetry is manifested in and can be used to help understand the dynamics of heterogeneous complex systems, we have used critical random Boolean networks as prototypical examples. Random Boolean networks are ideally suited for this purpose because they are relatively simple models that have heterogeneous structure and their constituent parts have their own dynamics. This heterogeneity occurs through the differing output functions of the nodes and through the directed edges defining the regulatory node interactions. Importantly, there is also a continuous transition in the nature of the dynamics of random Boolean networks. Like other condensed matter systems, their critical states at the continuous transition are scale invariant, making symmetry particularly important and apparent. For this reason, we have chosen to focus on the critical state properties of random Boolean networks in order to be as clear as possible.

We have shown that the symmetry underlying the critical dynamics of random Boolean networks is revealed by the ensemble-averaged likelihood that nodes with a given output function remain active on the dynamical attractors. Using numerical simulations, we found that the ratio of the number of nodes utilizing a particular Boolean function that are active on dynamical attractors to the number of all nodes utilizing that function depends solely on the function's canalization properties. Canalization describes the ability of a function to have its output fully determined by a subset of its inputs, thus rendering the remaining inputs irrelevant. That is, all functions with the same canalization properties have the same likelihood of being active. Thus, there is a symmetry between all Boolean functions with the same canalization properties. Using an analytic model of the critical dynamics involving a stochastic pruning process that is able to accurately reproduce our numerical results, we showed that the symmetry between nodes that have output functions with the same canalization properties becomes explicit.

Symmetry of Boolean networks was contrasted with the symmetry underlying the adaptive networks dynamics of a game that governs the evolution of the output functions in a way that causes the network to self-organize to a critical state. It was previously shown that classes of functions exist for this evolved state. For large networks, these classes are orbits of the Zyklenzeiger symmetry group, which is distinct from the canalization preserving symmetry group. For finite-size networks, a breaking of the Zyklenzeiger symmetry is known to occur, which causes the function classes to split. Other symmetries that may underlie other types of dynamics of Boolean networks, such as threshold adaptive dynamics, [63] could manifest themselves through similar methods of analysis that compare the frequency with which the various heterogeneous output functions are utilized by the nodes involved in the dynamics. Symmetries of Boolean network dynamics that appear in this way will always reflect point group symmetries, since Boolean functions are discrete and finite. In more general types of heterogeneous complex systems, however, symmetries among the heterogeneous components can also be continuous. That is, if instead of heterogenous Boolean functions, heterogeneous continuous functions are assigned to the nodes of the network, the underlying symmetry may be reflected by continuous symmetry groups. It would be interesting to extend the ideas and tools developed in this chapter to the more general cases of systems with continuous symmetries.

Finally, we note that in the analysis of the critical dynamics of Boolean networks presented here, we appealed to a prior understanding of the dynamical process in order to explicitly see its symmetry. Such prior understanding, however, is not required. In fact, our approach to understanding heterogeneous complex systems through analyzing the symmetry of the behavior of different components may find its greatest benefit when used in empirical analyses of the behavior of experimental complex systems for which there is no understanding at the outset. In these cases, as we discussed, a direct empirical study of the behavior of the heterogeneous network components on the various dynamical attractors can be made.

In the next chapter we consider a generalization of Boolean networks to investigate wether they have the same symmetry of critical dynamics. If the same canalization preserving symmetry governs their dynamics, this would suggests that all of these systems and models are in the same universality class of heterogenous complex systems.

Chapter 4

Symmetry of critical discrete multi-state network dynamics

4.1 Introduction

In this chapter, in order to show that the methods we developed to characterize heterogeneous complex systems are general and can be applied to models beyond Boolean networks, we study the critical dynamics of a class of models that are a generalization of Boolean networks, multi-state networks [107]. First, we investigate the criticality condition of multi-state networks with a new method to describe the phase space of multi-state networks. Then, we find the symmetry of dynamics of the critical state based on the distribution of the functions assigned to the active nodes of the network. Based on the same pruning process used for Boolean networks in chapter 2 and 3 that searches for the active nodes of the network, we can also predict the distribution of different function types on the active nodes of the network. We show that the same canalization preserving symmetry, as in Boolean networks, controls the critical discrete multi-state networks dynamics.

From a statistical mechanics point of view, random Boolean networks can be seen as variation of the *Ising model* where the topology is no longer a lattice, but interactions are now directed, random, and non-local. The number of interaction partners, K, is still fixed. The generalization of Boolean networks to multi-state networks is analogous to the generalization of Ising model to *Potts model* on lattices. The big difference is that in both the Ising and the Potts model one can readily write down and solve the Hamiltonian, while for multi-state networks and Boolean networks since the interaction between units are directed and typically asymmetric a Hamiltonian (energy function) does not exist [49]. The dynamical properties of multi-state networks can be approached in terms of damage spreading [92]. It is known that, like Boolean networks, they show a phase transition at some critical point which depends on the ensemble of functions assigned to nodes of the network and on K [107]. There are two dynamical phases: chaotic and frozen. A perturbation of one node's state propagates on average to more than one node at one time step in the chaotic phase, while it spreads on average to less than one node in the frozen phase. At the critical point the damage spreads on average to a single node per time step.

For simplicity, we start with *ternary networks* where the possible number of output states per node is three, $S = 3$, and then generalize our results to to multi-state networks with larger S.

4.2 Criticality condition

Multi-state networks are a class of discrete dynamical systems. The outputs of the nodes of the networks take values in discrete, finite sets and develop in discrete time steps. At each time step, the value of a variable is determined by an update rule that deterministically depends upon the values of some of the other variables at the

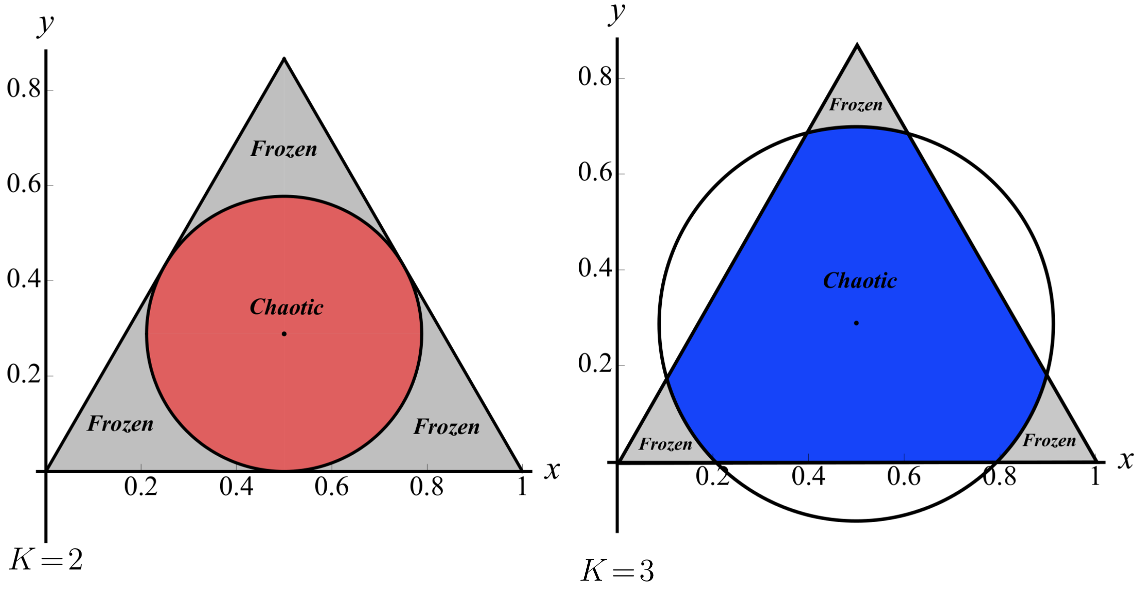


Figure 4.1: Ternary plots showing the phase diagrams of MSNs with $S=3$ (ternary networks) with $K = 2$ and $K = 3$ in-links per node. All the points in the grey area inside the equilateral triangles are all possible biases that are the solutions of the equation $p + q + r = 1$.The circles are the criticality circles. All points on the circle and inside the triangle are critical points of the network. While, (p, q, r)-tuples inside a criticality circle lead to chaotic dynamics, those in the remaining cusps of the triangle lead to frozen dynamics. For the case of $K = 2$ ternary networks, the contact point of the criticality circle and the ternary plot is the critical point of the corresponding Boolean network. For instance, the point (x,y)=(0.5,0) in the ternary graph is $(p, q, r) = (0.5, 0, 0.5)$ in the pqr space according to Eq. 4.4. This is the critical point of a Boolean network with $K = 2$ in-links per node. The two possible output values of this Boolean function can be either $\{0, 1\}$ or $\{0, 2\}$, or $\{1, 2\}$, depending on what biases p and r are toward.

previous time step, much like the Boolean networks dynamics.

Ternary networks are discrete multi-state networks where each node can have three possible states, for instance $\sigma_i \in \{0, 1, 2\}$. Each node receives K in-links randomly chosen from the N nodes of the network. Self-links and multiple edges may be forbidden or not. We do not consider self-links and multiple edges, but it does not change the results reported here. Also, a ternary function, a function that has three possible output states, is assigned to each node of the network. If the functions are chosen randomly form the set of all possible ternary functions as well, then the network is called random ternary network. Overall there are 3^{3^K} function types. In general, for a multi-state function with S states, there are S^{S^K} possible function types. The functions determine the value of their assigned nodes at each time step. The variables of a node's function are the previous-time-step values of K nodes which give in-link to that node. The functions of the network not only determine the dynamics of the nodes, but their distribution also determines the dynamical phase of the network.

Functions can be defined by their output values. In order to define a ternary function, similar to a Boolean function, the output values of all the possible input sets, 3^K for ternary functions, must be fixed. Table 4.1 shows a ternary function with 2 inputs, $K = 2$. When constructing the functions, one can bias the outputs towards different values. For instance, a ternary function can have bias p towards 0, bias q towards 1, and bias r towards 2. This means that the probability that a given function delivers 0 is p, q is the probability that 1 is the function's outcome, and r is the probability that 2 is the outcome. Since 0, 1, and 2 are all the possible outputs, it is obvious that $p+q+r=1$.

Input	00	01	10	11	02	20	22	12	21
Output	0	1	2	0	0	2	2	1	1

Table 4.1: Example of a ternary function, $S=3$, and with 2 inputs, $K=2$.

The criticality condition for multi-state networks relates the average number of in-links, $\overline{K}$, with the average probability for the functions to yield two different values for different arguments, $\overline{P}$, [107]:

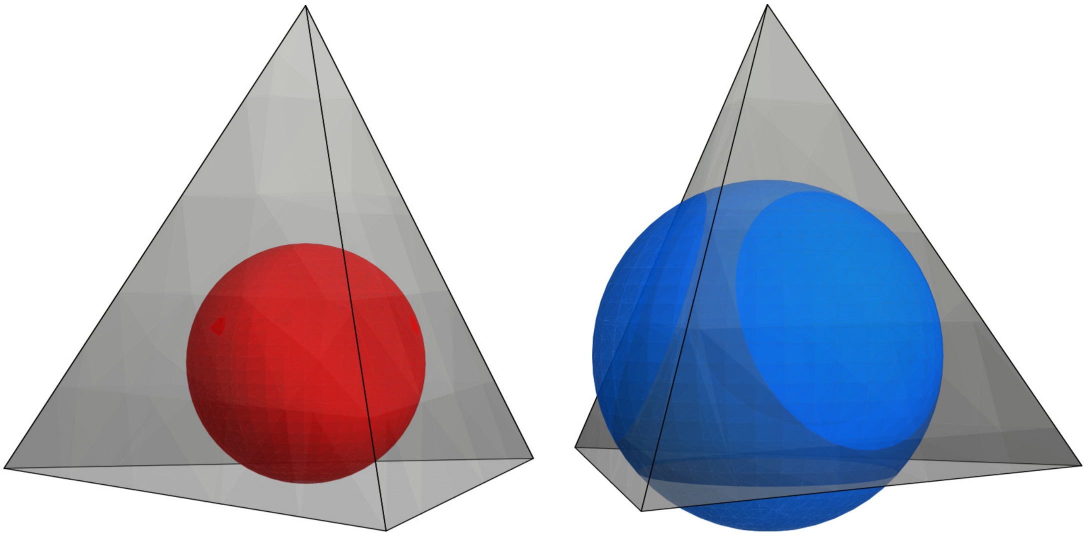
$$\overline{K}_{\text{crit}} \;=\; \frac{1}{\overline{P}} \tag{4.1}$$

If $S=2$ this equation simplifies to the well-known criticality condition of Boolean networks Eq. 1.4 discussed in chapter 1, which states that if the functions are chosen with a magnetization bias p, the criticality equation guarantees that the ensemble of networks is critical.

For $S=3$ the criticality equation becomes

$$2\overline{K}\left[pq+(p+q)\big(1-(p+q)\big)\right] \;=\; 1\,, \tag{4.2}$$

where p and q are the biases of the ternary functions, and r is replaced with $1-(p+q)$. Here, $\overline{P}$ in Eq. 4.1 is replaced with $pq+(p+q)\,(1-(p+q))$. This is because the average probability for the function to yield two different values for different arguments is equal to the probability that the output of the function is σ times the probability that by changing the argument of the function the output is something other than σ. In order to get the average probability we should sum over all $\sigma \in \;\{0,1,2\}$.



$K = 2$ $K = 3$

Figure 4.2: Phase diagrams of MSNs with $S = 4$ with $K = 2$ and $K = 3$ in-links per node. All the points in the grey area inside the tetrahedron are all the possible biases solving equation $p + q + r + t = 1$.The spheres are the criticality spheres. All points on the sphere surface and inside the tetrahedron are critical points of the network. While (p, q, r, t)-tuples inside a criticality sphere lead to chaotic dynamics, those in the remaining cusps of the triangle lead to frozen dynamics. The intersections of the criticality sphere and the tetrahedron are the criticality circles of the corresponding ternary network, one on each side.

A convenient way to visualize the criticality condition of ternary networks is to use *ternary plots*. A ternary plot or ternary diagram is a barycentric plot which displays the proportion of three variables that sum to a constant. These plots are widely used in physical chemistry, metallurgy, and other physical sciences to show the compositions of systems composed of three species [106]. In the case of our ternary functions the sum of the three biases is 1, $p + q + r = 1$. Therefore, all the points inside the triangle in Fig. 4.1 are the solutions of this equation. In other words, all the possible (p, q, r)-tuples lie within this triangle

The ternary plots are shown in terms of the coordinate axes x, and y. The (x, y)

pair can be converted to the (p, q, r) triple as follows:

$$\begin{cases} p &= \quad x - \frac{y}{\sqrt{3}} \\ q &= \quad \frac{2}{\sqrt{3}} \cdot y \\ r &= 1 - (p + q) \end{cases} \tag{4.3}$$

However, not all points of the ternary plot make a critical ensemble of networks. Converting Eq. 4.2 to the xy-coordinates we get

$$\left(x - \frac{1}{2}\right)^2 + \left(y - \frac{1}{2\sqrt{3}}\right)^2 \quad = \quad \frac{1}{3} - \frac{1}{2\overline{K}}\,. \tag{4.4}$$

This is the equation of a circle in the ternary plots coordinates, where the radius of the circle depends on $\overline{K}$, the average number of in-links of each node. We call it *criticality circle* because all the points that are on the circle and inside the triangle correspond to critical biases. The points of the intersection of the triangle with the criticality circle correspond to the case where one of the biases is 0. In fact, all of the points on the edges of the triangle are the ones where one of the biases is 0. If one of the biases is 0 it means that the function has only two possible output values or in other words it is a Boolean function! It can be easily shown that these critical points are exactly the ones that are predicted by the criticality equation of Boolean networks, Eq. 1.4. Figure 4.1 shows two 2D ternary plots, each of them corresponding to a different value of $\overline{K}$, and hence different radius.

Ternary diagrams can be easily generalized to higher dimensions. In general, critical biases are given by the points on a $(S-1)$-sphere inscribed into a $(S-1)$-simplex, where the radius of the sphere depends on K. If $S=4$, then the $(S-1)$-simplex plot will be three dimensional. We will have a regular tetrahedron instead of the equilateral triangle, and the criticality circle will become the *criticality sphere.* As before, the points that are on the surface of inscribed sphere correspond to the critical biases. This time there are four biases, since we have four possible states for each node. Figure 4.2 shows two 3-simplex plots, each of them corresponding to a different value of $\overline{K}$. The circles formed by the intersection of the sphere and each of the triangular sides, correspond to the criticality circle of the ternary networks, where $S=3$.

4.3 Symmetry in function frequencies

Here, we want to use the ideas and methods we previously applied to random Boolean networks in chapter 2 and 3. We want to find a manifestation of the symmetry of the dynamics of the critical multi-state networks, and compare it with that of the random boolean networks.

We start with ternary networks with two in-links per node, where $S = 3$ and $K=2$. Again, since active nodes determine the network dynamics, we look at their behaviors. For the nodes which are active even after a transient time, we count which ternary function occurs how often. In other words, we find the function frequency on the active nodes of the network. To make sure that the overall number of each function type is not correlated with number of that function on the active and frozen nodes, we divide it by the overall number of that function type. Figure 4.3 shows the ratio of the functions used by active nodes divided by the ones used by the total nodes of the network. We know from the Boolean case that the update functions are generally utilized with different frequencies and nodes with different functions can play different roles in the dynamics of a network [56].

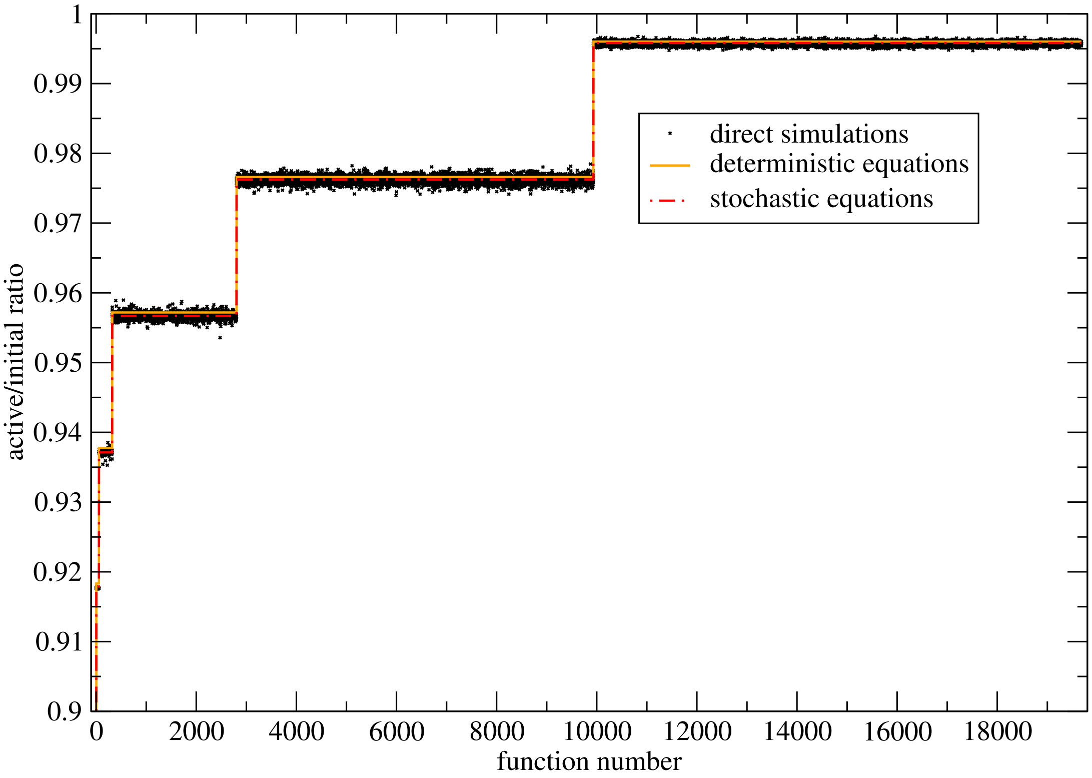


Figure 4.3: Function ratios for ternary networks with $N = 1000$ nodes, $K = 2$ in-links per node and biases $p = q = 1/6$, averaged over an ensemble of 10^7. The (black) crosses are the results of direct simulation an ensemble of 10^7 network realizations, the (solid) orange line is solution of the deterministic equations, the (dash-dot) red line is the numerical solutions of the stochastic equations including a noise term.

In Boolean networks, a *pruning process* of the nodes of a network allows to numerically determine the average number of active nodes with each function type in a stepwise procedure [61, 68]. The pruning process is discussed in details in chapter 2 and 3. Here we use the same process to determine the active nodes of the networks.

As before, to calculate the average number of each function type during the pruning process we need to set up a system of urns. If we use the urn system blindly, we should set up an urn for each type of function, and for ternary networks with even 2 inputs per node it will be at least $3^{3^K} = 3^{3^2} = 19683$ urns. Instead, we exploit the symmetry that we see in Fig. 4.3. There, we can see that functions group into classes that occur with the same frequency. The functions that are in the same class are dynamically equivalent. Therefore, we consider each class of functions as one entity, and write the change in the average number of functions in these classes. It should be emphasized that one would get the same results if instead they looked at each of the 19683 functions separately.

For the ternary networks with two in-links per node there will be 11 urns. There are 5 *original* urns and 5 *intermediate* ones. The 5 original urns correspond to the five classes of functions seen in Fig. 4.3. When a node receives a frozen input it might become frozen or not. If it becomes frozen it is moved to the frozen urn, but if does not freeze it will be moved to an intermediate urn. The nodes that are in the intermediate urns have already received a frozen input in the previous steps, and have functions that have effectively less open inputs. In the case of $K=2$, there is only one intermediate step, therefore, one intermediate urn for each original urn. If K was three there were two intermediate steps, and so on.

We can write down the following set of equations for the change in the average number of nodes in each urn:

$$\begin{aligned}
\Delta N_2 &= -2\frac{N_2}{N} & \Delta N_{2K_1} &= \frac{2}{6}\cdot\frac{2N_2}{N} - \frac{N_{2K_1}}{N} \\
\Delta N_3 &= -2\frac{N_3}{N} & \Delta N_{3K_1} &= \frac{3}{6}\cdot\frac{2N_3}{N} - \frac{N_{3K_1}}{N} \\
\Delta N_4 &= -2\frac{N_4}{N} & \Delta N_{4K_1} &= \frac{4}{6}\cdot\frac{2N_4}{N} - \frac{N_{4K_1}}{N} \\
\Delta N_5 &= -2\frac{N_5}{N} & \Delta N_{5K_1} &= \frac{5}{6}\cdot\frac{2N_5}{N} - \frac{N_{5K_1}}{N} \\
\Delta N_6 &= -2\frac{N_6}{N} & \Delta N_{6K_1} &= \frac{2N_6}{N} - \frac{N_{6K_1}}{N}
\end{aligned} \tag{4.5}$$

$$\begin{aligned}
\Delta N_F = &\frac{4}{6}\cdot\frac{2N_2}{N} + \frac{N_{2K_1}}{N} + \frac{3}{6}\cdot\frac{2N_3}{N} + \frac{N_{3K_1}}{N} + \frac{2}{6}\cdot\frac{2N_4}{N} \\
&+ \frac{N_{4K_1}}{N} + \frac{1}{6}\cdot\frac{2N_5}{N} + \frac{N_{5K_1}}{N} + \frac{N_{6K_1}}{N} - 1 \\
\Delta N = &\ 1
\end{aligned}$$

In the above equations, ΔN_i is the change in the average number of nodes in the ith class, and ΔN_{iK_1} shows the change in the average number of nodes in the intermediate urn associated with the class i. The N_is, and N_{iK_1}s show the average number of nodes in each urn, and N is the total number of nodes at each step. Since in this case $K=2$, this means that there is a probability of $2\times 1/N$ that a node in each of the original urns receive a frozen input from all N possible choices and leaves the original urn. That is why the change in $\Delta N_i = -2N_2/N$. For each intermediate urn, the change in their average number of nodes is of the form $(1-P_{1i})\cdot 2N_i/N - N_{iK_1}/N$. This means that of the $2N_i/N$ nodes that have already received a frozen input and have become effectively a $K=1$ ternary function, $1-P_{1i}$ of them does not become frozen and end up in the corresponding intermediate urn, and P_{1i} of them become frozen and are moved to the frozen urn. Also, all the nodes in the intermediate urns will become frozen upon receiving another frozen input, appearing as $-N_{iK_1}/N$ term. The pruning process stops when the effect of all the frozen nodes on the other nodes of the network has been considered, i.e. $N_F \leq 0$.

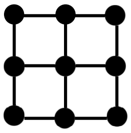
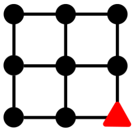
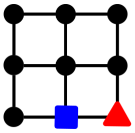
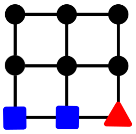
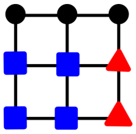
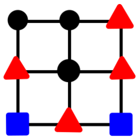
Sample Function						
P_0	1	0	0	0	0	0
P_1	1	4/6	3/6	2/6	1/6	0
Class Size	3	54	264	2484	7128	9750

Table 4.2: The 6 different canalization preserving classes of ternary functions with $K = 2$, along with the corresponding canalization values P_0 and P_1, and number of functions in each class. A geometric representation of a sample function from each function class is also shown.

P_is are the canalization values of the functions. Table 4.2 shows the canalization values of the ternary functions with two inputs per nodes. In the table, a sample function is chosen from each of the function classes with the same canalization value. The canalization values of the ternary functions, like Boolean functions [84], can be easily calculated by looking at their geometric representation. Each ternary function can be represented with a different coloring of a *hypergrid*. The dimension of the hypergrid depends on the number of in-links per node, K, and the color of a grid point corresponds to the output value of the function for that particular input set, which is in fact the coordinates of that grid point. Increasing S results in having more coloring for the grid points. A function with S possible output states requires S different colorings for the grid points.

Figure 4.3 shows the results of solving Eq. 4.5 and comparing it with direct simulations. As it can be seen in the figure the numerical solutions of Eq. 4.5 does not predict the direct simulation results precisely. This is because we have not considered the effects of fluctuations which can be done using the method applied to random Boolean networks in chapter 2 and 3. A noise term is added to the deterministic equations. Since the number of nodes in the frozen urn becomes very small at the end of the process, the effects of fluctuations is large in that urn. Therefore, we add a Gaussian noise term with zero mean and variance one to the equation for the frozen urn [61]. We have

$$\begin{aligned}\Delta N_F &= \frac{4}{6}\cdot\frac{2N_2}{N} + \frac{N_{2K_1}}{N} + \frac{3}{6}\cdot\frac{2N_3}{N} + \frac{N_{3K_1}}{N} + \frac{2}{6}\cdot\frac{2N_4}{N} \\ &\quad + \frac{N_{4K_1}}{N} + \frac{1}{6}\cdot\frac{2N_5}{N} + \frac{N_{5K_1}}{N} + \frac{N_{6K_1}}{N} - 1 + \xi \end{aligned} \tag{4.6}$$

where $\langle\xi\rangle$=0, and Var(ξ)=1. It should be noted, however, that unlike the critical Boolean networks where the adding the Gaussian noise term leads to accurately predicting the direct simulation results [56], for ternary networks we get the best fit by multiplying a coefficient of 0.55 to the Gaussian noise term. The error bars on this coefficient are not clear, and given the existence of statistical and potential systematic errors the possibility of 0.5 as the coefficient is not excluded. Figure 4.3 compares the results of direct simulations and the deterministic equations with and without fluctuations.

Equation 4.2 and can be generalized to higher S (and K). In general, critical biases are given by the points on a $S-1$-sphere inscribed into a $(S-1)$-simplex where the radius of the sphere depends on K. An S-simplex is a generalization of a triangle in S dimensions. A similar pruning process can also be set up to calculate the frequency of different function types on the active nodes.

We should also note that as S increases from 2 the effects of fluctuations become less important. In fact, if we solve the equations governing the average number of nodes in different urns for $S=4$, with and without the effects of fluctuations, we find that the difference is of the same order as of the $S=3$ networks ($\simeq 0.1$). This may mean that as S increases the fluctuations in the system cancel each other out and the mean-field results become more accurate.

4.4 Summary

We can conclude form this chapter and the previous two chapters that canalization preserving symmetry controls the critical discrete multi-state networks dynamics, as well as critical random Boolean networks dynamics. We learn that critical dynamics leads to a symmetry of the multi-value functions in a very similar way as it does for Boolean networks: Both numerical simulations and analytics suggested that the ratio of active nodes with particular Boolean function depends on the function's canalization properties. For a critical discrete multi-state networks with S possible states per node ($S>3$), there are $2\times S$ canalization preserving classes that determine a symmetry of the dynamics of the critical state. The canalization values of these classes depend on the value of K as well. If we fix $K=2$, for instance, then there are two canalization values for each class, P_0 and P_1. These values will be (from the highest to the lowest): (P_0, P_1)=(1,1), (0, $\frac{S-2}{2S}$), (0, $\frac{S-3}{2S}$), ..., (0, $\frac{1}{2S}$), (0, 0). All the functions that have the same total canalization value ($P_0 + P_1$) constitute a symmetry class, and occur with the same frequency on the active nodes of the network if normalized by the initial number of those function on all the nodes of the network.

The existence of the same symmetry for critical random Boolean networks dynamics and critical multi-state dynamics suggests that both of these models belong to the same universality class of heterogenous complex systems. To investigate this idea explicitly, in the next chapter we study a critical exponent of these two models to see if indeed they have the same critical exponent.

Chapter 5

Attractor length distribution of critical multi-state networks

5.1 Introduction

In this chapter we explicitly show that a universality class of heterogeneous complex systems exists by studying a particular scale invariant property of critical random Boolean networks and random multi-state networks, and showing that its scaling behavior is the same for a range of models. We find a dynamical critical exponent, α, for dynamical attractor length distribution of discrete state networks which include random Boolean networks (RBNs) and random multi-state networks (RMN) [92]. The dynamical attractor length distribution is an important dynamical characteristics of discrete state networks which gives us information about the dynamics of the network. It is an example of a quantity that is scale-invariant at the critical point in discrete state networks. Universality of this dynamical behavior is shown by considering three cases: dynamics of critical RBNs, dynamics on critical RBNs, and the dynamics of RMNs.

The numerical simulation results for the three cases mentioned are presented. We measure their critical exponent α at the large network size limit in addition to the corrections to the scaling exponent in each case. Then based on the structural properties of relevant nodes of RBNs at the critical point, we propose an analytical plausibility argument that predicts the critical exponent describing the distribution of attractor lengths of RBN. This argument is then generalized to other discrete state networks. Finally, we propose some fundamental features that characterize this universality class of heterogenous complex networks.

5.2 Numerical simulations

A dynamical feature of discrete state networks is that the sequence of the states of a network of size N is eventually periodic. As discussed in chapter 1 this directly follows from the facts that (i) the overall number of possible states, 2^N, is finite and (ii) the dynamics of the state of the nodes are deterministic. A repeating sequence of T states is called an *attractor* of length T [35]. There can be one, or more attractors, each of which has a basin of attraction corresponding to the region of the state space from which the dynamics eventually collapses onto that attractor. The dynamics of the network after the transient phase can be quantified by the number of attractors ν, their lengths T_j, and the size of their basins of attraction A_j with $j = 1, 2, \dots, \nu$.

The topic of determining the average attractor length $\langle T \rangle$ and the average number of attractors $\langle \nu \rangle$ in *critical* RBNs has a long history [58, 13, ?, 90]. It has been proven that both of these quantities increase with network size N faster than any power-law [88, 61]. However, determining the attractor length distribution has received less attention. This quantity is particularly important though, because it can be used to characterize different phases of dynamical behavior [12]. At the critical point, predicting the attractor length distribution will lead to the calculation of a critical exponent.

A numerical algorithm to study the attractor length distribution of Boolean networks was proposed in [20]. Using that algorithm and studying networks of up to $N{\sim}10^5$ nodes, the authors found that the attractor lengths are power-law distributed for unbiased random Boolean networks with $K=2$ in-links per node. Another study deals with biased random Boolean networks with $K=4$ in-links, where the outputs of the Boolean functions of the nodes are biased towards 1 or 0 [13, 12, 11]. For this case also a power-law attractor length distribution was found numerically on the critical line between the fixed and chaotic phases. This scale-free behavior is not restricted to the dynamics of critical RBNs. The attractor lengths are also observed to be power-law distributed in the self-organized stationary state of Boolean networks which is obtained when the nodes are competing in an evolutionary game [78]. In this game, the set of node functions $\mathcal{F}$ are evolved through a competition that punishes the majority behavior, and it was concluded that the stationary state is in fact critical [64, 63, 10].

These numerical findings suggest that there exists a class of complex networks

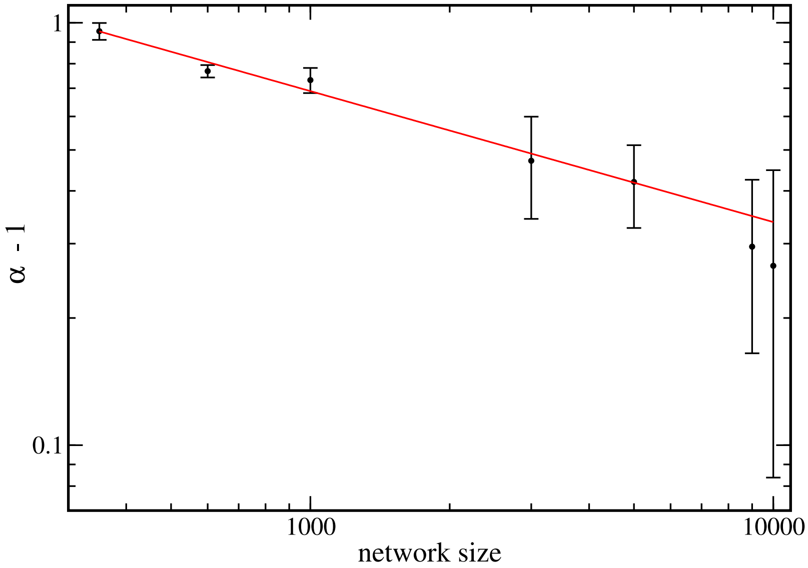


Figure 5.1: The exponents of the power-law distribution of attractor lengths are plotted for different network sizes of critical RBNs. It shows that $\alpha(N) - 1$ goes to zero for large network sizes. This means that critical exponent $\alpha(N)$ is 1 in the limit of large networks. The fitted line has the slope of $b = -0.31 \pm 0.04$ corresponding to the correction to the scaling exponent in this case. The error bars are 2 times standard deviation over network realizations.

with the same dynamical behavior. In this section we show computationally that all the above cases in fact have the same critical exponent of attractor length distribution and the correction to the scaling exponent is determined for each case. There are large finite size corrections to the scaling exponent which is different for different dynamics. Furthermore, the critical exponent of the random ternary networks, as a first step of generalization of RBNs to multi-state networks, is found. This shows that the breadth of the dynamics in this universality class is much larger than just the critical RBNs dynamics.

Figure 5.1 shows how the critical exponent α changes for different critical random Boolean networks of different sizes. The bias of the networks is adjusted to the critical bias to construct an ensemble of critical RBNs. Networks have been updated from different random initial states. The equation $\alpha(N) - 1 \sim N^{(-0.31\pm0.04)}$ shows the relation between the exponent of the power-law distributions and the network size. In order to calculate each exponent, Marquardt-Levenberg non-linear fitting [19] is used on the logarithmic binned data. For all different network sizes the logarithmic binning starts from 11, and the last data point which correspond to the attractor length cut off is always omitted. Non-linear fitting is also used to measure the equation for $\alpha(N)$. This fitting method is used for all the cases considered here.

The equation for $\alpha(N)$ implies that the exponent approaches 1 for large network sizes, and predict a critical of exponent of 1 for the dynamics of RBNs. It should be noted here that we did these simulations for critical RBNs with $K = 3$ inputs per node which is different from previous studies. Those studies suggested a critical exponent of 1, in the large network size limit, for networks with 2 and 4 inputs per node. The corrections to the scaling exponent depends on the number of inputs per node, K.

We have also measured the attractor length distribution of the networks that have been evolved to criticality through an evolutionary game [78]. In this game nodes get a negative score at a given time step, if they have the same output as the majority of the nodes. After some time steps, called an *epoch*, the node with the highest negative score is penalized by replacing its assigned Boolean function by another randomly chosen one. After a number of epochs the network reaches a steady state. In order to study the attractor length distribution of these networks at the steady state, we use the same attractor finding method as for the dynamics

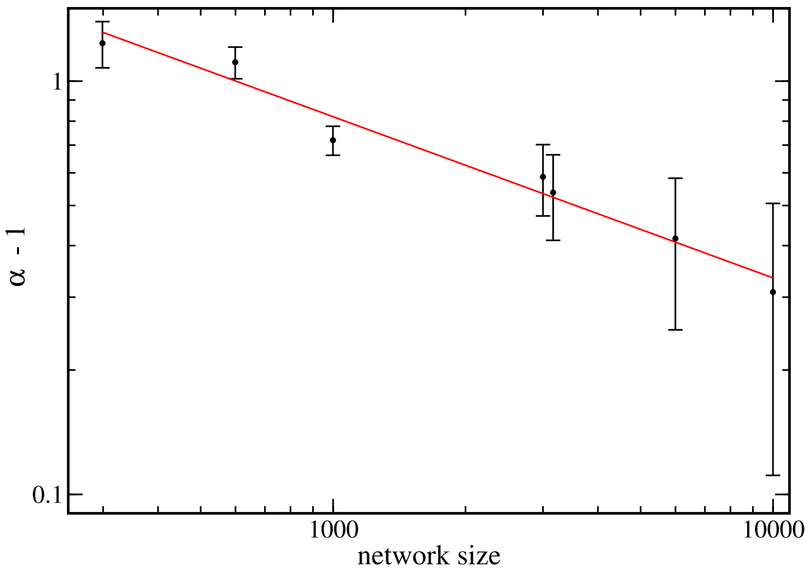


Figure 5.2: The exponents of the power-law distribution of attractor lengths are plotted for different network sizes of the evolutionary game introduced in Ref. [78]. $\alpha(N)-1$ goes to zero for large network sizes, showing that the critical exponent $\alpha(N)$ goes to 1 in the limit of large networks. The exponents for each case are calculated using the same method as for random Boolean networks dynamics and critical ternary networks. The slope of the red fitted line is $b = -0.39 \pm 0.06$ corresponding to the correction to the scaling exponent of the game. The error bars are 2 times standard deviation over network realizations.

of RBNs. The networks are updated from 1 000 000 random initial states, after they have reached the steady state, and the length of the attractors are then measured. Again, the attractor length distribution obeys a power-law distribution, this time with a different exponent than what was reported for a similar network size in the other two cases. Figure 5.2 shows the exponents calculated for different network sizes with the equation $\alpha(N) - 1 \sim N^{(-0.39 \pm 0.06)}$ relating the exponents with the network size. Interestingly enough, it can be seen that for large network sizes the exponent α approaches 1.

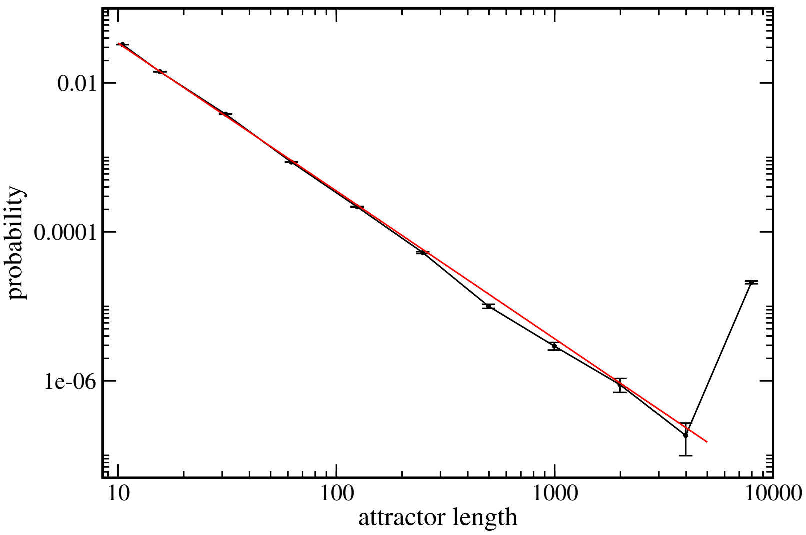


Figure 5.3: The ensemble-averaged probability distribution of the attractor lengths of $K = 2$ critical random ternary networks of size $N = 300$. It is averaged over 10 000 networks. The cut-off of length of the attractors is 10 000, and all the attractors not found after 10 000 updates are considered to have this length. The red line shows the fitting curve on the graph, and has the slope of -1.54 ± 0.07. The error bars are 2 times standard deviation over network realizations.

Figure 5.3 shows the average attractor length distribution for an ensemble of $K = 2$ critical random ternary networks. Ternary networks are multi-state networks where each node has 3 possible output values, 0, 1, and 2 for instance. Critical random ternary networks can be constructed by biasing the output values of the functions to the appropriate critical biases [107]. The critical biases of 1/6, 1/6, and 2/3 are chosen from the criticality circle introduced in chapter 4. Since there is no fundamental difference between the output values 0, 1, and 2 we impose a symmetry among them, by randomly biasing the output values of the functions to each of these values. In other words, bias 1/6 is randomly chosen to be toward 0, 1, or 2.

We used the method described in [63] to find the attractor lengths of the network

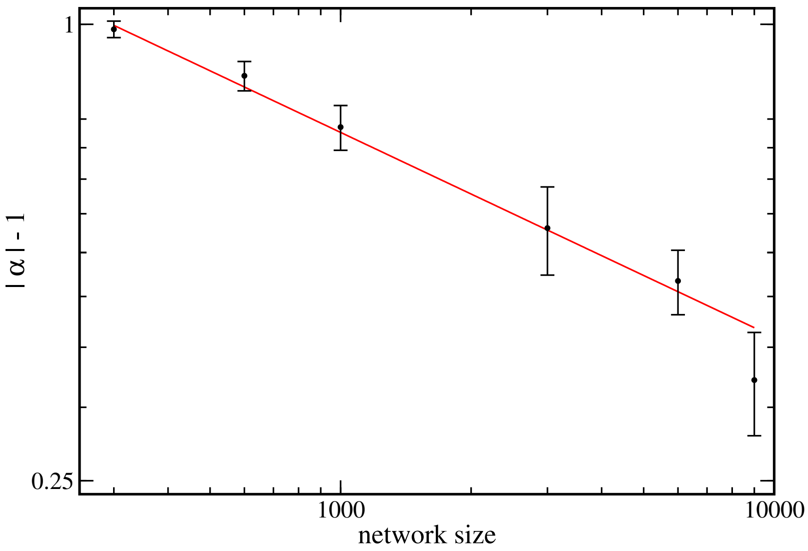


Figure 5.4: The exponent of the power-law distribution of attractor length is plotted for different network sizes of critical random ternary networks. It can be seen that $\alpha(N) - 1$ goes to zero for large network sizes. This shows that critical exponent is 1 in the limit of large networks. The slope of the fitted line which shows the correction to the scaling exponent is $b = -0.27 \pm 0.02$. The error bars are 2 times standard deviation over network realizations.

for each initial state. From Fig. 5.3 it can be seen that the attractor length distribution is power-law distributed, $P(\ell) = C_N \ell^{-\alpha(N)}$, with an exponent that depends on the network size. This is in fact true for all the cases we consider here. These simulations have been repeated for different network sizes. For all network sizes we obtain power-law distributions of the attractor lengths with different exponents, as is depicted in Fig. 5.4. This finding suggests that there are finite size corrections to the scaling behavior. The equation for the exponent $\alpha(N)$ approaches 1 for large network sizes, and predict a critical of exponent of 1 for the dynamics of random ternary networks.

These numerical results confirms that the attractor length distribution of critical random Boolean networks follow a scale-free distribution. Also, simulations showed the same dynamical behavior for the self-organized critical state of the evolutionary game played on random Boolean network, and for random ternary networks. More importantly the critical exponent is 1 in the large network size limit for all these different cases. The question that naturally arises is why this critical exponent is 1, and what are the common properties of these different systems leading to this universal behavior? In the next section we tackle these questions from a more analytical point of view.

5.3 An analytic explanation

Here we first analyze the attractor length distribution of critical random Boolean networks. We argue that the critical exponent of 1 is the consequence of structural and dynamical properties of random Boolean networks at the critical point. We propose a plausibility argument built based on these properties and show that it correctly captures the behavior of dynamical attractors of critical random Boolean networks and predict the critical exponent. Furthermore, we claim that this is a plausible method for describing the dynamical attractor length distribution of all networks in this universality class because of their expected properties at the critical point.

In order to do this, we are going to consider the *relevant components* of the network. From previous work it has been shown that the set of relevant nodes can be partitioned into components that are disjoint subgraphs of the network, called the relevant components [60]. The dynamics of each relevant component is independent of the others.

We are going to assume that all relevant components are simple loops. This is supported by previous numerical findings that in the limit of large network sizes the number of relevant nodes scales as $N^{1/3}$ with the number of nodes N, where a finite number of them have two relevant inputs and a vanishing proportion of them have more than two inputs [68]. From this, it follows that all relevant components except a finite number of them are simple loops without additional links, also known as simple components. In the limit of large network sizes, it is only the largest loop that is not simple, i.e. it has additional cross-links [60].

Another assumption that we make in our argument is that in critical RBNs the size of the basin of attraction does not depend on the length of the attractor, which is supported by previous numerical findings [17]. We call this assumption the *equipartition of the basins of attraction* assumption.

Finally, the third major assumption that we make is what we call *length distribution of simple loops* assumption. It is based on an essential result from previous studies is that the number of simple loops of length ℓ in the large network size limit, $N \to \infty$, is Poisson distributed with a mean $1/\ell$ [34]. Using this result, the probability of having a loop of length ℓ can be approximated with $1/\ell$ (times a coefficient) in the limit of large network size.

With these assumptions we want to calculate the attractor length distribution of critical RBNs. Since the dynamics of a critical Boolean network after the transient

input	output
0	0
1	1

copy function

input	output
0	1
1	0

invert function

Figure 5.5: Copy and invert functions.

times is determined only by the relevant nodes of the network, all possible attractor lengths of a network realization can be calculated by combinatorics involving the possible periods of the dynamics of the relevant components. More specifically, attractor lengths of a network are the least common multiple (LCM) of the possible periods of the relevant components of the network realization. For example, if there are only two relevant components with possible periods $\pi_1 \in \{2, 3, 6\}$ and $\pi_2 \in \{5\}$, then the possible attractor lengths of the network are $T \in \{10, 15, 30\}$. Since in the large network size limit, all the relevant components except a finite number of them are simple relevant components [60, 68], we make the approximation that in the thermodynamic limit, $N \to \infty$, *all* relevant components are simple loops, and only focus on the possible periods of these relevant components.

In a simple loop consisting of ℓ nodes, every node has exactly one input from the preceding node with the closing condition $s_{\ell+1} \equiv s_1$. In this case, the behavior of a node is determined by the input it receives from the preceding node which means it is effectively a node with $K = 1$ in-links. None of the nodes in the loop can have a constant Boolean function, because constant outputs would block the loop and immediately lead to a fixed point attractor. Thus, there are only two possible Boolean coupling functions to be assigned to each node, namely to *copy* or *invert* the state of the predeceasing node [35]. Figure 5.5 shows the copy and invert functions.

Loops are labeled either even or odd depending on the number of invert functions they contain. Moreover, any odd loop can be mapped to a loop with only a single invert function. This mapping does not change the number and the length of possible periods of the loop [35]. Every synchronous update of a simple loop can be imagined as an incremental rotation of the whole loop configuration. Therefore, in an even loop with ℓ nodes, after ℓ updates the same configuration is reached again. In an odd loop, on the other hand, because of the invert function, the same configuration is reached again after 2ℓ updates.

We should note however, that loops with a non-prime-number of nodes can have shorter attractor lengths. For example, on an even loop with $\ell = 4$ nodes with only copy functions a possible pattern is '0101' and it has a period of 2. In fact, for a simple loop of length ℓ the possible shorter periods of the loop are elements of its *divisor set* $D(\ell)$. (For an odd loop the possible shorter periods are twice the elements of its divisor set.)

In order to continue we are going to introduce *pseudo critical networks* which takes into account our assumptions, and allow us to calculate attractor length distributions for large network sizes. A given realization of a critical network can be simplified using a set of loops of different lengths, each corresponding to a relevant component. We call this set of loops a *pseudo critical networks.* As mentioned before, since the dynamics of a critical RBNs after the transient times is determined solely by the relevant nodes of the network, pseudo critical networks are all we need to determine the distribution of the lengths of the dynamical attractors of the network. Now, since the attractor lengths of the network are the LCM of the periods of the relevant loops, the attractor length distribution of a critical network will be the same as the distribution of the periods of the loops of the pseudo critical network. Therefore, first we need to find the probability of each of the possible periods of a loop with a certain length, and then based on that calculate the probability distribution of a given period in a pseudo critical network.

In order to find the probability of each of the possible periods of a simple loop, let us assume that the loop is even with length ℓ, and its set of divisors is $D(\ell)$. Since every update of a loop is equal to a rotation of the sequence of the outputs of the nodes on the loop, a loop of length ℓ with period $\ell_i \in D(\ell)$ is equivalent to a binary string of length ℓ with period ℓ_i under cyclic permutation. Hence, the probability that the loop has period ℓ_i is equal to the probability of having a binary sequences with period ℓ_i. The number of binary sequences with period ℓ_i is known to be equal to the number of binary Lyndon words of length ℓ_i. A binary Lyndon word of length ℓ_i is defined as a ℓ_i-character binary string which is lexicographically smallest among cyclic permutations of its characters. The number of binary Lyndon words of length ℓ_i is given by

$$L_2(\ell_i) = \frac{1}{\ell_i} \sum_{d \in D(\ell_i)} \mu(d) 2^{\ell_i/d}$$

where μ is the Möbius function, and the sum runs over all the divisors of ℓ_i, $d \in D(\ell_i)$ [97].

The assumption of the equipartition of the basins of attraction leads to stating that all attractors have the same probability of being reached given a uniform initial state distribution. In *equipartition of the basins of attraction* assumption, we assume that all the unique binary strings of different period have the same probability of being picked. In other words, the multiplicity of a period of a certain length is only determined by the number of ways that a distinct binary string with that period can be built, and not by also considering the cyclic permutations of each distinct binary strings. This way we are not considering the transient times, and we are implying that in order to calculate the probabilities of different periods, we do not need to consider how many different ways we can end up on that attractor, but we just need to know how many distinct ways we can build an attractor of a certain length. In fact, this assumption is based on what was found for the relation between the size of the basin of attraction and the attractor length of critical RBNs [17]. The basins of attraction in the network state space correspond to a combination of outgoing trees attached to simple loops in the state space. If all the attractors have the same probability of being reached, all possible unique periods of the simple loops should have the same probability of being reached since the attractor length of the network is nothing but the least common multiple of the simple loops.

The possible periods of a simple loop are all the divisors of ℓ. Given the equipartition of the basins of attraction assumption, for an even loop we have

$$\rho_{\ell}^{even}(\ell_i) = \frac{L_2(\ell_i)}{\displaystyle\sum_{\ell_j \in D(\ell)} L_2(\ell_j)}$$

where $\rho_{\ell}^{even}(\ell_i)$ is the probability of having a period ℓi for an even loop of length ℓ. In the denominator, the outer sum is over all divisors of ℓ, $\ell_j \in D(\ell)$.

A corresponding equation can be set up for an odd loop with the only difference being that at each time step in addition to the rotation of the sequence of the outputs, the output values that pass through the only "invert" function are negated as well. Hence, we have to change the possible periods in the equations to twice of what we have considered for the even loops, $\ell_j \in 2\times D(\ell)$.

Now we can calculate the probability distribution of a given period in a pseudo critical network. The probability of having a period of length ℓ is equal to the probability of having an even loop of either (1) length of ℓ with period ℓ or (2) length ℓ_i with period ℓ where $\ell \in D(\ell_i)$ plus the probability of having an odd loop of either (3) length $\ell/2$ with period ℓ or (4) length ℓ_i with period ℓ where $\frac{\ell}{2} \in D(\ell_i)$. Furthermore, the probability of having an even (odd) loop of certain length and specified period is equal to the probability of having a loop of the desired length times the probability of having the specified period for that loop. Therefore, we have

$$\begin{aligned} P(\ell) = & \, p(\ell).\, \rho_{\ell}^{\text{even}}(\ell) + \sum_{\ell_i} p(\ell_i).\, \rho_{\ell_i}^{\text{even}}(\ell) \\ & + \Big((\ell{+}1) \bmod 2\Big)\Big[p(\ell/2).\, \rho_{\ell/2}^{\text{odd}}(\ell/2) + \sum_{\ell_i} p(\ell_i).\, \rho_{\ell_i}^{\text{odd}}(\ell/2)\Big] \end{aligned} \tag{5.1}$$

where the probability of having a loop of length ℓ is shown by $p(\ell)$. The factor behind the third and forth terms guarantees that we add those terms only for even ℓs, otherwise, $\ell/2$ is not an integer. Now using our last assumption, the length distribution of simple loops assumption, the probability of having a loop of length ℓ can be approximated with $1/\ell$ (times a coefficient) in the limit of large network size.

Plugging these into equation 5.1 we have:

$$\begin{aligned}
P(\ell) \sim & \frac{\rho_\ell^{\text{even}}(\ell)}{\ell} + \sum_{\ell_i} \frac{1}{\ell_i}.\, \rho_{\ell_i}^{\text{even}}(\ell) \\
& + \Big((\ell+1) \bmod 2\Big)\Big[\frac{1}{\ell/2}.\, \rho_{\ell/2}^{\text{odd}}(\ell/2) + \sum_{\ell_i}\frac{1}{\ell_i}.\,\rho_{\ell_i}^{\text{odd}}(\ell/2)\Big] \\
\sim & \frac{1}{\ell}.\, \frac{\frac{1}{\ell}\sum\limits_{d\in D(\ell)} \mu(d)2^{\ell/d}}{\sum\limits_{\ell_j\in D(\ell)} \frac{1}{\ell_j}\sum\limits_{d\in D(\ell_j)}\mu(d)2^{\ell_j/d}} \\
& + \sum_{\ell_i} \frac{1}{\ell_i}.\, \frac{\frac{1}{\ell}\sum\limits_{d\in D(\ell)} \mu(d)2^{\ell/d}}{\sum\limits_{\ell_j\in D(\ell_i)} \frac{1}{\ell_j}\sum\limits_{d\in D(\ell_j)}\mu(d)2^{\ell_j/d}} \\
& + \Big((\ell+1) \bmod 2\Big)\frac{1}{\ell/2}.\, \frac{\frac{1}{\ell/2}\sum\limits_{d\in D(\ell/2)} \mu(d)2^{\ell/2d}}{\sum\limits_{\ell_j\in D(\ell/2)} \frac{1}{\ell_j}\sum\limits_{d\in D(\ell_j)}\mu(d)2^{\ell_j/d}} \\
& + \Big((\ell+1) \bmod 2\Big)\sum_{\ell_i}\frac{1}{\ell_i}.\, \frac{\frac{1}{\ell/2}\sum\limits_{d\in D(\ell/2)} \mu(d)2^{\ell/2d}}{\sum\limits_{\ell_j\in D(\ell_i)} \frac{1}{\ell_j}\sum\limits_{d\in D(\ell_j)}\mu(d)2^{\ell_j/d}}
\end{aligned} \tag{5.2}$$

Equation 5.2 can be simplified further. First, we should note that as the network size becomes larger, the larger simple loops appear. In the limit of large loop length, $\ell \to \infty$, we can approximate the binary Lyndon word of length ℓ with its first term as

$$L_2(\ell) \simeq \frac{1}{\ell} 2^\ell$$

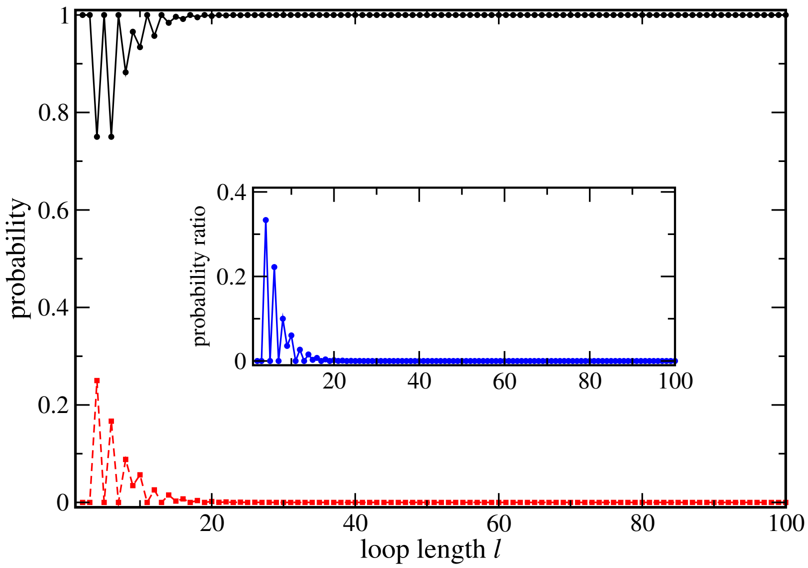


Figure 5.6: The probability that the period of an even loop is equal to its length (period=ℓ, black) is compared to the probability that the period of the loop is equal to its second largest divisor (period=ℓ/q where q is the first prime factor of ℓ, dashed red). The inset shows the ratio of these two probabilities. It can be seen that this ratio goes to zero very rapidly. This implies that the most probable period of a simple loop is its length and this probability approaches 1 very quickly.

and hence the probability that a loop of length ℓ has period ℓ_i is equal to

$$\rho_\ell(\ell_i) = \frac{L_2(\ell_i)}{\displaystyle\sum_{\ell_j \in D(\ell)} L_2(\ell_j)} \simeq \frac{L_2(\ell_i)}{L_2(\ell)} \simeq \frac{\ell}{\ell_i} 2^{\ell_i - \ell}$$

where in the denominator only the largest term, $L_2(\ell)$, is kept. Second, as Fig. 5.6 shows the probability that a loop has a period equal to its length is much larger than the probability of other periods. In fact, using the above approximation for large loops, the ratio of the largest period of a loop of length ℓ, ℓ, to the second largest period of the loop, ℓ/q, is equal to

$$\frac{\rho_\ell(\ell/2)}{\rho_\ell(\ell)} \simeq \frac{(\frac{\ell}{\ell/2})2^{\ell/2-\ell}}{2^{\ell-\ell}} \simeq 2^{1-\ell/2}$$

where q is the first prime factor of ℓ. As Fig. 5.6 shows, this ratio goes to zero

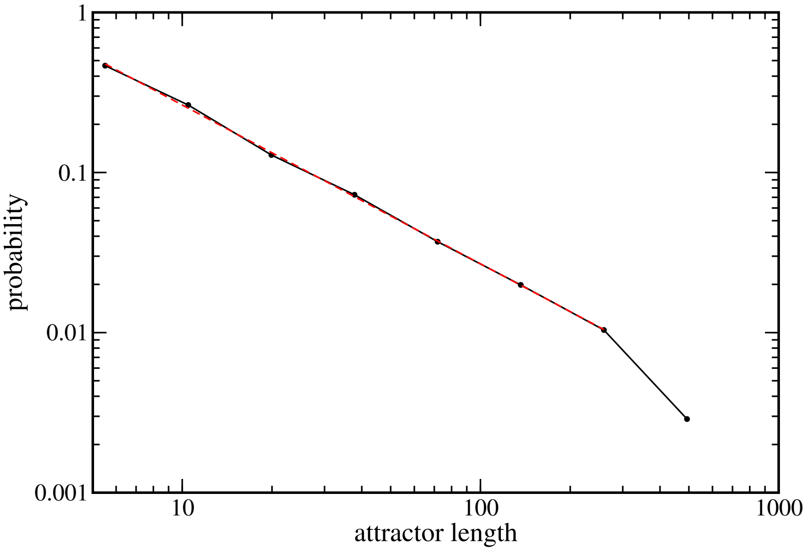


Figure 5.7: Probability distribution of attractor lengths using Lyndon words enumeration. Attractors of length up to 500 are considered. The data has been normalized to present the probability distribution up to the cut-off of 500. We have logarithmic binned the data from attractor length of 4 to the end. The red dotted line shows the non-linear fitting on the data, excluding the last point. This point corresponds to the cut-off of the attractor length. The reason that this point is suddenly decreasing is because we are not considering loops with lengths larger than 500 which would have contributed to the attractor lengths smaller than 500. The slope of the fitted line is $\alpha = -0.992 \pm 0.01$.

quickly as ℓ becomes larger. Therefore, in the limit of large loop length ($\ell > 10$) we can approximate the period of a loop with its length. Using these two results we can simplify equation 5.2 as

$$\begin{aligned} P(\ell) &\sim \frac{\rho_\ell^{\text{even}}(\ell)}{\ell} + \Big((\ell{+}1) \bmod 2\Big)\left[\frac{1}{\ell/2} . \, \rho_{\ell/2}^{\text{odd}}(\ell/2)\right] \\ &\sim \frac{1}{\ell} 2^{\ell-\ell} + \Big((\ell{+}1) \bmod 2\Big)\left[\frac{1}{\ell/2} 2^{\ell/2-\ell/2}\right] \\ &\sim \frac{1}{\ell} \end{aligned}$$

Figure 5.7 shows the behavior of $P(\ell)$ as a function of ℓ on a logarithmic plot. It is in fact the numerical plot of Eq. 2. It should be noted that data has been

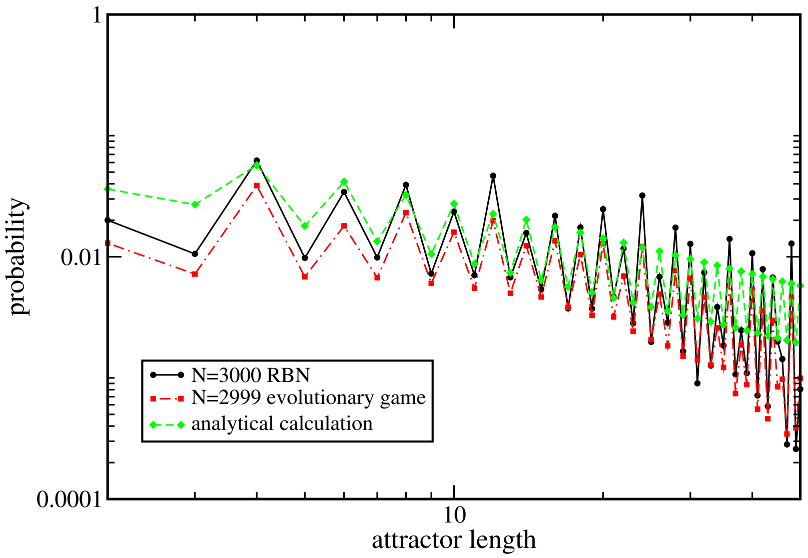


Figure 5.8: The histogram of attractor lengths of different dynamics: random Boolean networks dynamics and the evolutionary game. The results are also compared with the analytical predictions calculated based on the binary Lyndon words enumeration. The analytical results are proportional to probability distribution. It can be seen that the fluctuations between the even and odd attractor lengths is present for all lengths if there is no logarithmic binning.

log binned. If we do not log bin the attractor length distribution, we will see the oscillatory behavior corresponding to the difference between even and odd attractor lengths. As it can be seen from Eq. 2, $P(\ell)$ is higher compared to the $P(\ell-1)$ and $P(\ell+1)$ if ℓ is even. This is depicted in Fig. 5.8, where the oscillatory behavior is present for the numerical simulation results of the RBNs and the evolutionary game as well as the analytical calculations results.

5.4 Summary and discussion

We have shown that a class of models of heterogenous complex systems exists that displays universal critical scaling behavior. The same canalization preserving symmetry controls their dynamics and they have the same critical exponent for their attractor length distribution. This confirms that these models are in the same universality class. This universality class of heterogenous complex systems consist of all discrete state networks.

An analytical explanation based on the structure of critical random Boolean networks was presented to explain the scale-free behavior in Boolean networks. This argument is based on the structure of relevant nodes of the networks, that determine the dynamics of the networks. In large network limits these nodes form simple loops. Therefore, by calculating the periods of these loops, the attractor of the network can be determined as the least common multiple of these periods. The probability of different periods of the loops can be computed by enumerating the binary Lyndon words. Also, a new computational method, relevant component method, is introduced based on the analytical understanding of the critical random Boolean networks, which allows for computing the attractor distribution of networks in the thermodynamics limit.

In general, only a full state space enumeration of the dynamics allows to obtain the exact attractor distribution which is possible only for small system sizes. This is true even if an intelligent pruning algorithm is used to disregard irrelevant nodes and simulates only the dynamics of the relevant nodes for a given realization.

For this reason, previous studies of the attractor length distribution have relied on sampling which has its own potential problems [17]. One problem that can occur when generating attractor-statistics by sampling of various network realizations is undersampling. Undersampling occurs when simulations are done without any prior knowledge about the structure of the state space. Another known problem with sampling is that the frequency with which attractors are found depends on the size of their basin of attraction [88].

Our new method has neither the problem of undersampling nor the problem of being biased by the basin sizes, and allows to effectively study arbitrary large networks. This is because this method considers all the relevant components of the network, and the lengths of the loops are randomly chosen based solely on their probability of occurrence and this does not depend on the size of the basin of attraction. In other words, if we use our assumptions these pitfalls can be avoided.

Also, the numerical results of the relevant component method were compared with that of the direct simulations of critical random Boolean networks, the results of an evolutionary game played on random Boolean networks, and direct simulations of critical ternary networks. Ternary networks are discrete multi-state networks where each node have 3 possible output values [107], and can be considered as the first step to generalization of Boolean networks to multi-state networks. All variants of direct simulation show that in the large network size limit the attractor length distribution has a broad power-law distribution with a critical exponent of 1. There are finite size corrections for all cases, but at the large network size limit they behave similarly, and have the same critical exponent as we predicted analytically. We should mention that here we have reported the results for networks with $K = 3$ in-links per node, but the we expect the results to hold for networks with other connectivities as well. We expect the exponent of attractor length distribution of all critical discrete multi-state networks to merge to 1, although the corrections to scaling are not the same for different topologies and dynamics.

The fact that this critical exponent is the same through all these different networks and different dynamics is probably due to the structure of the critical networks, where relevant nodes to the dynamics of the network form simple relevant loops. Therefore, all the models that have the same structure and symmetry of dynamics at the critical point should fall into this universality class. Networks with discrete canalization preserving symmetry are in this class. In fact, it might be the case that continuous models with this symmetry might be in this class, but we have not studied these systems here.

Chapter 6

Summary and discussion

In this work we studied statistical properties of heterogeneous complex systems. We extended the ideas used in condensed matter physics to heterogenous complex systems. As prototypical examples of these systems we have looked at random Boolean networks [58] and random multi-state networks [92]. Random Boolean networks are mainly inspired by biology, but are useful as discretization of continuous systems whenever all elements display a switch-like behavior with fast transitions between a low and a high saturation value. These systems can range from biological systems to social and economical systems as well. Multi-state networks are a generalization of random Boolean networks where the the elements of the system can have more than two discrete values. We looked at these heterogenous systems at the critical point where the continuous phase transition happens. In condensed matter systems, symmetry and universality becomes important at the continuous phase transition.

In chapter 2, we studied the frequency of the Boolean functions assigned to the active nodes of critical networks. Active nodes, unlike frozen nodes, remain active and their output states change as network is updated. Analyzing the frequency ratio of the functions used by active nodes to the ones used by all node of the network reveals a symmetry of critical random Boolean networks dynamics. It shows that not all function types behave differently at the critical point. In fact, there are classes of functions that occur with the same frequency, indicating a dynamical similarity among the functions of the same class. We showed that there are 3 classes of functions for $K = 2$ and 10 classes of functions for $K = 3$ critical random Boolean networks. Furthermore, it was shown how using this symmetry beforehand could simplify the analysis of the dynamics.

In chapter 3, based on the findings of chapter 2, we studied the nature of the symmetry observed and described it in group theory language. It was shown that the common characteristic of the functions that are in the same class is their canalization values. Canalization [101] is a concept borrowed from developmental biology and in this context, it shows how robust the functions are to the changes of their inputs. For $K = 2$ networks we proposed a minimum symmetry group that describes the symmetry observed, and we called it the canalization preserving symmetry group. The general characteristics of the canalization preserving symmetry group were also described for higher Ks.

In chapter 4, we investigated if there are other models, in addition to random Boolean networks that have he same symmetry of the dynamics and are in the same universality class. We considered random multi-state networks, as a generalization of random Boolean networks, and studied their phase transition. We studied the criticality condition for random multi-state networks, and proposed a novel method of visualizing the phase space of these networks. The canalization preserving symmetry was shown to govern random multi-state networks dynamics as well. Although random multi-state networks have different number of function classes compared to random Boolean networks, the nature of the symmetry is the same, and the multi-state functions that are in the same class have the same canalization values. This suggests the existence of a universality class.

In chapter 5, we explicitly showed that a universality class exists by studying a particular scale invariant property of critical random Boolean networks and random multi-state networks, and showing that its scaling behavior is the same for a range of models. We studied attractor length distribution of critical random Boolean networks and critical multi-state networks. Since the state space of the network is finite and the dynamics is deterministic, the states of the network eventually fall on attractors. An attractor is a recurrent set of a network's states. The number and size of attractors are two key characteristics of the dynamics. We showed numerically (and in agreement with previous research) that the attractor length distributions of these critical networks obey power law distribution. We tracked this scale free distribution in critical multi-state networks and other dynamics of random Boolean networks as well, namely an evolutionary game played on random Boolean networks where the networks self-organize to a critical state. In fact, for all of these cases in the limit of large network size the exponent of the distribution is 1. In order to describe the universality of this critical exponent, we proposed an argument based on the structure of critical networks and the importance of relevant nodes in determining the dynamics of the network. Based on this and using the Lyndon words enumeration, we could predict the critical exponent of 1 in the limit of large network sizes. This confirms the existence of a universality class.

In this work we have shown that the ideas of universality and symmetry from condensed matter physics can be extended to heterogeneous complex systems. We have succeeded in determining the symmetry of critical discrete states networks and showing that it can be used as a useful approach in simplifying the analysis of the behavior of heterogeneous complex systems at their critical points. We have also found a universality class of heterogenous complex systems, and studied some of the critical properties of the systems in this universality class [9].

6.1 Applications to real-world systems

We conclude this section by some discussion of the importance of critical heterogenous complex systems in real world, and some speculations of how our work can be used to help in real world systems.

Criticality has been shown to be important in many diverse complex systems in nature. For example many active matter systems are critical or near critical. Active matter describes systems whose constituent elements consume energy and are thus out-of-equilibrium, and examples of which include flocks or herds of animals and birds [27], and components of the cellular cytoskeleton. Other examples of complex systems operating near criticality include solar flares [99, 79], rice piles [40], and neural avalanches [16]. However, not all of the complex systems are considered as heterogeneous complex systems.

Gene regulatory networks are examples of heterogeneous complex systems. The gene transcription networks of *S. cerevisiae*, *E. coli*, and *B. subtilis*, as well as the network of segment polarity genes of *D. melanogaster*, and the network of flower development of *A. thaliana* have been shown to operate close to criticality [7]. Gene expression dynamics in the macrophage of innate immune system has been shown to exhibit criticality [76] as well. A reason that these systems are critical might be due to the evolvability and robustness. The organized expression of the different genes in an organism is essential to sustain functionality under the random external perturbations. To cope with such external variability, the dynamics of the genetic network must possess two important properties: It must be robust enough to guarantee stability under a broad range of external conditions, and it must be flexible enough to recognize and integrate specific external signals that may help the organism to change and adapt to different environments [7]. This compromise between robustness and adaptability has been observed in dynamical systems operating at the edge of transition between ordered and chaotic phase.

Another advantage might be that at critical points the phenomenon of critical slowing down happens which means that the system takes more time to relax correlations as it approaches its critical point. Critical slowing down allows the system to generate a wide range of time scales, typically much larger than the natural time scales of the system. This might be a useful mechanism for the system to process information and respond over longer times than the network dynamics would normally allow [72].

To extend our ideas to real world systems, first we need to realize if a system is critical. If it is critical we expect some symmetries to play a role in its dynamics. However, a single system possesses large fluctuations. Therefore, in this dissertation we considered ensembles of networks. To do an ensemble analysis in an experimental setting, we would need to consider an ensemble of different systems, and by considering the symmetry of their dynamics we could determine a fundamental nature of their evolution.

For example, if the full connectivity data of gene regulatory networks of an ensemble of biological organisms or even a particular subset of organisms were known, our methods could be used. Unfortunately, current data is limited and an analysis of this sort is not yet feasible. However, in the future, as the gene regulatory systems of more organisms are mapped out such analysis should become possible. In that case our methods would allow us to establish which symmetries control biological evolution and help us understand fundamental features of nature.

Appendix A

These are the mean-field equations for the average number of nodes in each of the urns set up for Boolean networks with $K = 2$ inputs per node. The equations are written for each of the Boolean function types shown in Fig. 3.1.

$$
\begin{aligned}
\Delta N_1 &= -2\frac{N_1}{N}, \\
\Delta N_2 &= -2\frac{N_2}{N}, \\
\Delta N_3 &= -2\frac{N_3}{N}, \\
\Delta N_4 &= -2\frac{N_4}{N}, \\
\Delta N_5 &= -2\frac{N_5}{N}, \\
\Delta N_6 &= -2\frac{N_6}{N}, \\
\Delta N_7 &= -2\frac{N_7}{N},
\end{aligned}
$$

$$\begin{aligned}
\Delta N_8 &= -2\frac{N_8}{N}, \\
\Delta N_9 &= -2\frac{N_9}{N}, \\
\Delta N_{10} &= -2\frac{N_{10}}{N}, \\
\Delta N_{11} &= -2\frac{N_{11}}{N}, \\
\Delta N_{12} &= -2\frac{N_{12}}{N}, \\
\Delta N_{13} &= -2\frac{N_{13}}{N}, \\
\Delta N_{14} &= -2\frac{N_{14}}{N}, \\
\Delta N_{1K1} &= \frac{1}{2}\cdot 2\frac{N_1}{N} - \frac{N_{1K1}}{N}, \\
\Delta N_{2K1} &= \frac{1}{2}\cdot 2\frac{N_2}{N} - \frac{N_{2K1}}{N}, \\
\Delta N_{3K1} &= \frac{1}{2}\cdot 2\frac{N_3}{N} - \frac{N_{3K1}}{N},
\end{aligned}$$

$$\begin{aligned}
\Delta N_{4K1} &= \frac{1}{2}\cdot 2\frac{N_4}{N} - \frac{N_{4K1}}{N}, \\
\Delta N_{5K1} &= \frac{1}{2}\cdot 2\frac{N_5}{N} - \frac{N_{5K1}}{N}, \\
\Delta N_{6K1} &= 2\frac{N_6}{N} - \frac{N_{6K1}}{N}, \\
\Delta N_{7K1} &= \frac{1}{2}\cdot 2\frac{N_7}{N} - \frac{N_{7K1}}{N}, \\
\Delta N_{8K1} &= \frac{1}{2}\cdot 2\frac{N_8}{N} - \frac{N_{8K1}}{N}, \\
\Delta N_{9K1} &= 2\frac{N_9}{N} - \frac{N_{9K1}}{N}, \\
\Delta N_{10K1} &= \frac{1}{2}\cdot 2\frac{N_{10}}{N} - \frac{N_{10K1}}{N}, \\
\Delta N_{11K1} &= \frac{1}{2}\cdot 2\frac{N_{11}}{N} - \frac{N_{11K1}}{N}, \\
\Delta N_{12K1} &= \frac{1}{2}\cdot 2\frac{N_{12}}{N} - \frac{N_{12K1}}{N}, \\
\Delta N_{13K1} &= \frac{1}{2}\cdot 2\frac{N_{13}}{N} - \frac{N_{13K1}}{N}, \\
\Delta N_{14K1} &= \frac{1}{2}\cdot 2\frac{N_{14}}{N} - \frac{N_{14K1}}{N},
\end{aligned}$$

$$\begin{aligned}
\Delta N_F &= -1+\frac{1}{2}\cdot 2\frac{N_1}{N}+\frac{1}{2}\cdot 2\frac{N_2}{N}+\frac{1}{2}\cdot 2\frac{N_3}{N}+\frac{1}{2}\cdot 2\frac{N_4}{N}+\frac{1}{2}\cdot 2\frac{N_5}{N}+\frac{1}{2}\cdot 2\frac{N_7}{N}+\frac{1}{2}\cdot 2\frac{N_8}{N}\\
&+ \frac{1}{2}\cdot 2\frac{N_{10}}{N}+\frac{1}{2}\cdot 2\frac{N_{11}}{N}+\frac{1}{2}\cdot 2\frac{N_{12}}{N}+\frac{1}{2}\cdot 2\frac{N_{13}}{N}+\frac{1}{2}\cdot 2\frac{N_{14}}{N}+\frac{N_{1K1}}{N}+\frac{N_{2K1}}{N}\\
&+ \frac{N_{3K1}}{N}+\frac{N_{4K1}}{N}+\frac{N_{5K1}}{N}+\frac{N_{6K1}}{N}+\frac{N_{7K1}}{N}+\frac{N_{8K1}}{N}+\frac{N_{9K1}}{N}+\frac{N_{10K1}}{N}\\
&+ \frac{N_{11K1}}{N}+\frac{N_{12K1}}{N}+\frac{N_{13K1}}{N}+\frac{N_{14K1}}{N}.
\end{aligned}$$

Appendix B

These are the mean-field equations for the average number of nodes in each of the urns set up for Boolean networks with $K = 3$ inputs per node. The equations are written for each of the Zyklenzeiger function classes shown in Fig. 2.6.

$$\begin{aligned}
\Delta N_B &= -3\frac{N_B}{N}, \\
\Delta N_{C_a} &= -3\frac{N_{C_a}}{N}, \\
\Delta N_{C_b} &= -3\frac{N_{C_b}}{N}, \\
\Delta N_D &= -3\frac{N_D}{N}, \\
\Delta N_E &= -3\frac{N_E}{N}, \\
\Delta N_{F_a} &= -3\frac{N_{F_a}}{N}, \\
\Delta N_{F_b} &= -3\frac{N_{F_b}}{N},
\end{aligned}$$

$$
\begin{aligned}
\Delta N_{F_c} &= -3\frac{N_{F_c}}{N}, \\
\Delta N_{G} &= -3\frac{N_{G}}{N}, \\
\Delta N_{H_a} &= -3\frac{N_{H_a}}{N}, \\
\Delta N_{H_b} &= -3\frac{N_{H_b}}{N}, \\
\Delta N_{I} &= -3\frac{N_{I}}{N}, \\
\Delta N_{J} &= -3\frac{N_{J}}{N}, \\
\Delta N_{BB_a} &= \frac{1}{2}\cdot 3\frac{N_B}{N} - 2\frac{N_{BB_a}}{N}, \\
\Delta N_{C_aB_a} &= \frac{2}{6}\cdot 3\frac{N_{C_a}}{N} - 2\frac{N_{C_aB_a}}{N}, \\
\Delta N_{C_aB_b} &= \frac{2}{6}\cdot 3\frac{N_{C_a}}{N} - 2\frac{N_{C_aB_b}}{N}, \\
\Delta N_{C_bB_a} &= \frac{2}{6}\cdot 3\frac{N_{C_b}}{N} - 2\frac{N_{C_bB_a}}{N}, \\
\Delta N_{C_bB_b} &= \frac{2}{6}\cdot 3\frac{N_{C_b}}{N} - 2\frac{N_{C_bB_b}}{N}, \\
\Delta N_{DB_a} &= \frac{2}{6}\cdot 3\frac{N_D}{N} - 2\frac{N_{DB_a}}{N}, \\
\Delta N_{DB_b} &= \frac{3}{6}\cdot 3\frac{N_D}{N} - 2\frac{N_{DB_b}}{N}, \\
\Delta N_{EB_b} &= \frac{4}{6}\cdot 3\frac{N_E}{N} - 2\frac{N_{EB_b}}{N}, \\
\Delta N_{EC} &= \frac{1}{6}\cdot 3\frac{N_E}{N} - 2\frac{N_{EC}}{N},
\end{aligned}
$$

$$\begin{aligned}
\Delta N_{F_aB_b} &= 3\frac{N_{F_a}}{N} - 2\frac{N_{F_aB_b}}{N}, \\
\Delta N_{F_bB_b} &= 3\frac{N_{F_b}}{N} - 2\frac{N_{F_bB_b}}{N}, \\
\Delta N_{F_cB_b} &= 3\frac{N_{F_c}}{N} - 2\frac{N_{F_cB_b}}{N}, \\
\Delta N_{GB_a} &= \frac{2}{6}\cdot 3\frac{N_G}{N} - 2\frac{N_{GB_a}}{N} \\
\Delta N_{GB_b} &= \frac{3}{6}\cdot 3\frac{N_G}{N} - 2\frac{N_{GB_b}}{N} \\
\Delta N_{GC} &= \frac{1}{6}\cdot 3\frac{N_G}{N} - 2\frac{N_{GC}}{N} \\
\Delta N_{H_aB_a} &= \frac{4}{6}\cdot 3\frac{N_{H_a}}{N} - 2\frac{N_{H_aB_a}}{N}, \\
\Delta N_{H_aC} &= \frac{2}{6}\cdot 3\frac{N_{H_a}}{N} - 2\frac{N_{H_aC}}{N}, \\
\Delta N_{H_bB_a} &= \frac{4}{6}\cdot 3\frac{N_{H_b}}{N} - 2\frac{N_{H_bB_a}}{N}, \\
\Delta N_{H_bC} &= \frac{2}{6}\cdot 3\frac{N_{H_b}}{N} - 2\frac{N_{H_bC}}{N}, \\
\Delta N_{IB_b} &= \frac{3}{6}\cdot 3\frac{N_I}{N} - 2\frac{N_{IB_b}}{N} \\
\Delta N_{IC} &= \frac{3}{6}\cdot 3\frac{N_I}{N} - 2\frac{N_{IC}}{N} \\
\Delta N_{JC} &= 3\frac{N_J}{N} - 2\frac{N_{JC}}{N}
\end{aligned}$$

$$
\begin{aligned}
\Delta N_{BB_aK_1} &= \frac{N_{BB_a}}{N} - \frac{N_{BB_aK_1}}{N} \\
\Delta N_{C_aB_aK_1} &= \frac{N_{C_aB_a}}{N} - \frac{N_{C_aB_aK_1}}{N} \\
\Delta N_{C_aB_bK_1} &= \frac{N_{C_aB_b}}{N} - \frac{N_{C_aB_bK_1}}{N} \\
\Delta N_{C_bB_aK_1} &= \frac{N_{C_bB_a}}{N} - \frac{N_{C_bB_aK_1}}{N} \\
\Delta N_{C_bB_bK_1} &= \frac{N_{C_bB_b}}{N} - \frac{N_{C_bB_bK_1}}{N} \\
\Delta N_{EB_bK_1} &= \frac{N_{EB_b}}{N} - \frac{N_{EB_bK_1}}{N} \\
\Delta N_{ECK_1} &= 2\frac{N_{EC}}{N} - \frac{N_{ECK_1}}{N} \\
\Delta N_{F_aB_bK_1} &= \frac{N_{F_aB_b}}{N} - \frac{N_{F_aB_bK_1}}{N} \\
\Delta N_{F_bB_bK_1} &= \frac{N_{F_bB_b}}{N} - \frac{N_{F_bB_bK_1}}{N} \\
\Delta N_{F_cB_bK_1} &= \frac{N_{F_cB_b}}{N} - \frac{N_{F_cB_bK_1}}{N} \\
\Delta N_{GB_aK_1} &= \frac{N_{GB_a}}{N} - \frac{N_{GB_aK_1}}{N} \\
\Delta N_{GB_bK_1} &= \frac{N_{GB_b}}{N} - \frac{N_{GB_bK_1}}{N} \\
\Delta N_{GCK_1} &= 2\frac{N_{GC}}{N} - \frac{N_{GCK_1}}{N}
\end{aligned}
$$

$$
\begin{aligned}
\Delta N_{H_aB_aK_1} &= \frac{N_{H_aB_a}}{N} - \frac{N_{H_aB_aK_1}}{N} \\
\Delta N_{H_aCK_1} &= 2\frac{N_{H_aC}}{N} - \frac{N_{H_aCK_1}}{N} \\
\Delta N_{H_bB_aK_1} &= \frac{N_{H_bB_a}}{N} - \frac{N_{H_bB_aK_1}}{N} \\
\Delta N_{H_bCK_1} &= 2\frac{N_{H_bC}}{N} - \frac{N_{H_bCK_1}}{N} \\
\Delta N_{ICK_1} &= 2\frac{N_{IC}}{N} - \frac{N_{ICK_1}}{N} \\
\Delta N_{JCK_1} &= 2\frac{N_{JC}}{N} - \frac{N_{JCK_1}}{N}
\end{aligned}
$$

$$
\begin{aligned}
\Delta N_F &= -1 + \frac{1}{2} \cdot 3\frac{N_B}{N} + \frac{2}{6} \cdot 3\frac{N_{C_a}}{N} + \frac{2}{6} \cdot 3\frac{N_{C_b}}{N} + \frac{1}{6} \cdot 3\frac{N_D}{N} + \frac{1}{6} \cdot 3\frac{N_E}{N} + \frac{N_{BB_a}}{N} \\
&+ \frac{N_{GB_aK_1}}{N} + \frac{N_{C_aB_a}}{N} + \frac{N_{C_aB_aK_1}}{N} + \frac{N_{C_aB_b}}{N} + \frac{N_{C_aB_bK_1}}{N} + \frac{N_{DB_a}}{N} + \frac{N_{DB_aK_1}}{N} \\
&+ \frac{N_{DB_b}}{N} + \frac{N_{DB_bK_1}}{N} + \frac{N_{EB_b}}{N} + \frac{N_{EB_bK_1}}{N} + \frac{N_{ECK_1}}{N} + \frac{N_{F_aB_b}}{N} + \frac{N_{F_aB_bK_1}}{N} \\
&+ \frac{N_{F_bB_b}}{N} + \frac{N_{F_bB_bK_1}}{N} + \frac{N_{F_cB_b}}{N} + \frac{N_{F_cB_bK_1}}{N} + \frac{N_{H_aB_a}}{N} + \frac{N_{H_aB_aK_1}}{N} + \frac{N_{H_aCK_1}}{N} \\
&+ \frac{N_{H_bB_a}}{N} - \frac{N_{H_bB_aK_1}}{N} + \frac{N_{H_bCK_1}}{N} + \frac{N_{ICK_1}}{N} + \frac{N_{JCK_1}}{N}.
\end{aligned}
$$

Bibliography

[1] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, Jan. 2002.

[2] R. Albert and H. G. Othmer. The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster*. *Journal of Theoretical Biology*, 223(1):1–18, July 2003.

[3] M. Aldana. Boolean dynamics of networks with scale-free topology. *Physica D: Nonlinear Phenomena*, 185(1):45–66, Oct. 2003.

[4] L. a. N. Amaral and J. M. Ottino. Complex networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, 38(2):147–162, Mar. 2004.

[5] P. W. Anderson. More is different. *Science*, 177(4047):393–396, Aug. 1972.

[6] P. Bak and K. Sneppen. Punctuated equilibrium and criticality in a simple model of evolution. *Physical Review Letters*, 71(24):4083–4086, Dec. 1993.

[7] E. Balleza, E. R. Alvarez-Buylla, A. Chaos, S. Kauffman, I. Shmulevich, and M. Aldana. Critical dynamics in genetic regulatory networks: Examples from four kingdoms. *PLoS ONE*, 3(6):e2456, June 2008.

[8] A. Barrat, M. Barthlemy, and A. Vespignani. *Dynamical Processes on Complex Networks*. Cambridge University Press, Cambridge, 1 edition edition, Nov. 2008.

[9] B. Barzel and A.-L. Barabsi. Universality in network dynamics. *Nature Physics*, 9(10):673–681, Oct. 2013.

[10] K. E. Bassler, C. Lee, and Y. Lee. Evolution of developmental canalization in networks of competing boolean nodes. *Physical Review Letters*, 93(3):038101, July 2004.

[11] U. Bastolla and G. Parisi. Closing probabilities in the Kauffman model: An annealed computation. *Physica D*, 98(1):1–25, 1996.

[12] U. Bastolla and G. Parisi. A numerical study of the critical line of Kauffman networks. *Journal of Theoretical Biology*, 187(1):117–133, July 1997.

[13] U. Bastolla and G. Parisi. The modular structure of Kauffman networks. *Physica D: Nonlinear Phenomena*, 115(34):219–233, May 1998.

[14] U. Bastolla and G. Parisi. Relevant elements, magnetization and dynamical properties in Kauffman networks: A numerical study. *Physica D: Nonlinear Phenomena*, 115(34):203–218, May 1998.

[15] M. Batty. The size, scale, and shape of cities. *Science*, 319(5864):769–771, Feb. 2008.

[16] J. M. Beggs and D. Plenz. Neuronal avalanches in neocortical circuits. *The Journal of Neuroscience*, 23(35):11167–11177, Dec. 2003.

[17] A. Berdahl, A. Shreim, V. Sood, M. Paczuski, and J. Davidsen. Random sampling versus exact enumeration of attractors in random boolean networks. *New. J. Phys.*, 11(4):043024, 2009.

[18] L. Bettencourt and G. West. A unified theory of urban living. *Nature*, 467(7318):912–913, Oct. 2010.

[19] P. Bevington and D. K. Robinson. *Data Reduction and Error Analysis for the Physical Sciences.* McGraw-Hill Science/Engineering/Math, Boston, 3rd edition edition, July 2002.

[20] A. Bhattacharjya and S. Liang. Power-law distributions in some random boolean networks. *Physical Review Letters*, 77(8):1644–1647, Aug. 1996.

[21] S. Bilke and F. Sjunnesson. Stability of the Kauffman model. *Physical Review E*, 65(1):016129, Dec. 2001.

[22] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424(4-5):175–308, Feb. 2006.

[23] N. Boccara. *Modeling Complex Systems.* Springer, New York, 2nd ed. 2010 edition edition, Sept. 2010.

[24] G. Boole. *The Mathematical Analysis of LogicBeing an Essay Towards a Calculus of Deductive Reasoning*. 1847.

[25] S. Bornholdt. Less is more in modeling large genetic networks. *Science*, 310(5747):449–451, Oct. 2005.

[26] S. Bornholdt and T. Rohlf. Topological evolution of dynamical networks: Global criticality from local dynamics. *Physical Review Letters*, 84(26):6114–6117, June 2000.

[27] A. Cavagna, A. Cimarelli, I. Giardina, G. Parisi, R. Santagati, F. Stefanini, and M. Viale. Scale-free correlations in starling flocks. *Proceedings of the National Academy of Sciences*, 107(26):11865–11870, June 2010.

[28] P. M. Chaikin and T. C. Lubensky. *Principles of Condensed Matter Physics*. Cambridge University Press, Sept. 2000.

[29] D. Challet and Y.-C. Zhang. Emergence of cooperation and organization in an evolutionary game. *Physica A: Statistical Mechanics and its Applications*, 246(34):407–418, Dec. 1997.

[30] R. Cohen and S. Havlin. *Complex Networks: Structure, Robustness and Function*. Cambridge University Press, New York, Aug. 2010.

[31] I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin. Effective leadership and decision-making in animal groups on the move. *Nature*, 433(7025):513–516, Feb. 2005.

[32] B. Derrida and Y. Pomeau. Random networks of automata: A simple annealed approximation. *EPL (Europhysics Letters)*, 1(2):45, Jan. 1986.

[33] B. Derrida and D. Stauffer. Phase transitions in two-dimensional Kauffman cellular automata. *EPL (Europhysics Letters)*, 2(10):739, Nov. 1986.

[34] B. Drossel. Number of attractors in random boolean networks. *Phys. Rev. E*, 72:016110–5, July 2005.

[35] B. Drossel. Random boolean networks. In H.-G. Schuster, editor, *Reviews of Nonlinear Dynamics and Complexity*, volume 1, pages 69–110. Wiley, 2008.

[36] G. W. Flake. *The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation*. A Bradford Book, Cambridge, Mass, Jan. 2000.

[37] H. Flyvbjerg. Recent results for random networks of automata. Technical report, Niels Bohr Inst., Copenhagen (Denmark), 1987.

[38] H. Flyvbjerg. An order parameter for networks of automata. *Journal of Physics A: Mathematical and General*, 21(19):L955–L960, Oct. 1988.

[39] H. Flyvbjerg and N. J. Kjr. Exact solution of Kauffman's model with connectivity one. *Journal of Physics A: Mathematical and General*, 21(7):1695, Apr. 1988.

[40] V. Frette, K. Christensen, A. Malthe-Srenssen, J. Feder, T. Jssang, and P. Meakin. Avalanche dynamics in a pile of rice. *Nature*, 379(6560):49–52, Jan. 1996.

[41] C. Gershenson. Updating schemes in random boolean networks: Do they really matter? In J. Pollack, M. Bedau, P. Husbands, T. Ikegami, and R. Watson, editors, *Artificial Life IX Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems*, pages 238–243. MIT Press, Feb. 2004. arXiv: nlin/0402006.

[42] N. Goldenfeld. *Lectures On Phase Transitions And The Renormalization Group*. Addison-Wesley, Reading, Mass, July 1992.

[43] D. Green, T. Leishman, and S. Sadedin. The emergence of social consensus in boolean networks. In *IEEE Symposium on Artificial Life, 2007. ALIFE '07*, pages 402–408. Institute of Electrical and Electronics Engineers (IEEE), 2007.

[44] F. Greil and B. Drossel. Dynamics of critical Kauffman networks under asynchronous stochastic update. *Physical Review Letters*, 95(4):048701, July 2005.

[45] D. Griffiths. *Introduction to Elementary Particles*. Wiley-VCH, Weinheim Germany, 2nd edition edition, Oct. 2008.

[46] T. Gross and B. Blasius. Adaptive coevolutionary networks: a review. *Journal of The Royal Society Interface*, 5(20):259–271, Mar. 2008.

[47] T. Gross, C. J. D. D'Lima, and B. Blasius. Epidemic dynamics on an adaptive network. *Physical Review Letters*, 96(20):208701, May 2006.

[48] T. Gross, L. Rudolf, S. A. Levin, and U. Dieckmann. Generalized models reveal stabilizing factors in food webs. *Science*, 325(5941):747–750, Aug. 2009.

[49] T. Gross and H. Sayama. Adaptive networks. In T. Gross and H. Sayama, editors, *Adaptive Networks*, pages 1–8. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[50] R. Hamming. Error detecting and error correcting codes. *Bell System Technical Journal, The*, 29(2):147–160, Apr. 1950.

[51] M. Hargittai. *Symmetry Through the Eyes of a Chemist.* Springer, 3 edition, 2009.

[52] M. A. Harrison. The number of transitivity sets of boolean functions. *Journal of the Society for Industrial and Applied Mathematics*, 11(3):806–828, Sept. 1963.

[53] M. A. Harrison. *Introduction to switching and automata theory.* McGraw-Hill, 1965.

[54] H. Hilhorst and M. Nijmeijer. On the approach of the stationary state in Kauffman's random boolean network. *Journal de Physique*, 48(2):185–191, 1987.

[55] C. S. Holling. Understanding the complexity of economic, ecological, and social systems. *Ecosystems*, 4(5):390–405, Aug. 2001.

[56] S. Hossein, M. D. Reichl, and K. E. Bassler. Symmetry in critical random boolean network dynamics. *Physical Review E*, 89(4):042808, Apr. 2014.

[57] L. P. Kadanoff. *Statistical Physics: Statics, Dynamics and Renormalization.* World Scientific Pub Co Inc, Singapore; River Edge, N.J., June 2000.

[58] S. Kauffman. Homeostasis and differentiation in random genetic control networks. *nature*, 224:177–178, Oct. 1969.

[59] S. A. Kauffman. *The origins of order: self-organization and selection in evolution.* Oxford University Press, New York, 1993.

[60] V. Kaufman and B. Drossel. Relevant components in critical random boolean networks. *New Journal of Physics*, 8(10):228, Oct. 2006.

[61] V. Kaufman, T. Mihaljev, and B. Drossel. Scaling in critical random boolean networks. *Phys. Rev. E*, 72:046124–9, Oct. 2005.

[62] F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang. The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences of the United States of America*, 101(14):4781–4786, Apr. 2004.

[63] M. Liu and K. E. Bassler. Emergent criticality from coevolution in random boolean networks. *Physical Review E*, 74(4):041910, Oct. 2006.

[64] M. Liu and K. E. Bassler. Finite size effects and symmetry breaking in the evolution of networks of competing boolean nodes. *Journal of Physics A: Mathematical and Theoretical*, 44(4):045101, Jan. 2011.

[65] B. D. MacArthur, R. J. Sánchez-García, and A. Ma'ayan. Spectral characteristics of network redundancy. *Physical Review E*, 80(2):026117, Aug. 2009.

[66] R. N. Mantegna and H. E. Stanley. *Introduction to Econophysics: Correlations and Complexity in Finance*. Cambridge University Press, Cambridge, UK ; New York, Nov. 1999.

[67] J. L. McCauley. *Dynamics of Markets: Econophysics and Finance*. Cambridge University Press, Cambridge, 1 edition edition, Apr. 2007.

[68] T. Mihaljev and B. Drossel. Scaling in a general class of critical random boolean networks. *Physical Review E*, 74(4):046101, Oct. 2006.

[69] J. H. Miller and S. E. Page. *Complex Adaptive Systems: An Introduction to Computational Models of Social Life*. Princeton University Press, Princeton, N.J, Mar. 2007.

[70] M. Mitchell. *Complexity: A Guided Tour*. Oxford University Press, Oxford; New York, Sept. 2011.

[71] M. Möller and B. Drossel. The formation of the frozen core in critical boolean networks. *New Journal of Physics*, 14(2):023051, Feb. 2012.

[72] T. Mora and W. Bialek. Are biological systems poised at criticality? *Journal of Statistical Physics*, 144(2):268–302, July 2011.

[73] M. Newman. *Networks: An Introduction*. Oxford University Press, Oxford ; New York, 1 edition edition, May 2010.

[74] M. Newman, A.-L. Barabsi, and D. J. Watts. *The Structure and Dynamics of Networks:*. Princeton University Press, Princeton, 1 edition edition, May 2006.

[75] M. E. J. Newman. Complex systems: A survey. *American Journal of Physics*, 79(8):800, 2011. arXiv: 1112.1440.

[76] M. Nykter, N. D. Price, M. Aldana, S. A. Ramsey, S. A. Kauffman, L. E. Hood, O. Yli-Harja, and I. Shmulevich. Gene expression dynamics in the macrophage exhibit criticality. *Proceedings of the National Academy of Sciences*, 105(6):1897–1900, Feb. 2008.

[77] N. H. Packard. *Adaptation Toward the Edge of Chaos.* University of Illinois at Urbana-Champaign, Center for Complex Systems Research, 1988.

[78] M. Paczuski, K. E. Bassler, and . Corral. Self-organized networks of competing boolean agents. *Phys. Rev. Lett.*, 84:3185, Apr. 2000.

[79] M. Paczuski and D. Hughes. A heavenly example of scale-free networks and self-organized criticality. *Physica A: Statistical Mechanics and its Applications*, 342(12):158–163, Oct. 2004.

[80] M. Paczuski, S. Maslov, and P. Bak. Avalanche dynamics in evolution, growth, and depinning models. *Physical Review E*, 53(1):414–443, Jan. 1996.

[81] S. Pandey, R.-S. Wang, L. Wilson, S. Li, Z. Zhao, T. E. Gookin, S. M. Assmann, and R. Albert. Boolean modeling of transcriptome data reveals novel modes of heterotrimeric g-protein action. *Molecular Systems Biology*, 6:372, June 2010.

[82] M. Pascual and J. A. Dunne, editors. *Ecological Networks: Linking Structure to Dynamics in Food Webs.* Oxford University Press, Oxford ; New York, Dec. 2005.

[83] U. Paul, V. Kaufman, and B. Drossel. Properties of attractors of canalyzing random boolean networks. *Physical Review E*, 73(2):026118, Feb. 2006.

[84] C. J. O. Reichhardt and K. E. Bassler. Canalization and symmetry in boolean models for genetic regulatory networks. *Journal of Physics A: Mathematical and Theoretical*, 40(16):4339, Apr. 2007.

[85] M. D. Reichl and K. E. Bassler. Canalization in the critical states of highly connected networks of competing boolean nodes. *Physical Review E*, 84(5):056103, Nov. 2011.

[86] M. Rybarsch and S. Bornholdt. Binary threshold networks as a natural null model for biological networks. *Physical Review E*, 86(2):026114, Aug. 2012.

[87] B. Samuelsson and J. E. S. Socolar. Exhaustive percolation on random networks. *Physical Review E*, 74(3):036113, Sept. 2006.

[88] B. Samuelsson and C. Troein. Superpolynomial growth in the number of attractors in Kauffman networks. *Physical Review Letters*, 90(9):098701, Mar. 2003.

[89] I. Shmulevich and S. A. Kauffman. Activities and sensitivities in boolean network models. *Physical Review Letters*, 93(4):048701, July 2004.

[90] J. E. S. Socolar and S. A. Kauffman. Scaling in ordered and critical random boolean networks. *Physical Review Letters*, 90(6):068702, Feb. 2003.

[91] R. Sole and B. Goodwin. *Signs Of Life How Complexity Pervades Biology.* Basic Books, New York, Jan. 2002.

[92] R. V. Sole, B. Luque, and S. Kauffman. Phase transition in random networks with multiple states. *arXiv:adap-org/9907011*, July 1999.

[93] R. V. Sol. *Phase Transitions.* Princeton University Press, Princeton, N.J, Aug. 2011.

[94] H. E. Stanley. *Introduction to Phase Transitions and Critical Phenomena.* Oxford University Press, New York, July 1987.

[95] S. H. Strogatz. Exploring complex networks. *Nature*, 410(6825):268, Mar. 2001.

[96] S. H. Strogatz. *Nonlinear Dynamics and Chaos: with Applications to Physics, Biology, Chemistry, and Engineering.* Westview Press, Cambridge, MA, 1 edition edition, Jan. 2001.

[97] G. Tenenbaum. *Introduction to Analytic and Probabilistic Number Theory.* Cambridge University Press, 1995.

[98] R. J. Trudeau. *Introduction to Graph Theory.* Dover Publications, New York, 2nd edition edition, Feb. 1994.

[99] V. Uritsky, M. Paczuski, J. Davila, and S. Jones. Coexistence of self-organized criticality and intermittent turbulence in the solar corona. *Physical Review Letters*, 99(2):025001, July 2007.

[100] T. Vicsek, A. Czirk, E. Ben-Jacob, I. Cohen, and O. Shochet. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 75(6):1226–1229, Aug. 1995.

[101] C. H. Waddington. Canalization of developement and the inheritance of acquired characters. *Nature*, 150:563–565, Nov. 1942.

[102] A. Wagner. *Robustness and Evolvability in Living Systems:*. Princeton University Press, 1 edition edition, July 2007.

[103] C. C. Walker and W. R. Ashby. On temporal characteristics of behavior in certain complex systems. *Kybernetik*, 3(2):100–108, May 1966.

[104] L. P. Wang, E. E. Pichler, and J. Ross. Oscillations and chaos in neural networks: an exactly solvable model. *Proceedings of the National Academy of Sciences of the United States of America*, 87(23):9467–9471, Dec. 1990.

[105] R.-S. Wang, A. Saadatpour, and R. Albert. Boolean modeling in systems biology: an overview of methodology and applications. *Physical Biology*, 9(5):055001, Oct. 2012.

[106] Wikipedia. Ternary plot, Aug. 2014. Page Version ID: 614445271.

[107] D. M. Wittmann, C. Marr, and F. J. Theis. Biologically meaningful update rules increase the critical connectivity of generalized Kauffman networks. *Journal of Theoretical Biology*, 266(3):436–448, Oct. 2010.

[108] J. M. Yeomans. *Statistical Mechanics of Phase Transitions*. Clarendon Press, May 1992.

[109] J. G. T. Zañudo, M. Aldana, and G. Martínez-Mekler. Boolean threshold networks: Virtues and limitations for biological modeling. In S. Niiranen and A. Ribeiro, editors, *Information Processing and Biological Systems*, number 11 in Intelligent Systems Reference Library, pages 113–151. Springer Berlin Heidelberg, Jan. 2011.

[110] Y.-C. Zhang. Modeling market mechanism with evolutionary games. *Europhys. News*, 29:51, 1998.