TUM School of Natural Sciences
Technische Universität München

Computational Models for Cellular Pattern Formation with Local Interactions

A Cellular Automata Approach

Stephan Michael Kremser

Vollständiger Abdruck der von der TUM School of Natural Sciences der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz:

Prof. Dr. Friedrich C. Simmel

Prüfer*innen der Dissertation:

1. Prof. Dr. Ulrich Gerland

2. Prof. Dr. Martin Zacharias

Die Dissertation wurde am 12.04.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Natural Sciences am 14.06.2023 angenommen.

Abstract

The formation and maintenance of correct patterns of different cell types is essential in developmental and synthetic biological system. While there are experimentally and theoretically well established models describing pattern formation with long-range interactions, the theoretical description of pattern formation with short-range interaction lags behind. For these interaction types, detailed models as well as conceptual ideas like the lateral inhibition model are available for specific systems. However, more general conceptual frameworks appear not to be widely developed for or adapted to pattern formation with biological cells.

This thesis aims to contribute to such general conceptual frameworks by studying a minimal model which is obtained by a threefold discretization. Spatial discretization is achieved by using the biological cells itself, usually placed on a grid, as basic units. The internal cellular details are aggregated into a small finite amount of discrete "states", effectively corresponding to fixed points of a continuous dynamical systems description. Since changes between such states take a finite amount of time, additionally a discretization in time can be employed. The natural mathematical model with such a threefold discretization is a so called cellular automaton (CA). The spatio-temporal dynamics in a CA is described by an update function called "rule". CA will be used in different variants throughout this thesis: starting with a basic CA, and then branching out from this basic version to include additional motives found in biological system in a one by one fashion - always removing the previous addition before adding the next element as central theme for a chapter.

As a first variant of this model class, the formation of axial patterns, more specifically the iconic French Flag, is studied. Using a basic 3 and 4 state CA, evolutionary algorithms, consensus procedures, and engineering approaches are employed to find dynamical rules which produce said French Flag. For the 3 state space, a single type of rule is found which solves the task by partially rewriting and partially sorting random initial patterns. These two patterning approaches, rewriting and sorting, can also be used individually to create axial patterns in 4 state space. They show different properties when characterized with regard to robustness to noise and asynchrony, kinetics, and scaling, and can be modified to handle different initial conditions and target axial patterns.

After having identified rules for which form a specific target pattern without a specific initial condition or external inputs, subsequently rules which allow the pattern to be flexibly programmable through external inputs are investigated. Such inputs could come from organizer cells or optical stimuli. By mapping the problem to a network, a limited set of rules which allow for the maximum flexibility of creating any target pattern upon appropriate external input are found. The ten representative members of this set of rules all follow an information conservation criterion, which can be shown to be sufficient for maximum flexibility, but have different kinetics of pattern formation. Extending the problem to several dimensions allows for robustness against noise, and combining several dynamical rules improves the robustness against the timing of the external inputs.

CA usually assume a synchronous update, which is also found in some biological systems like in vertebrate somite development. However, synchrony may not always be a given. To study conditions under which the assumption of synchrony is valid, noise is included in the time point of the update together with a signaling delay as a buffer against it. With this model it is investigated how active synchronization algorithms can conserve an effectively synchronous update. Results include that in general with more information available to them, synchronization algorithms are more successful in conserving synchronization, although that relationship is not strictly monotonic. The algorithms are also applicable in several dimensions and regular as well as irregular interaction topologies, and can deal with quenched and annealed noise. Coupling pattern formation to synchronization shows that dynamical rules which quickly reach a steady state are more robust to asynchrony. Additionally coupling the synchronization to the pattern formation, an additional class of rules which frequently change the current pattern appear as robust.

As a next modeling approach, short range interactions are combined with the classical morphogen gradient constructing axial patterns. It is shown that the local interactions can help to smooth boundaries between the different homogeneous fields of cell types in a noisy environment. Particularly successfully is the direct addition of the gradient and the short range signaling input, which even in the presence of comparably strong noise in the local signaling can still improve the boundary roughness, all while preserving the general scaling properties of the morphogen gradient pattern formation.

Biological cells are inherently mechanical objects which exert forces on each other. Furthermore, they grow, divide, and die. To include those aspects, the CA dynamics is coupled to a vertex model which effectively changes the underlying grid the CA cells sit on. It is shown that even in such a potentially dynamic and varying environment, “Salt-and-Pepper” patterns can still be formed successfully.

To take the minimal CA models closer to integrating experimental data directly and bridge the gap to more detailed but also more specific models of pattern formation with local interactions, a general class of probabilistic CA mimicking simple verbal models is trained to emulate experimentally observed pattern formation processes in the small intestine epithelium of mammals. Comparison to available experimental test data shows that CA models have the potential to provide a platform to mathematically represent verbal biological models and check if the proposed verbal mechanisms match the available experimental data.

Finally, an outlook on how CA models can be generalized with the use of network theory is given and it is discussed how the recently available large scale “-omics” data-sets can be incorporated in the modeling process.

Zusammenfassung

Die Bildung und Erhaltung von korrekten Zelltypmustern ist wesentlich für Systeme in der Entwicklungsbiologie und der synthetischen Biologie. Während es für die Musterbildung mit langreichweitigen Interaktionen sowohl theoretisch als auch experimentell etablierte Modelle gibt, hinkt die theoretische Beschreibung für Systeme mit kurzreichweitigen Interaktionen hinterher. Für diese Art der Interaktion sind in erster Linie für spezifische Systeme detaillierte Modelle und konzeptionelle Ideen vorhanden. Allgemeine übergreifende Konzepte scheinen jedoch bis jetzt nicht entwickelt oder nicht an die Musterbildung mit biologischen System angepasst worden zu sein.

Diese Dissertation zielt darauf ab zu einem derartigen allgemeinen, konzeptionellen Rahmen beizutragen indem, ein minimales Model untersucht wird, das durch eine dreifache Diskretisierung konstruiert wird. Räumliche Diskretisierung wird dadurch erreicht, dass die biologischen Zellen selbst als grundlegende Einheiten des Modells, üblicherweise auf einem Gitter verteilt, benutzt werden. Die intrazellulären Details werden in eine endliche Anzahl von diskreten Zuständen aggregiert. Diese Zustände entsprechen effektiv den Fixpunkten einer Beschreibung durch nichtlineare, kontinuierliche Systeme. Da die Wechsel zwischen solchen Zellzuständen eine endliche Zeit benötigen, kann zusätzlich eine Diskretisierung in der Zeit genutzt werden. Das natürliche mathematische Modell, welchem eine derartige dreifache Diskretisierung inhärent ist, ist der zelluläre Automat (CA). Die raum-zeitliche Dynamik eines CA wird durch eine Aktualisierungsfunktion, die „Regel“ genannt wird, beschrieben. CA werden im Rahmen dieser Thesis in unterschiedlichen Variationen benutzt: Ausgehend von einem grundlegenden CA zweigen unterschiedliche Modellvarianten ab, die zusätzliche Motive beinhalten, die in der Entwicklungsbiologie gefunden werden. Dabei wird üblicherweise Stück für Stück vorgegangen, indem zuerst das zuvor hinzugefügte Motiv wieder entfernt wird, bevor das nächste Motiv als zentrales Element eines Kapitels dieser Arbeit hinzugefügt wird.

Als erste Variante dieser Modelklasse wird die Bildung von axialen Mustern, genauer die Bildung der ikonischen „French Flag“, betrachtet. Unter Verwendung von CA mit 3 und 4 Zuständen werden evolutionäre Algorithmen, Konsensusprozeduren und konstruktive Ansätze benutzt um dynamische Regeln zu finden, welche diese „French Flag“ produzieren. Im 3-Zustandsraum wurde eine einzige Art von Regel, die das Problem durch teilweises Überschreiben und teilweises Sortieren des ursprünglichen Musters löst, gefunden. Die zwei Herangehensweisen – "Überschreiben" und "Sortieren" – können im 4-Zustandsraum auch jede für sich genutzt werden, um axiale Muster zu erzeugen. Sie haben dabei unterschiedliche Eigenschaften im Bezug auf Robustheit gegenüber Rauschen und Asynchronität, Kinetik und Skalierung, und können modifiziert werden, um mit unterschiedliche Anfangs- oder Zielbedingungen umgehen zu können.

Nach der Indentifikation von Regeln, welche ein spezifisches Zielmuster bilden ohne von einem spezifischen Anfangsmuster auszugehen, wurden im Folgenden nach Regeln gesucht, die eine flexible Programmierbarkeit des Zielmusters eines Zellfeldes durch externe Eingaben für wenige Zellen des Feldes erlauben. Derartige Eingaben können von einer „Organizer“ Zelle oder durch externe optische Stimuli erzeugt werden. Indem das Problem auf ein Netzwerk abgebildet wird, konnte eine beschränkte Menge von Regeln identifiziert werden, die, gegeben passende externe Eingaben, die maximale Flexibilität der Erzeugung jedes gewünschten Zielmusters erlauben. Die 10 repräsentativen Mitglieder dieses Regelsets gehorchen alle einem Informationserhaltungskriterium, von dem gezeigt werden kann, dass es hinreichend für maximale Flexibilität ist, aber zeigen im Allgemeinen eine unterschiedliche Musterbildungskinetik. Eine Erweiterung des Problems auf mehrere Dimensionen erlaubt die Erhöhung der Robustheit gegenüber Rauschen, und die Kombination mehrerer unterschiedlicher dynamischer Regeln verbessert die Robustheit bezüglich der zeitlichen Koordinierung der externen Eingaben.

Für CA wird üblicherweise ein synchrones Update angenommen, welches sich auch in biologischen Systemen wie der Somitogenese in Wirbeltieren findet. Diese Synchronizität mag jedoch nicht in allen Kontexten voraussetzbar sein. Um zu untersuchen, unter welchen Bedingungen die Annahme der Syn-

chronizität valide ist, wird ein zeitliches Rauschen zusammen mit einer Verzögerung als Puffer gegen das Rauschen in das Model integriert. In diesem Rahmen wird untersucht, wie aktive Synchronisationsalgorithmen ein effektiv synchrones Update erhalten können. Es wird gezeigt, dass Synchronisationsalgorithmen besser funktionieren je mehr Informationen sie verarbeiten können, auch wenn dieser Zusammenhang nicht monoton ist. Diese Algorithmen können auch in mehreren Dimensionen und sowohl auf regulären als auch auf irregulären Gittertopologien angewendet werden, und sie können sowohl mit „quenched“ als auch mit „annealed“ Rauschen umgehen. Die Kopplung der Musterbildung an die Synchronisation zeigt, dass dynamische Regeln, welche rasch einen Gleichgewichtszustand erreichen, robust gegenüber Asynchronität sind. Bei einer zusätzlichen Kopplung von Synchronisation an die Musterbildung zeigt sich, dass dann auch die Klasse der Regeln, welche häufig das aktuelle Muster wechselt, relativ robust gegenüber Asynchronität ist.

Als weitere Modellvariation werden die kurzreichweitigen Interaktionen mit klassischen Morphogengradienten zur Konstruktion axialer Muster kombiniert. Es wird gezeigt, dass lokale Interaktionen dabei helfen können glatte Grenzen zwischen homogenen Zelltypfeldern in der Präsenz von Rauschen zu erzeugen. Insbesondere ist die direkte Summe von kurz- und langreichweitigen Signalen erfolgreich, welche sogar im Falle von relativ starkem Rauschen des lokalen im Vergleich zum globalen Signals die Grenzrauigkeit verbessert, während die allgemeinen Skalierungseigenschaften des Morphogengradienten erhalten bleiben.

Biologische Zellen sind inhärent mechanische Objekte, die Kräfte auf einander ausüben. Außerdem wachsen sie, teilen sich und sterben. Um diese Aspekte in ein Model zu integrieren, wird die CA Dynamik in ein Vertexmodel integriert, welches das dem CA zugrundliegende Gitter, auf dem die Zellen platziert sind, modifiziert. Es wird gezeigt, dass selbst in einer solch potentiell dynamischen und wechselhaften Umgebung „Salt and Pepper“ Muster erfolgreich erzeugt werden können.

Um die minimalen Modelle näher an die Integration von experimentellen Daten heranbringen zu können und ein Brücke zu detaillierten aber spezifischeren Musterbildungsmodellen mit lokalen Wechselwirkungen zu schlagen, wird eine allgemeine Klasse von probabilistischen CA, welche einfache verbale Modelle eines biologischen Prozesses abbildet, trainiert, einen beobachteten Musterbildungsprozess im Dünndarm von Säugetieren nachzubilden. Ein Vergleich mit experimentellen Daten zeigt, dass CA Modelle das Potential haben eine Plattform zu bieten, um verbale biologische Modelle, welche aus experimentellen Daten erzeugt werden, mathematisch abzubilden und zu überprüfen, ob die verbal vorgeschlagenen Mechanismen zu den verfügbaren experimentellen Daten passen.

Schlussendlich wird ein Ausblick darauf gegeben, wie CA Modelle mithilfe von Netzwerktheorie verallgemeinert werden können und es wird diskutiert wie die seit kurzem vorhandenen „-omics“ Datensätze in die Modellbildungsprozesse eingebunden werden können.

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

One of the most intriguing processes in nature is the development of organisms from a single cell, the fertilized egg, to complex, multicellular beings, consisting of 10^{13} cells in the case of humans [1]. This process is studied in developmental biology [2, 3] and involves, amongst other processes, the specification of cell types at correct positions relative to each other. While a correct formation of cell type patterns leads to functioning organisms, failures in the pattern formation process can lead to malformations or death. As an example, perturbations in the Notch signaling pathway, one of the major molecular means of communication between cells, can lead to an incorrect formation of arteries, wrong spacing of sensory organs, or malformation of pigmentation stripes [4]. Thus, the study of how patterns form is central to understanding successful and failed development.

The pattern formation process itself is characterized by a self organization of the cells into these patterns, sometimes augmented by external signals. It involves changes of cell fate, cell division, death and migration steered by mechanical, chemical and electrical interaction between cells. They are studied experimentally using a small set of so-called "model organisms". These exemplary species are often convenient to work with experimentally [5, 6, 7], have an interesting phylogenetic position [5, 7], and, finally, are simply investigated by many [6, 8]. Examples include the worm *Caenorhabditis elegans*, the fruitfly *Drosophila melanogster*, the zebrafish *Danio rerio*, the African clawed frog *Xenopus laevis*, the mouse *Mus musculus*, and a few hand full of other species. From a phylogenetic coverage perspective this leads to the situation where a very few spots in the phylogenetic tree out of the 1.2 million known species and predicted 8.7 million eukaryotic species [9] are covered, as well as "local perturbations" around them. Although it might thus seem difficult to form general statements about pattern formation processes in developmental biology by studying only a small number of selected examples, it is assumed that only a comparably small number of different signaling pathways exist, which are conserved across many species [10], and thus the found principles can be hoped to be more widely applicable. Additionally, different species are remarkably similar in early development, exemplified by the close resemblance of the embryos of mammals to each other [11].

What are the pattern formation processes found in developmental biology? An example in a model species is the specification of sensory organ precursors on the *drosophila* thorax, which involves a prepattern, which gets further refined by local cell-cell interactions [12]. Another example is the segmentation in vertebrae, which usually involves locally coupled oscillators in cells whose phases get slowed and arrested by moving wavefront converting the temporal oscillations into a spatial pattern [13]. Yet, despite the (partially) available knowledge about some specific examples, "How do cells know where they are" is a simple question without a simple answer [14].

Understanding why a developmental mechanism works a certain way and not another, not only requires a detailed molecular knowledge about the specific developmental example, but also a complete picture of what can be achieved with the same or similar molecular ingredients in other contexts. Synthetic biology provides another approach to investigate pattern formation by recombining or recreating the molecular ingredients. In contrast to developmental biology, where usually the focus is more on knocking out components of a system in order to determine their necessity, synthetic approaches determine the sufficiency of components by a bottom-up approach [15]. Beyond sufficiency tests, creating intelligent, soft materials consisting of synthetic cells which are able to be programmed to form desired spatio-temporal patterns with the aim of creating new technologies is a goal major goal in synthetic biology [16].

From a theoretical perspective, there has been a long standing interest in extracting general patterning principles from experimental observations. A concept that received ample attention was that of the morphogen [17]. Coined as term by Turing, morphogens are chemical substances diffusing through a tissue and performing chemical reactions [18]. In Turing's work, two of these morphogens, a long range inhibitor

and a short range activator, act together to form stationary pattern of varying concentration of with finite wavelengths. Wolpert introduced in his solution to his famous French Flag problem of forming an axial pattern in the form of the eponymous Flag a spatial dependence of the morphogen concentration, usually called morphogen gradient, that is read out by the cells to instruct them about their position [19, 20]. Although in his seminal 1969 paper Lewis Wolpert complained that "A feature of developmental processes which is not often discussed is the extent to which there are, or will emerge, general or universal principles which are applicable to development in the same way that there appears to be universal rules for genetics, or, of more relevance, for the transcription and translation of genetic material at the molecular level" [20], the time has changed since "[p]attern formation [was] a rather neglected area of developmental biology" [20]. Both above proposed mechanisms of forming a pattern have since found solid experimental underpinning [21, 22, 23, 24, 25, 26].

Apart from the long ranging morphogens, cells can also communicate with signals of shorter range. The probably most well known example for this type of communication is through the interaction of the transmembrane receptor Notch and its counterpart ligands Delta and Jag, whose binding to the Notch receptor leads to a cleavage of the intracellular domain of Notch which subsequently modulates the expression of target genes [27]. Depending on the system and target genes, this may lead to a inhibition or induction [28], or even synchronized oscillations [29]. Due to its transmembrane nature, Delta-Notch or Jag-Notch signaling relies on direct cell-cell contact and is thus strictly local. With synthetic approaches, Notch can also be modified to target different genes and bind with different ligands, even diffusible ones [30, 31, 32]. While being the probably most well known mean of short range communication between cells, also Eph/Ephrin [33] as well as membrane tethered Wingless [34] in the class of membrane bound molecules, or Gap Junctions [35], small channels between the membranes of adjacent cells which allow the direct passage of molecules, have been studied. While there have been specific, mostly detailed models developed for the different means of local communication, a general study of pattern formation with local interactions in a unified framework seems to be missing. The benefit of such a general study could be the discovery of general patterning strategies as well as limits of the described patterning processes. For synthetic systems, these could be used as design instructions and bounds.

In order to conduct such a general study, using a simple coarse grained model has the advantage of potentially not requiring detailed prior information about the properties of the individual model components at the expense of relying on prior intuition about the cellular phenomenon [36]. Simple models have been shown to be able to reproduce complex dynamics and showing independence of many detailed model parameters [37]. Furthermore, since if a phenomenon of interest is recapitulated in a quantitative model, this indicates that all necessary players and interactions may have been identified, or, in the case of a negative results, that the phenomenon is not understood at the desired level yet [36]. Thus, the model class of top down, agent based models can be seen as a way to formalize and generate hypothesis about the understand such processes [38]. To keep it as simple as possible, the modelling framework chosen throughout this thesis are cellular automata (CA). Developed by Ulam, von Neumann, Conway and Wolfram [39, 40, 41, 42, 43, 44], they naturally encompass local interactions in a discretized space with a discrete set of states and a discrete time dynamics, with the latter two assumptions being motivated by the usual mapping of many molecular states to few cell fates [45] between which transitions than usually happen on larger time scales. In addition to a basic version of this model, variants which systematically include other pattern formation motives like organizer cells, long range signaling through morphogens, mechanical interactions, cell movement, division and death will be investigated to study the consequences of the combination of different motives.

This thesis is organized as follows: After an introduction to the molecular foundations of local, short range interactions in cellularized systems (chapter 2), CA are introduced as model systems with a discussion of the thereby made modelling assumptions (chapter 3). As a first goal, CA rules which form a specific pattern are investigated (chapter 4), followed by introducing organizer cells and searching for CA rules which are able to form any pattern upon appropriate organizer input (chapter 5). Then the effect of (a)synchrony is studied chapter 6. The effects local interactions can have in smoothening boundaries created by a global gradient are the focus of a equilibrium and a non-equilibrium model in chapter 7. CA

are combined with vertex models to include mechanical interactions, cell migration, division and death in chapter 8. How CA models can be used to approximate experimental data is explored with a probabilistic CA model for the case of small intestine tissue homeostasis in chapter 9. Finally, the found results are summarized and future avenues of research discussed (chapter 10).

2 How Do Cells Interact Locally? - Molecular Components of Short Range Interactions

2.1 Local Interactions

While for the manner of exchanging signals between individual cells there would in principle be a myriad of different possible signal carriers and ways of processing these signal carriers to influence gene expression levels, only a small number of conserved signaling pathways are found in nature [10]. For short range interactions, the probably most commonly discussed pathway is the Notch signaling pathway which is thus the main focus of this section. Other means of exchanging signals between close by cells have been experimentally studied and will be summarized briefly as well. The concept of local interaction on spatially discrete systems can also be found in purely synthetic, non cellularized systems. For a brief description of such a system see appendix A.1. Another synthetic system based on DNA nanorobotic arms [46] has been proposed by Simmel and coworkers [47, 48].

2.1.1 Notch Signaling Pathway

The Notch signaling pathway, usually assessed to be among the major developmental signaling pathways, is highly conserved among metazoans [58, 59, 60]. For a recent general reviews see [27] and [60].

Notch is a transmembrane protein, comprised of an extracellular part, a transmembrane part and an intracellular domain (NICD) (fig. 2.1 (a, b)). Briefly, the unidirectional signaling is initiated by a ligand binding to the extracellular domain of Notch, allowing a cleavage of NICD from the transmembrane part. The NICD translocates into the nucleus and activates transcription of target genes (fig. 2.1 (a, b)).

Canonical ligands of the usual variants Notch1-4 are Delta (Dll1,3,4)/Serate (fig. 2.1 (a)) and Jagged (Jag1,2), which are all transmembrane proteins [53]. Thus, direct cell-cell contacts are necessary to introduce signaling. Upon binding, the mechanical stretch of Notch makes its extracellular domain susceptible to cleavage through the ADAM[1]-metalloproteinase, allowing a subsequent cleavage of the NICD through a γ-secretase [4] (fig. 2.1 (c)). The NICD translocates into the nucleus, where it acts as transcription factor which modulates downstream targets by assembling a transcription activation complex together with the CSL[2] protein and co-activators like Mam[3], SKIP[4], and HATs[5] [4, 27] (fig. 2.1 (c)). Since no modification of NICD occurs during its translocation to the nucleus, the activation strength only depends on the nuclear concentration of NICD [60]. Activated target genes may include *Notch* itself, as well as *Jag* and *Hes*, which in turn represses itself, *Delta* and differentiation factors [27, 52, 58] (fig. 2.1 (c)). After expression, Notch may be glycolized by Fringe (increasing (decreasing) the sensitivity for Delta (Jagged) ligands) and can subsequently be cleaved by Furin-like convertase into intracellular and extracellular domains and transported to the plasma membrane [27, 60].

Delta from the same cell may also inhibit the Notch receptor by binding without introducing the necessary cleavages for signaling ("cis inhibition") [53] (fig. 2.1 (c)). In contrast to the usual trans activation with a ligand from a different, the response to cis inhibition is sharp and not graded [61].

Notch signaling strength depends on the contact area between the sender and receiver cell, with a larger contact area leading to a larger signal strength [4, 28, 62] (fig. 2.1 (d)). Since the signaling involves direct

[1] a disintegrin and metalloproteinase
[2] CFB1-Suppresor of Hairless-LAG1
[3] Mastermind protein
[4] Ski interacting protein
[5] histone acetyltransferases

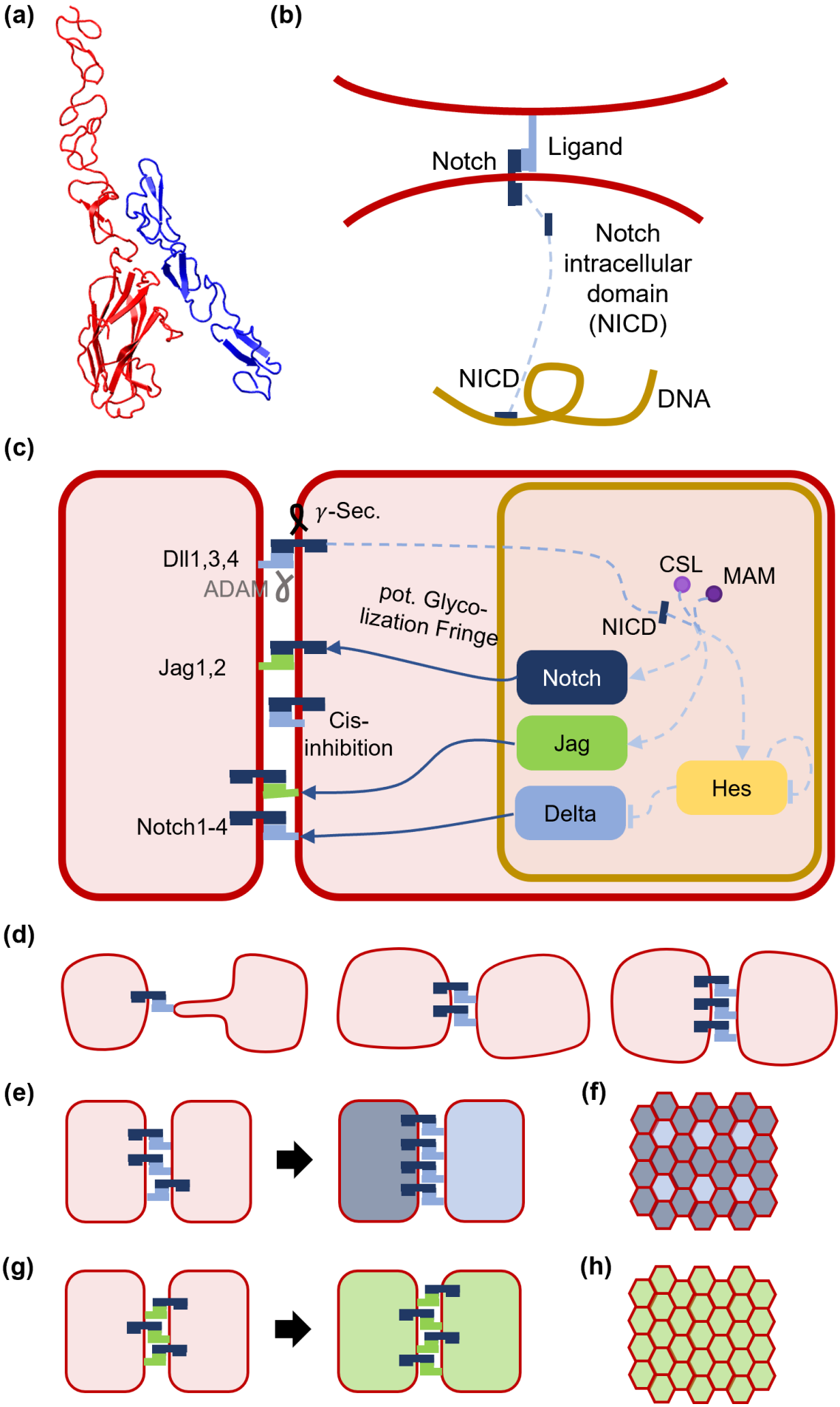

(a)
(b)
Notch
Ligand
Notch intracellular domain (NICD)
NICD
DNA
(c)
γ-Sec.
Dll1,3,4
ADAM
Jag1,2
Cis-inhibition
Notch1-4
pot. Glyco-lization Fringe
CSL
MAM
NICD
Notch
Jag
Delta
Hes
(d)
(e)
(f)
(g)
(h)

Figure 2.1 Delta Notch Signaling. (previous page) (a) Complex of NOTCH1 (EGF11-13) bound to DELTA-LIKE 4 (N-EGF2) [49, 50], created with VMD [51] (VMD was developed by the Theoretical and Computational Biophysics Group in the Beckman Institute for Advanced Science and Technology at the University of Illinois at Urbana-Champaign.). (b) Illustration of the principle idea of Notch signaling: The extracellular part of Notch binds to a ligand attached to a neighboring cell, initiating a cleavage of the Notch intracellular domain (NICD). The NICD translocates onto its target on the DNA and enables transcription of the downstream target genes. (c) A more detailed view, adapted from [27] and [52]: The possible canonical ligands Delta Dll1,3,4 and Jaged Jag1,2 bind to the Notch (Notch1-4) receptor [53], which, through mechanical pulling, exposes the extracellular part to cleavage through the ADAM-metalloproteinase followed by a subsequent cleavage of the NICD through a γ-secretase [4]. The NICD translocates into the nucleus where it assembles a transcription activation complex together with the CSL protein and co-activators like Mam, SKIP, and HATs [54], which modulates the expression of target genes. This target genes may include the Notch receptor Jag/Ser, Notch itself, and Hes, which in turn can inhibit Delta expression, differentiation factors and its own expression [27, 52]. Glycolization of Notch trough Fringe can modulate the Notch receptor to increase (decrease) its affinity to Delta (Jagged) [27]. Expressed Delta ligands of the same cell can inhibit Notch receptors (cis-inhibition), preventing binding of Delta from other cells [55]. (d) Notch signaling strength depends on the contact area between cells. The signaling can potentially not only reach directly neighboring cells, but via filopodia also cells further away [4, 56] (e) For initially similar cells, concentration fluctuations lead to a stable expression (repression) of Delta (Notch) in one and Notch (Delta) in the other cell [57]. (f) This leads to "Salt-and-Pepper" like patterns of cells with high/low Notch (low/high Delta) expression [28]. (g) Conversely, the Jag concentration dynamics has a fixed point for large Jag concentration in both neighboring cells [57], leading to a lateral induction (h) of Jag concentration.

contact between cells, a naive expectation would be a pattern wavelength of at most 2 cells when the interaction range is the length-scale determining the pattern range [28]. Yet, even in this case it is possible to achieve large wavelength signaling by using filopodia, i.e., protrusion of the cell which can reach cells further away [56] (fig. 2.1 (d)). As shown by Yaron et al. [63], juxtracine signaling, and thus Notch signaling, is inherently noisy regardless of receptor density or cell-cell contact area, and for proper readout temporal integration of signal is necessary.

The pattern which is formed depends on the expressed ligand [27, 28]. A detailed modelling of the dynamics of the Notch pathway is usually achieved using the framework of differential equations, e.g., in [64]. Starting with a 2 cell model, an expression of Jag upon NICD activation has stable fixed points for no Jag expression in both cells, or a high Jagged expression in both cells [27, 57]. If one of the cells has a nonzero Jag concentration, expression of Jag is induced in the other cell ("lateral induction"). If Hes is expressed upon NICD activation, this leads in turn to a repression of Delta expression in the cell, leading to a low expression of Delta together with a large expression of Notch in one of the cells and a large expression of Delta and low expression of Notch in the other ("lateral inhibition") [27, 57] (fig. 2.1 (e, f)). On a larger scale, this causes a so-called "Salt-and-Pepper" expression of Notch and Delta. Depending on the exact time scales of the expression dynamics, anti-synchronized Delta-Hes oscillations may appear during neurogenesis, leading to a dynamic "Salt-and-Pepper" pattern [65]. Hes oscillations, coupled and synchronized between cells through Delta Notch interactions, are also the basis of the clock and wavefront mechanism of vertebrate somitogenesis [13].

The Notch signaling pathway is also able to discriminate between different Delta ligands by triggering either pulsatile (Dll1) or sustained (Dll4) activation dynamics, which in turn activates either Hes1 or Hey1/L genes, respectively [66, 67].

The Notch signaling pathway can be modified synthetically to change its function. Through genome engineering, an optogenetic allele of Delta can be integrated in the genome of *Drosophila melanogaster*, which can inhibit endogenous Delta activity upon activation [68]. By exchanging the extracellular and intracellular domain of Notch, i.e., the recognition domain of the ligand and the effector domain affecting the transcriptional output, the so-called "synNotch" pathway is able to affect differentiation, spatial patterning and boolean decissions in many mammalian cell types [30]. Using this approach, the possibility of affecting cell-cell adhesion and thus changing the mechanical properties of cells has been demonstrated [31]. The extracellular domain of the Notch receptor can further be modified to detect arbitrary diffusible, syn-

thetic morphogens [32]. Different synNotch pathways can be engineered to not share common signaling intermediates and are thus functionally orthogonal to each other [30].

2.1.2 Eph/Ephrin Singaling

Ephrins and their receptors Ephs are families of membrane bound molecules similar to Notch and its ligands [2]. They each divide into two classes, A and B, where binding between receptors and ligands of the same class but not the opposite class is possible[6][2]. In contrast to Notch, they are not only capable of forward signaling[7], but also reverse signaling[8] and bidirectional signaling [33]. Similar to Notch signaling, signals may also be exchanged between non nearest neighbor cells via protrusions [69], but also exchange of signals at a distance via exosomes is possible [70].

Eph/Ephrin signaling influences the mechanical properties of cells with their main signaling target being the actin cytoskeleton [2]. Their sole interaction can mediate cell-cell adhesion, and the effects of their signaling can induce cell-substrate adhesion, cell migration and axon extension [71]. However, the signaling may also induce repulsive effects. The effect of a particular Eph/Ephrin interaction is not straightforward, but depends on the cell types, the number of receptors and their clustering, the total intensity of the Eph signals, and crosstalk with other pathways [2, 71]. Eph/Ephrin interactions in neural cells have been found to guide cell migration over large distances and establish left-right asymmetries, as well as playing a role in tissue separation and boundary formation processes [33]. Eph/Ephrin signaling can also influence cell differentiation, e.g., in the small intestine epithelium it regulates stem cell renewal by controlling their position and thus exposure to proliferation factors [71, 72].

2.1.3 Gap Junctions

Gap junctions are small channels in the cell membranes of animals that connect two adjacent cells. They consist of 6 connexin molecules forming a half channel called connexon, which adheres to a connexon of an adjacent cell forming a gap junction bridging an approximately $2\text{-}3nm$ gap between the two cells [35, 73]. Connexons are able to transport ions, small molecules, hormones or messenger molecules like cAMP through their approximately $1.5nm$ channels, allowing for local signal relay between cells [73]. Signaling through gap junctions can be regulated rapidly by adjusting the membrane potential across the channel, as well as their probability of opening [35]. A slow regulation is possibly via changing connexin synthesis, assembly, modification, or degradation rates [35].

In the context of electrically excitable cells, gap junctions have been discovered as a possibility to allow for electrical transmission between cells, permitting, amongst others, the ability to synchronize cells regarding electrical and mechanical output [35]. Further, by sharing signaling molecules across cells, spatial gradients can be buffered locally. As an example from a developmental application, it has been shown that gap junctions are essential for correct color pattern formation in zebrafish, although the exact mechanism is unclear [74].

Channels and pumps in cell membranes can more generally be used to exchange ions with the environment, most famously used by neurons, but also non-neural tissue is capable of using bioelectricity and gap junctions for computational purposes, even on the tissue level [75].

2.1.4 Tethered Wingless

Wingless (Wnt) is a diffusible morphogen and ligand of the Wnt pathway, a key pathway in embryonic development, e.g., eponymously during development of the wing in *Drosophila* [2]. Through genome engineering Wnt was expressed in a membrane tethered form, which surprisingly produced, apart from a temporal delay, a similar pattern than diffusing Wnt [34].

[6] Exception to this rule is EphA4 binding both A and B ligands [2]
[7] Signal is transduced in cell with receptor
[8] Signal is transduced in cell with ligand

2.1.5 Synthetic Systems: Signaling Between Artificial Cells

Attempting to bottom up construct artificial tissue made of cells proposes unique challenges. To separate the inner chemical milieu of the cells from the external environment, a form of compartmentalization is needed [76]. The permeability of said compartments needs to be chosen carefully to allow external supply with chemical energy to allow the non equilibrium operation of the cells as well as passing of signaling molecules[9] [16]. Different approaches have been proposed based on phospholipid membranes [16], DNA [77], integrated proteins [78], peptides [79] and proteins [80], as well as polymeres [81], all with different mechanical and chemical properties. To allow the passage of signaling molecules, membrane pores like α hemolysin or transmembrane receptors can be integrated and facilitate information exchange with the environment [16]. The "cell type" of these closed compartments is then determined by the concentration of certain molecules. For open compartments, the "cell type" can be realized by immobilization or selective binding of reaction components [16].

The exchange of signaling molecules through membranes is dependent on the membrane and signaling molecule composition. Smaller, apolar molecules like 3OC6-HSL might pass directly through the membrane, larger, charged molecules like arabinose might need membrane pores [26]. Another possibility is the use of membrane free compartments which allow the direct exchange of proteins [82].

As an example, Dupin et al. have shown that in emulsion-based artificial cellular compartments with synthetic in vitro gene circuits artificial signalling and differentiation processes can be realized [26].

2.1.6 Intermediate Summary

To summarize so far: There are various options for exchanging signals locally. They can be membrane bound, diffusion or bio-elelectricity based. Depending on the concrete choice of signaling option, the communication might be an one or two way street. There may be crosstalk between different means of communication. The signaling can for example affect gene expression and cellular mechanics. The signaling pathways can be modified synthetically, or the whole "cellular" system can be created synthetically by constructing some form of computational circuits which are compartmentalized and stand in contact to each other by some mean of exchanging signals.

2.2 Other Developmental Motives Interacting With Local Signaling

While local, short range interactions between cells are an important factor in development and homeostasis of tissues, other developmental motives may act at the same time and thus the combined influence of short range interactions with said motives can open up completely new possibilities and challenges for pattern formation. In the following section, some of these motives are described and their interplay with patterning through local signaling discussed.

2.2.1 Organizer cells

Organizer cells are cells that do not receive or act on any signals they receive from other cells, i.e., are autonomous, but signal themselves to their surroundings. The most famous example of this is the Speemann-Mangold organizer [2, 83], a group of cells in amphibians that determines the fate of the cells around them show by grafting experiments. Another type of cell that acts as organizer cells are the anchor cells in *C. elegans* Vulva development steering the fate specification of the six vulva precourser cells in a row [28, 84]. Also other types of signaling centers are known in developmental biology like the Nieuwkoop center [2]. From a synthetic perspective, signaling centers could be engineered using optogenetic manipulation, e.g., of Delta-Notch signaling [68]. Having control over the signals cells send to their neighbors could have the

[9] Continuous production of signaling molecules requires energy and nutrient supply, but may either be emulated by a passive, large reservoir with low permeability or active production in cells [16]

advantage of inducing changes in patterns at cellular resolution while only addressing a single or a small number of cells with external signals.

2.2.2 Morphogen Gradients

Morphogen gradients, as described in chapter 1, are a solution to the problem of axial pattern formation by using varying concentrations of diffusible molecules read out by cells to determine the type of a cell. They can potentially interplay with short range interactions as shown in *Drosphila* development: In the early development, when the embryo is not yet cellularized, spatial pooling by diffusion of downstream effectors can increase the precision of spatial localization [14, 85]; In the wing imaginal disc, communication via the Fat signaling pathway generates precise positioning of a boundary in a sharp Wingless gradient [14, 86]. During the specification of sensory organ precursor cells in the bristle formation, local interactions act as a refinement of a gradient prepattern [12]. In the chick embryo during primitive streak formation, a combination of local and global signaling has recently been shown to be a good explanation of the available data [87].

In synthetic systems, SynNotch [30] is an example of an engineered signaling toolkit that can both encompass direct contact interactions [31] as well as interactions with diffusible morphogens [32], allowing in principle for a combination of both signals.

For a potential coupling of different signals into a gene expression output different possibilities exist [88, 89]. If, for simplification, the input function between activator concentration and promoter activity is approximated by a logic approximation[10], and both input signals (activators) are required to bind to the promotor for high expression, this can be modeled by an AND gate [89]

$$f = A\Theta\left(c_1 - K_1\right)\Theta\left(c_2 - K_2\right). \tag{2.1}$$

If binding of either is sufficient, this corresponds to an OR gate [89]

$$f = A\left[\Theta\left(c_1 - K_1\right) + \Theta\left(c_2 - K_2\right) - \Theta\left(c_1 - K_1\right)\Theta\left(c_2 - K_2\right)\right]. \tag{2.2}$$

If the inputs are additive, also a SUM representation [89, 90] may be chosen

$$f = A_1 c_1 + A_2 c_2. \tag{2.3}$$

2.2.3 Synchronization

Synchronization of chemical oscillations is an actively studied motive in developmental biology, see e.g., [13, 91]. The basis of chemical oscillations is usually a negative feedback with delay [89]. More complex motives are known, for example the Repressilator which involves an additional step [89, 92], or combinations of negative and positive feedback with the possible inclusion of a bi-stable behaviour for increased robustness [89].

When coupled, oscillators can synchronize their oscillation. Coupling between the oscillators can for example occour by sharing a common element in their feedback loops [93]. An example of coupled oscillations in developmental systems is the *zebrafish* segementation clock [13]: The expression of the Hes/Her gene experiences a negative feedback with delay, leading to an oscillation of the expression level; additionally, the expression is influenced by the NICD, while the Hes/Her protein can repress Delta expression; this allows for an influence of the Hes/Her gene expression of one cell on the Hes/Her gene expression of a neighboring cell via Delta/Notch interactions through direct cell-cell contacts.

Oscillations have also been shown in synthetic systems, e.g., of artificial oscillators in droplets [94], and synchronization has been studied with cell-free genetic oscillators in a spatially distributed system of on-chip DNA compartments via diffusive molecules [95].

[10] For an activator concentration c, the rate of production f of a target protein is the given by a step function $f = A\Theta\left(c - K\right)$ instead the usual model of a Hill function $f = A\frac{c^n}{c^n + K^n}$.

2.2.4 Cellular Mechanics, Cell Growth, Division and Death

From a physicists perspective, a simplified description of a cell might be of a soft sphere filled with a fluid. As such, it has a hydrostatic pressure associated with it, as well as pressure forces acting on it from the contact with other cells or external fluid. The membrane of the cell may stick to other cells, e.g., through the cadherine glykoproteins at the surfaces. The cell also has an internal mechanical structure consisting of a network of proteins linked together, the cytoskeleton. Cells can grow, regulated by the respective rates of biosynthesis of molecules relative to the degradation rates. Cell division, also a regulated process, can increase the number of cells in a tissue, as well as cell death, which in case of apoptosis is again highly regulated, can decrease the number of cells in a tissue.

All these motives can have an impact on patterning by local interaction. In the case of cell division and death, simply the existence or non existence of cells might influence the resulting pattern. The size of the cell can have an impact on the strength of the short range signals exchanged between cells [4, 28, 62]. Cell movement within the tissue might change the topology of the network of local interactions. Cadherine proteins mediating mechanical adhesion might also be involved in signal transmission [96, 97].

The other way round, the local interactions might also have an influence on the discussed motives. Notch signaling can influence tissue growth [98]. Eph/Ephrin can modify the mechanical properties of cells by targeting the actin cytoskeleton [2] as influence cell adhesion and migration [71] (see also section 2.1.2). The synNotch signaling platform can be programmed to influence cell adhesion [31].

2.2.5 Outlook on the Following Chapters

In the following chapter 3, a computational model for patterning with local interactions is presented. Then, in chapter 4, the simplest version of this model is used to investigate how pattern formation due to local interactions can lead to global patterns. After that, the model is supplemented one by one with the additional motives discussed in this section and the sole interplay of each of motive with local interactions is investigated - organizer cells (chapter 5), synchronization (chapter 6), morphogen gradients (chapter 7), and cellular mechanics, cell growth, division, and death (chapter 8).

3 How to Model Local Interactions? - Cellular Automata

In the previous chapter, the biological basis of pattern formation by local interactions have been presented. In this chapter, the main modelling tool used in this thesis to investigate these pattern formation processes is introduced: cellular automata (CA)[1]. While very simple in their structure, in their native formulation CA already rely only on local interactions to form patterns and are at the same time able to produce complex spatio-temporal dynamics, rendering it a great candidate for a model. The first section of the chapter discusses the possibly simplest and most famous version of CA, so called elementary CA (ECA). This framework gets generalized in the next section, followed by a discussion of possible classification quantities and examples of modeling biological and artificial systems with CA. In the final section, light is shed on the assumptions behind CA as modelling tool, as well as a discussion of their advantages and disadvantages.

3.1 Elementary Cellular Automata

CA are a general modeling tool developed by Ulam, von Neumann, Conway and Wolfram [39, 40, 41, 42, 43, 44]. For didactic purposes, first the simplest form of CA, so called elementary CA (ECA), are introduced. The model is then generalized and central definitions and results from cellular automata theory is summarized.

A elementary cellular automaton is a grid of cells holding a single value $x_i^t \in \{0,1\}$ each, where $i \in \{0,...,L-1\}$ specifies the position along the grid with size L and t specifying the time step (fig. 3.1 (a)). The values of the cells in the bulk of the grid are propagated in time according to a dynamical function, the rule,

$$x_i^{t+1} = f(x_{i-1}^t, x_i^t, x_{i+1}^t). \tag{3.1}$$

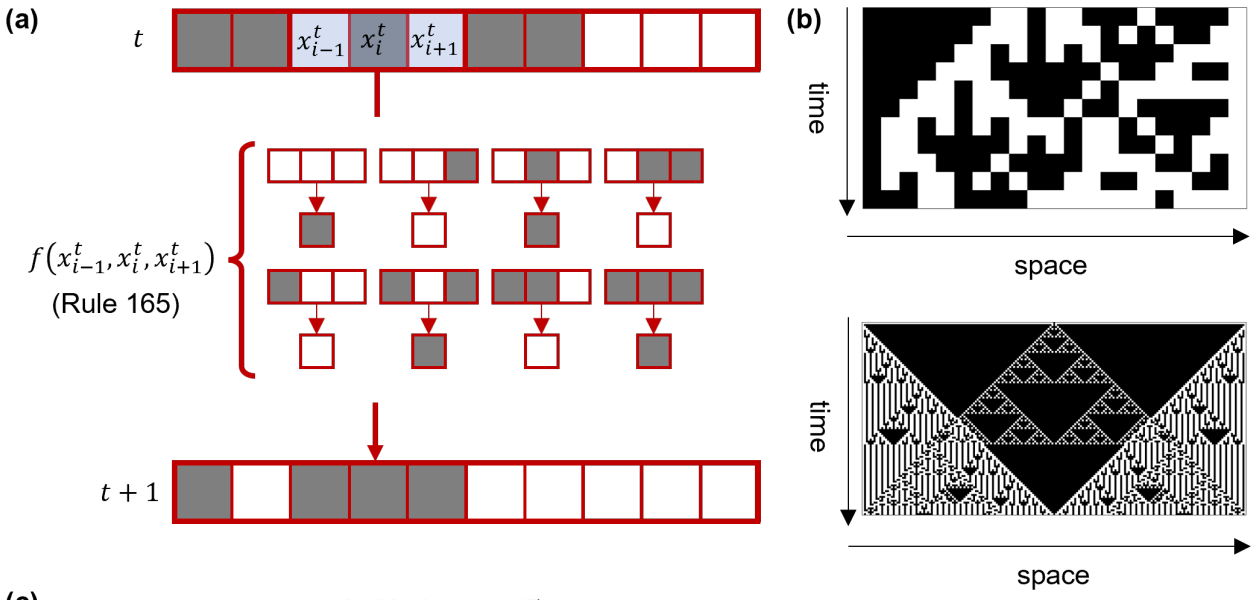
The rule is a input-output mapping specifying the next value of cell i given the current values of cell i as well as the current values of cell $i-1$ and $i+1$, the nearest neighbors. All cells of the grid are updated simultaneously according to the rule. The resulting dynamics over several steps in time is usually depicted by drawing the values of the grid for each time step underneath the previous one (fig. 3.1 (b)). Since this graphical representation is visually reminiscent of the so called kymographs used in other fields to depict space and time dependent variables, see, e.g., [99], this terminology will be used onward.

To be able to talk about rules in an efficient manner, they are enumerated by first assigning an order to the inputs of the rule by interpreting them as three digit binary numbers and second arrange the outputs into a single number by bringing them into a sequence according to their descending input number. This single output number is yet again interpreted as binary number unambiguously specifying the rule (cp. [42]). Converting to a decimal number yields the usually used identifier for the rule (fig. 3.1 (c)).

It is possible to reformulate each ECA rule in an algebraic or boolean [100] form, e.g., rule 165 can be rewritten as $x_i^{t+1} = \neg\left(x_{i-1}^t \oplus x_{i+1}^t\right)$ or as $x_i^{t+1} = \left(1 + x_{i-1}^t + x_{i+1}^t\right) \mod 2$ [101].

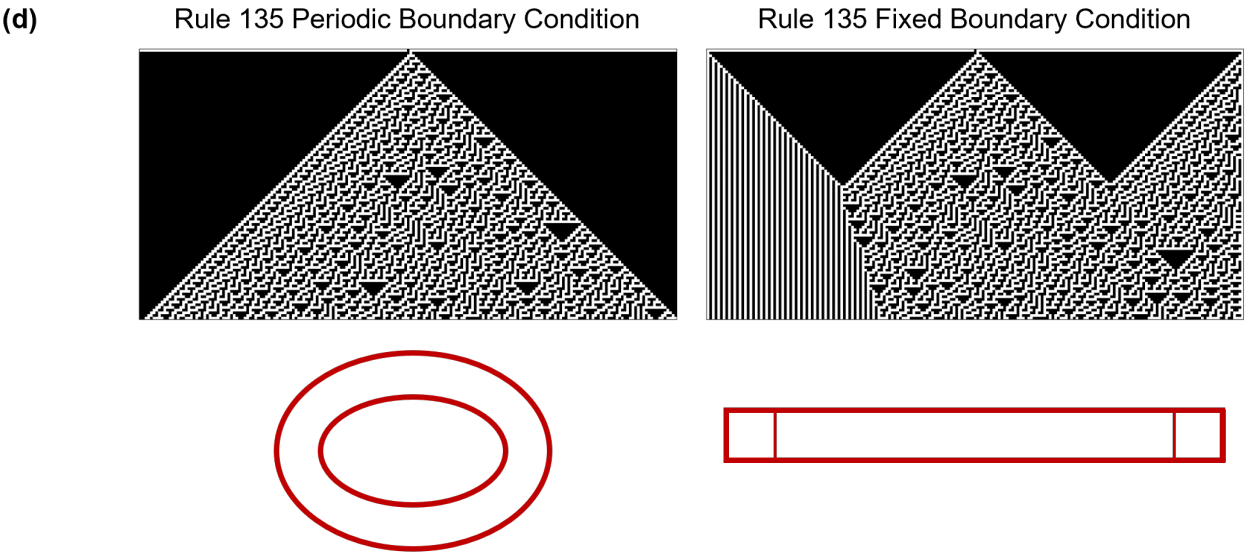
As infinite systems cannot be simulated practically, boundary conditions have to be chosen to fully specify the simulation. The usual types of choices are either fixed or periodic boundary conditions (fig. 3.1 (d)). Fixed boundary conditions can be used when they are known for the system to be modeled or when the influence of a certain choice of boundary values on the dynamics is of interest. Periodic boundary conditions have a use-case if the problem at hand is cyclic, e.g., in tube like cellular structures, which appear in

[1] "CA" will be used as an abbreviation both for the singular and plural form throughout the thesis



(c)

$$10100101_2 = 165_{10}$$



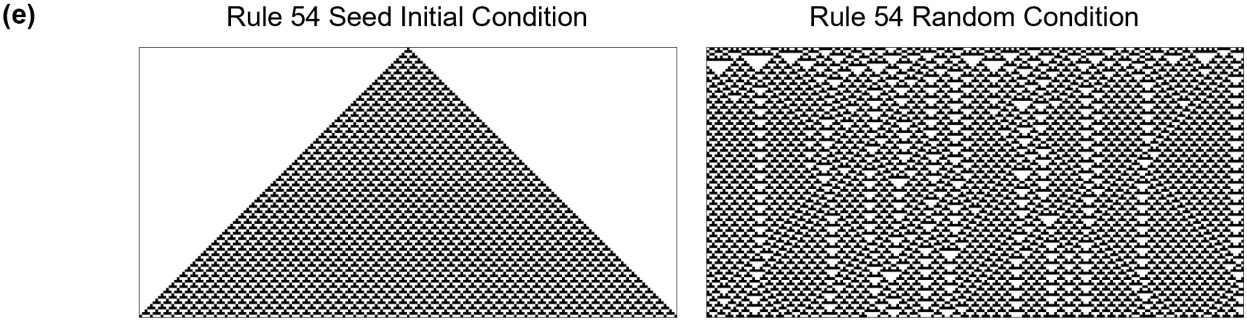


Figure 3.1 Elementary Cellular Automata. (a). A cellular automaton consists of a grid of cells holding a variable each which can take only the values 0 (depicted by white) or 1 (black). The system is propagated in time by a so called rule, a function which specifies which value a cell takes in the next time step given the value of the cell and its immediate two neighbors. All cells are updated simultaneously according to the same rule. (b) Simulation of rule 165 from random and seed initial condition, presented in a space-time graph, a so called kymograph. (c) Rules are enumerated by ordering their outputs according to the magnitude of the inputs in binary (e.g., the input $010 \hat{=} 010_2 = 2_{10}$ contributes to third entry from behind in the binary rule string), and then interpreting this number as a binary and converting it to decimal, here $10100101_2 = 165_{10}$. (d) Illustration of periodic and fixed boundary conditions for rule 135 and as geometrical scetch. While in the periodic boundary conditions the behaviour introduced by the seed is not influenced by interactions "across" the boundary, the fixed boundary influences the kymograph. (e) To start the dynamics, initial conditions have to be chosen. Here, the examples "seed" (homogeneous with a single differing site) and random initial condition are exemplified.

developmental system in the notochord for example [102], or when being interested in the "bulk" behaviour, with the hope that system is large enough that influences "across" the "boundary" do not influence the bulk behaviour on the relevant timescales. The latter is illustrated in fig. 3.1 (d), where the left kymograph with periodic boundary conditions could also be achieved by simulating a larger system with the same initial condition (homogeneous white with a single black in the middle) with fixed boundaries that are a distance larger than the spatial extension of the shown system from the current boundaries of the system away.

To start the dynamics an initial value for each cell needs to be specified. Typical initial conditions are the homogeneous initial condition (each cell has the same value), the "seed" initial condition (homogeneous except for one differing cell), or the random initial condition where usually among all possible 2^L initial conditions a sample is drawn with equal probability (fig. 3.1 (e)).

Exemplary kymographs for all 256 ECA rules for both seed and random initial conditions and periodic as well as fixed boundary conditions can be found in appendix A.2.1.

3.2 General Cellular Automata

Abstracting from the initial example of an ECA, a more formal definition based on [103] of CA can be formulated for the purpose here as:

Definition 3.2.1. A cellular automaton $\mathcal{C}$ is a tuple $\mathcal{C} = (S, \mathcal{T}, \mathcal{N}, f, I_0, B, U)$ where

(i) S is a set of k states, $S = \{0, ..., k-1\}$

(ii) $\mathcal{T}$ is a tessellation of a subspace D of the Euclidian space $\mathbb{R}^n$, $D \subset \mathbb{R}^n$, consisting of active sites $c_i \in \mathcal{T}_A$ enumerated by $i \in \{0, ..., L\}$ and boundary sites $b_j \in \mathcal{T}_B$ enumerated by $j \in \{0, ..., T_B\}$ with $\mathcal{T} = \mathcal{T}_A \cup \mathcal{T}_B$

(iii) $\mathcal{N}$ is a neighborhood function $\mathcal{N} : \mathcal{T} \longrightarrow \bigcup_{i=0}^{L} \mathcal{T}$ that maps every site c_i to a finite sequence $\mathcal{N}(c_i)$ consisting of distinct cells.

(iv) f is the update function (local update rule) $f : S^{\max_i |N(c_i)|} \to S$ governing the update dynamics of each site c_i,

$$x(c_i, t+1) = f\left((x(c_i, t))_{j=1}^{|N(c_i)|}, t \right) \tag{3.2}$$

(v) I_0 is the initial condition $I_0 : \mathcal{T} \to S^T$, assigning every active site c_i an initial value $x(c_i, t)$

(vi) B is the boundary condition $\mathcal{T}_B \to \mathcal{T}_B^T$, giving a value to each boundary value $x(b_j, t)$

(vii) $U(t)$ is an update schedule $\mathbb{N}_0 \to \{0, ..., L\}^L$, specifying which cell updates at a certain time step t.

Some notes on this definition:

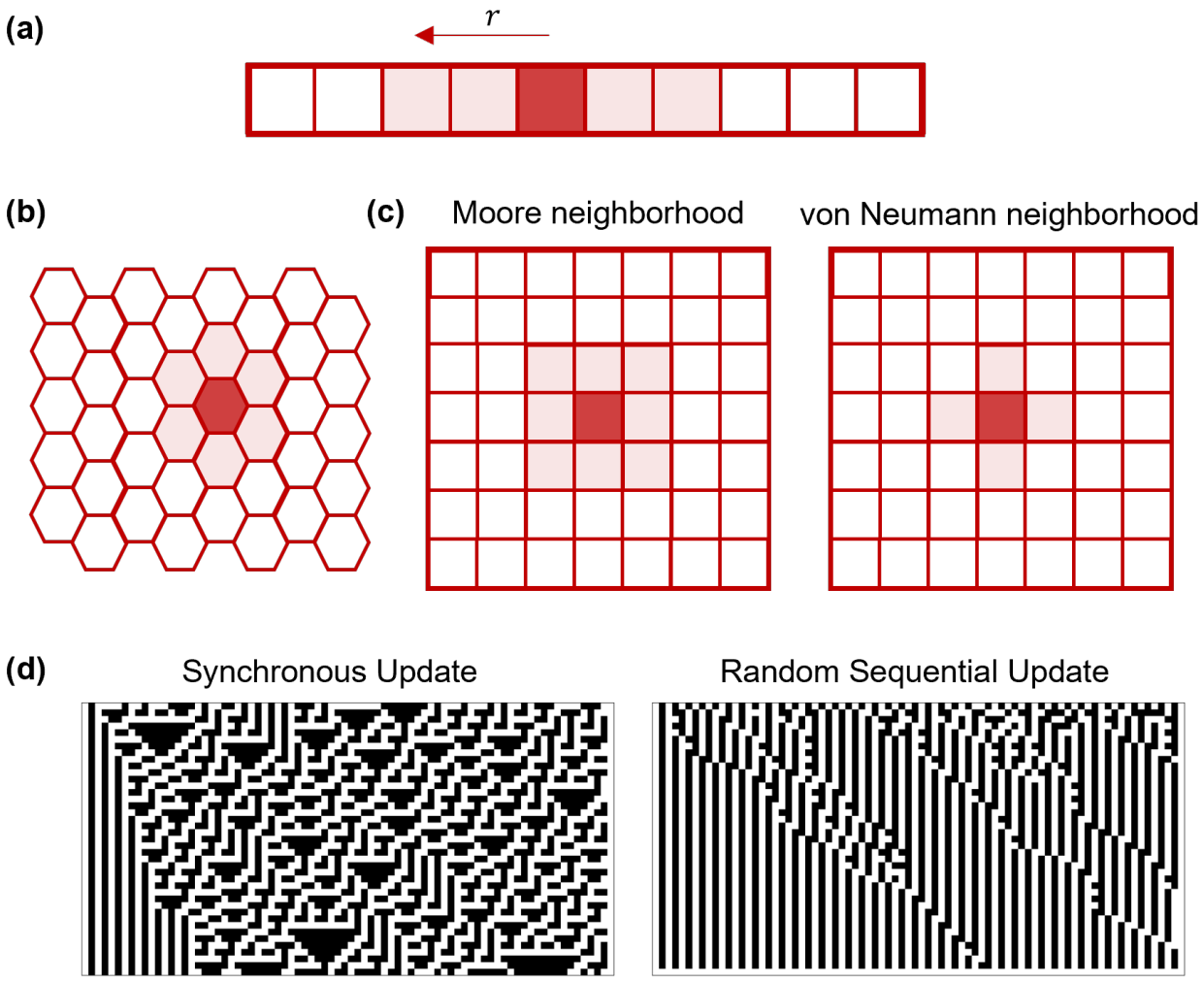


Figure 3.2 CA Definition. (a) In 1D CA, the neighborhood can be specified by a radius which is the distance in number of cells in which cells are counted as belonging to the neighborhood. In the shown example $r = 2$. (b) Hexagonal tessellation of the plane (c) Square tessellations of the plane with examples of the Moore and von Neumann neighborhood. (d) Kymograph of ECA rule 135 with fixed boundary conditions from the same random initial condition with a synchronous and random sequential update, respectively.

1. Typical tessellations in one dimension are the above mentioned line with fixed or periodic boundaries. For two dimensions, square, triangular and hexagonal tessellations are common (fig. 3.2 b, c), but also irregular tessellations may be used to cater the model to biological systems (cp. appendix A.7).

2. In one dimension, the neighborhood function is usually specified by assuming that it is symmetric and giving a radius r determining the number of cells included in the neighborhood in one direction from the cell in question (fig. 3.2 (a)), i.e., ECA would be $r = 1$ CA. Radii larger than one can be realized in a biological setting by either communicating with diffusible molecules or via filopodia [104]. In two dimensions on square grids the usually used neighborhoods are the von Neumann neighborhood including the cell itself and the neighbors at its faces, as well as the Moore neighborhood, additionally including the cells at corners of the original cell (cp. fig. 3.2 c).

3. The size of the neighborhood $|N(c_i)|$ may depend on the cell c_i when the tessellation is irregular, e.g., a cell might have 5 or 6 neighbors.

4. The boundary condition B is simply a set of cells that does not follow the cellular automaton dynamics f, but is either fixed or follows some externally imposed dynamics. This allows an external influence on the pattern formation.

5. In the usual case the update schedule contains all cells in $\mathcal{T}_A$, i.e., a synchronous update. Other well known update schemes are random update (at each point in time a random cell updates with all cells having the same probability of updating) and random sequential update (the update sequence is random, but all cells have to perform an update before the sequence is reshuffled) (fig. 3.2 (d)). Also more complex update schemes beyond this definition with the order depending on the states of the cells are possible and will be discussed in chapter 6 and appendix A.6.1.

6. For the update function f some constraints have been given names. If f does only depend on the sum of the values of the states in the neighborhood (including the cell to be updated itself),$f = f\left(\left(\sum_{j=1}^{|N(c_i)|} c_i\right), t\right)$, this rule is called "totalistic". If f depends on the sum of the values of the states in the neighborhood excluding the cell itself, and the value of the cell itself, the rule is called "outer totalistic".

7. If the update function f is not a deterministic, but a probabilistic function where each output is chosen with a certain probability depending on the inputs, the CA is called probabilistic.

3.3 Classification of Cellular Automata

Classifying CA serves the purpose of creating groups with homogeneous properties, enabling to predict the behavioral characteristics of the class members by investigating a few representatives and have a common identifier for the class object that enables a common research language [103]. For CA, a formal, universally agreed classification scheme has not been reached yet. Here, a few well known schemes which will be used later are briefly described. For a more comprehensive overview see the recent review [103].

3.3.1 Equivalence classes

The idea of equivalence classes can easily described for elementary CA. In total, there are $2^{2^3} = 256$ ECA rules. It is possible to reduce the number of possible different dynamical behaviors by grouping together rules which show exactly the same dynamics after either exchanging the left and the right side in the inputs of the rule as well as mirroring the initial condition, or exchanging the values of the states (colors) in the rule as well as the initial condition, or doing both at the same time (fig. 3.3). If mirroring the side, exchanging the colors or applying both at the same time would for every case lead to a different rule, together with the original rule each equivalence class could be expected to have 4 members and

thus there are $\frac{256}{4} = 64$ classes. Actually, 88 equivalence classes are found since sometimes one of the equivalence operations leads back to the original rule or to another equivalent rule. The behavior of each rule in a class is equivalent and can be computed from the behavior any other rule in the same class by making the appropriate exchanges (color, direction, or both), i.e., for a characterization it is enough to analyze a single member of the class. For other types of CA and research problems, similar equivalence classes can be found, see e.g., chapter 4. While the behavioral mapping for equivalence classes is one to one, the downside is their sheer number, limiting their use quite considerably.

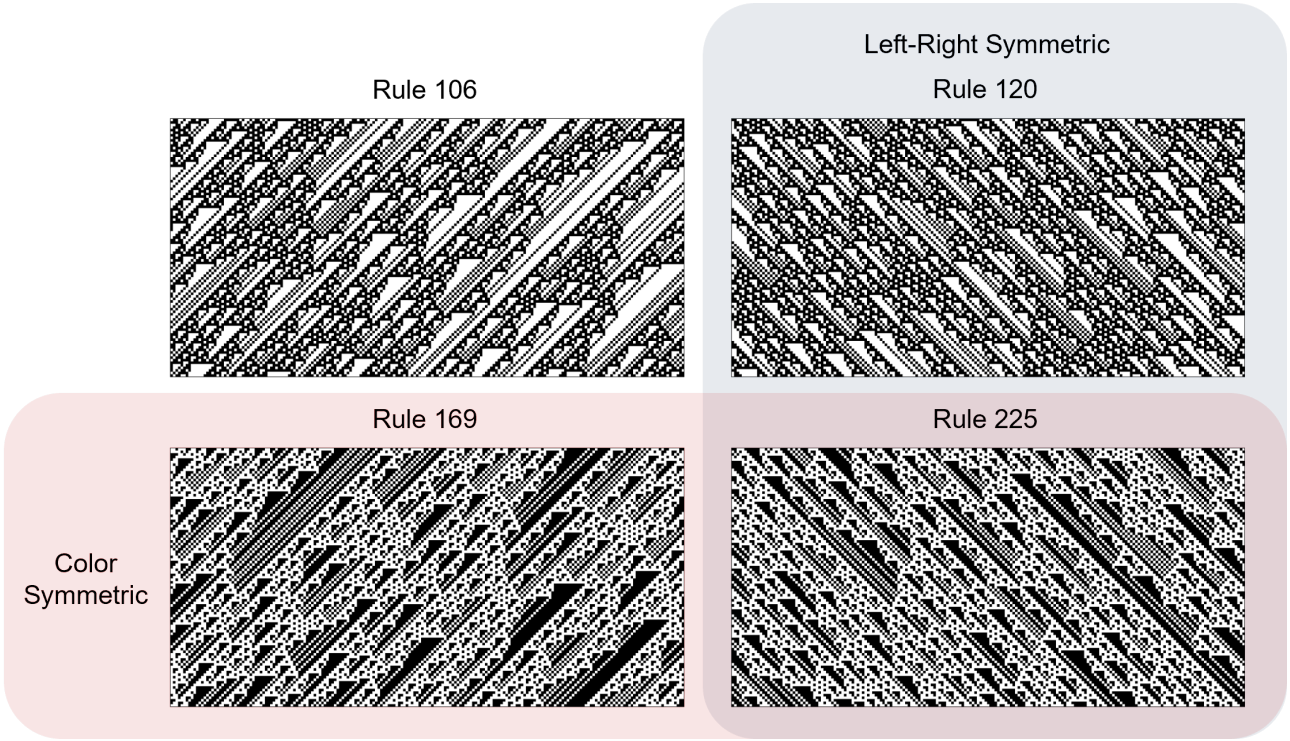


Figure 3.3 88 Equivalence Classes. The 256 ECA rules can be subdivided in 88 equivalence classes: each rule has associated rules which can be generated by exchanging left and right, exchanging the meaning of the colors or both at the same time.

3.3.2 Wolfram Classification

A way more condensing classification has been proposed by Wolfram [44]. By visual inspections, he identified 4 kinds of behavior, which the respective members of a class follow for almost all initial conditions. The behaviors are exemplified in fig. 3.4:

(W1) Class 1 consists of rules which quickly converge to a homogeneous final state.

(W2) Class 2 rules exhibit “a simple stable or periodic structures” [43, p. 17] in the long term behavior.

(W3) For class 3, this behavior appears to be chaotic.

(W4) For class 4, “complicated localized structures, some propagating” are found. [43, p. 17].

It is claimed by Wolfram that this classification is able to order all one dimensional CA [44], although it is admitted, that the classification if often formally undecidable [105], which was shown by Culik and Yu [106]. The Wolfram classification scheme can arguably still be seen as the most influential since its widespread use and being the reference point for other classification schemes [103].

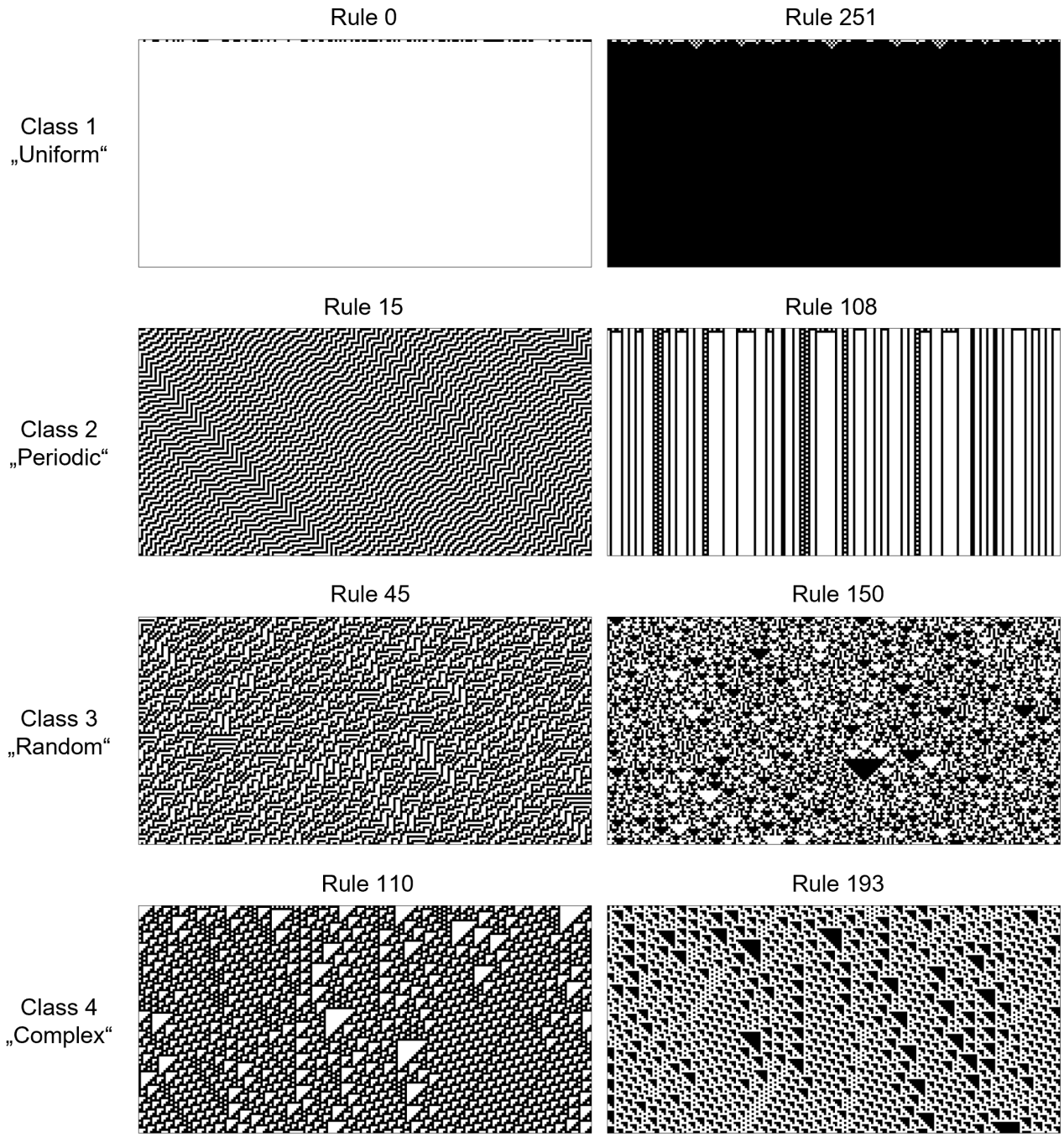


Figure 3.4 Wolfram Classes. Examples for the 4 Wolfram classes: Class 1 (first row), consisting of rules with very simple behavior, “almost all initial condition lead to exactly the same uniform final state” [105, p. 231], showcased here by rule 0 and 251. Class 2 (second row), with “many different possible final states, but all of them consist just of a certain set of simple structures that either remain the same forever or repeat every few steps” [105, p. 235], showcased here by rule 15 and 108. Class 3 (third row), depicting a behavior that is “complicated, and seems in many respects random” [105, p. 235], showcased by rules 45 and 150. Class 4 (fourth row), with rules that “involves a mixture of order and randomness” [105, p. 235] with localized structures which are produced, move around and interact [105].

3.3.3 Rule Tables Based Schemes

The rule table or "genotype" of the rule [103] determines the update dynamics of each site. Since the rule table is quite small compared to the data from simulating the rule from many initial conditions to determine the "phenotype" of the rule, a characterization of the rule solely based on its table promises a low resource possibility of classifying the rules. Due to the strong non-linearity of the rules where slight changes in the rule table can change the kymograph strongly (cp. fig. A.1), defining a parameter based on the rule table that projects the essential features of the spatio-temporal dynamics is a challenging task. In the following we describe a few such parameters.

1. **Langtons** λ: Langton's Lambda is a parameter defined in reference to a so called "quiescent" state q which, when it is the sole input, maps to itself, $f(q,q,q,...) = q$. Then, λ is the fraction of inputs that do not map to the quiescent state,

$$\lambda = \frac{k^{|N|} - n_q}{k^{|N|}} \tag{3.3}$$

with n_q the number of input configurations that map to q, k the number of states and $|N|$ the size of the neighborhood [107] (fig. 3.5 (a)). $k^{|N|}$ is the total number of inputs. Of course, it is also possible to calculate λ with respect to a state that is not quiescent. For the example in fig. 3.5 (a), with a $k = 2$ state rule with a neighborhood size of $|N| = 3$ and quiescent state $q = 0$, $k^{|N|} = 8$ and $n_q = n_0 = 4$ and thus $\lambda = \frac{1}{2}$. In general, $\lambda = 0$ maps all inputs to the quiescent state, i.e., it is the most homogeneous rule, while the most heterogeneous one, represented by an equal mapping to all possible states, has $\lambda = 1 - \frac{1}{k}$. From $\lambda = 0$ to $\lambda = 1$, the Wolfram classes 1-3 lie approximately in increasing order [107]. Class 4 rules are dispersed at the transition between class 2 and 3 rules as the transition length, i.e., the time the CA takes for a periodic behavior to emerge, increases [107].

2. **Kolmogorov Complexity**: The Kolmogorov complexity of a string x is defined as the length of the shortest program running on a Turing machine that outputs x and halts[2]. Naively, a more complex rule is expected to produce a more complex kymograph [108]. Unfortunately, the Kolmogorov complexity can not be computed directly [109], it is only possible to give an upper bound [109]. The usual way to achieve this is writing down the outputs of the rule table as a string and compressing that string with a lose-less compression method like the Lempel-Ziv compression algorithm [110]. The estimator for the complexity is then the length of the compressed string plus the length of the compression algorithm (fig. 3.5 (b)). Since the latter is the same for all rules, i.e., an additive constant, it can be dropped when comparing complexities from the same estimator.

3. μ**-Sensitivity**: The μ-sensitivity is defined [103, 111] as the changes in the output of the rule by changing one of the sites in the neighborhood

$$\mu = \frac{1}{|N|\, k^{|N|}} \sum_{c_1 c_2 ... c_{|N|}} \sum_{q=1}^{|N|} \frac{\partial f}{\partial c_q} \tag{3.5}$$

with the Boolean derivative [112]

$$\frac{\partial f}{\partial c_q} = \begin{cases} 0 & \text{if } f\left(c_1, ..., c_q, ..., c_{|N|}\right) = f\left(c_1, ...\bar{c}_q, ..., c_{|N|}\right) \\ 1 & \text{otherwise} \end{cases} \tag{3.6}$$

with $\mu \in \left[0, \frac{1}{2}\right]$ for ECA (fig. 3.5 (c)). The μ sensitivity is low for rules producing ordered behavior and large for rules with chaotic behavior [111].

[2] More specific, the Kolmogorov complexity of a string x is defined as

$$K_\varphi(x|y) = \min\left\{l\left(P\right), \varphi\left(P, y\right) = 0\right\} \tag{3.4}$$

with Turing machine φ, programs P, length function l, and input string y. y is usually taken as empty string, and since different Turing machines φ changing the result only be a constant, both references are usually dropped.

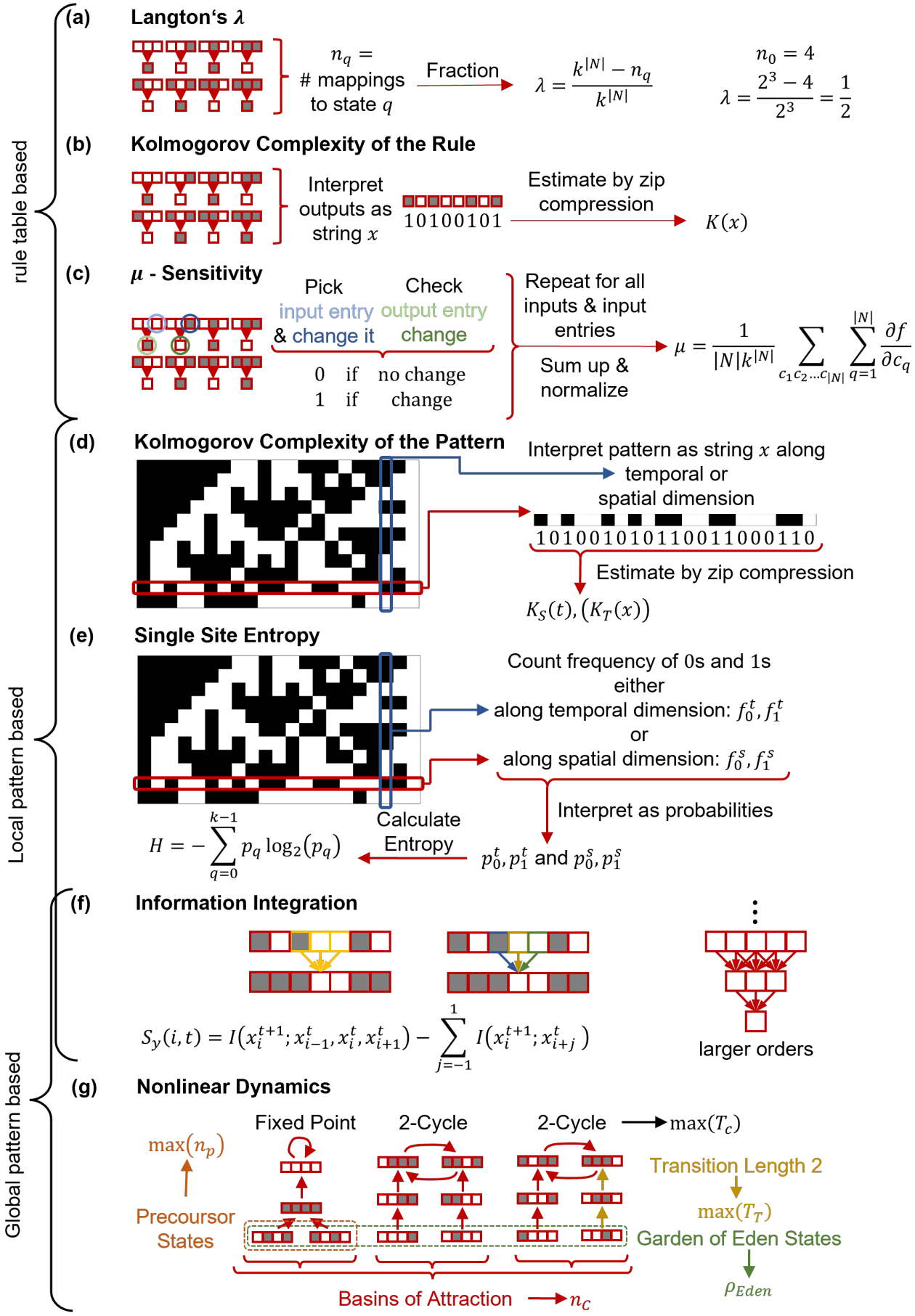

rule table based
(a)
Langton's λ
n_q = # mappings to state q
Fraction
$\lambda = \frac{k^{|N|} - n_q}{k^{|N|}}$
$n_0 = 4$
$\lambda = \frac{2^3 - 4}{2^3} = \frac{1}{2}$
(b)
Kolmogorov Complexity of the Rule
Interpret outputs as string x
10100101
Estimate by zip compression
$K(x)$
(c)
μ - Sensitivity
Pick input entry & change it
Check output entry change
0 if no change
1 if change
Repeat for all inputs & input entries
Sum up & normalize
$\mu = \frac{1}{|N|k^{|N|}} \sum_{c_1 c_2 \ldots c_{|N|}} \sum_{q=1}^{|N|} \frac{\partial f}{\partial c_q}$
Local pattern based
(d)
Kolmogorov Complexity of the Pattern
Interpret pattern as string x along temporal or spatial dimension
10100101011001100011 0
Estimate by zip compression
$K_S(t), (K_T(x))$
(e)
Single Site Entropy
Count frequency of 0s and 1s either
along temporal dimension: f_0^t, f_1^t
or
along spatial dimension: f_0^s, f_1^s
Interpret as probabilities
p_0^t, p_1^t and p_0^s, p_1^s
Calculate Entropy
$H = -\sum_{q=0}^{k-1} p_q \log_2(p_q)$
(f)
Information Integration
$S_y(i,t) = I(x_i^{t+1}; x_{i-1}^t, x_i^t, x_{i+1}^t) - \sum_{j=-1}^{1} I(x_i^{t+1}; x_{i+j}^t)$
larger orders
Global pattern based
(g)
Nonlinear Dynamics
$\max(n_p)$
Precoursor States
Fixed Point
2-Cycle
2-Cycle
$\max(T_c)$
Transition Length 2
$\max(T_T)$
Garden of Eden States
ρ_{Eden}
Basins of Attraction
n_C

Figure 3.5 Quantitative Classification Schemes Schemes to classify rules quantitatively include the rule table and pattern based schemes. Into the former category fall Langton's λ (a), the Kolmogorov complexity of the rule (b) and the μ sensitivity. The latter quantity can be split up in schemes which focus on local properties and schemes which focus on global properties. The Kolmogorov complexity of the pattern (d) as well as the single site entropy (e) in spatial or temporal direction are examples for schemes focusing on local properties, while looking at attractors and transients (g) focus on global properties. Information integration or synergy (f) may fall in both categories.

3.3.4 Spatio-temporal Dynamics Based Schemes

Looking more on the "phenotype" of a rule, i.e., the spatio-temporal dynamics a rule generates for a certain initial condition, may also be a fruitful approach to classify CA. Wolframs classification based on visual inspection (section 3.3.2) is one possible approach. More quantitative approaches take either small snapshots of the spatio-temporal dynamics and calculate an observable from these snapshots (local pattern based), averaging over snapshots from different initial conditions, or try to classify the dynamics more based on the whole spatio-temporal dynamics, possibly for all initial conditions (global pattern based).

1. **Kolmogorov Complexity**: Interpreting the pattern as string allows to assign each pattern a Kolmogorov complexity by using an estimation algorithm as described in section 3.3.3 (cp. fig. 3.5 (d)). To arrive at a single number for a rule, ensemble averaging over many initial conditions and either additional temporal averaging or using a pattern snapshot from time $t >> L$ can be used.

2. **Single Site Entropy**: Intuitively, looking at the number of occurrences of the states in a pattern (spatial dimension) or along the time trace of a fixed cell (temporal dimension), a different frequency distribution can be expected for different rules, with rules producing fixed and/or homogeneous patterns having a more peaked frequency distribution, while rules producing chaotic patterns/time traces have a more uniform frequency distribution. This intuition about the frequency distribution can be quantified by the Shannon entropy of a rule. The information theoretic Shannon entropy over a discrete probability distribution p_i is defined as [113]

$$H = -\sum_{i=0}^{k-1} p_i \log_2 (p_i) \,. \tag{3.7}$$

   It quantifies the amount of information that can be expected to be acquired by drawing a sample from the distribution or, phrased differently, the number of yes or no questions that have to be asked on average to identify the sample. [114, 115]. Interpreting now the either spatial or temporal frequency distribution as probability for the occurrence of a certain state and performing appropriate ensemble averages, the so called single site entropy of a rule can be defined [107] (cp. fig. 3.5 (e)).[3]

3. **Information Integration/Synergy**: To introduce information integration, first the concept of mutual information has to be introduced. The mutual information between I two CA states x_i^t and x_j^{t+s} is defined as [114]

$$I(x_i^t; x_j^{t+s}) = -\sum_{a,b} \frac{p\left(x_i^t = a, x_j^{t+s} = b\right)}{p\left(x_i^t = a\right) p\left(x_j^{t+s} = b\right)} = H\left(x_i^t\right) - H\left(x_i^t | x_j^{t+s}\right) \tag{3.8}$$

   with i and j being the spatial positions of the cells and the sum runs over the possible states. It measures the reduction in uncertainty in the knowledge about x_j^{t+s} when learning x_i^t. Now the information integration or synergy can be defined [116, 117] as the difference between the mutual information between a state x_i^t and a set of states Υ and the sum of the mutual information between x_i^t and the individual members of Υ,

$$S_y = I\left(x_i^t; \Upsilon\right) - \sum_{x \in \Upsilon} \left(I\left(x_i^t; x\right)\right) . \tag{3.9}$$

[3] Note that there are also classifying attempts for CA using entropy concepts with a more complex "unit cell" than a single site by combining several sites together, the so called block entropies [116].

For example, for the Υ containing the same cell at the previous time step as well as the two neighbors at the previous time step (cp. fig. 3.5 (f)), the information integration/synergy is

$$S_y = I\left(x_i^t; x_{i-1}^{t-1}, x_i^{t-1}, x_{i+1}^{t-1}\right) - \sum_{j=-1}^{1}\left(I\left(x_i^t; x_{i+j}^{t-1}\right)\right) \tag{3.10}$$

As a "whole" - "sum of the parts" quantity it measures the dependencies that are introduced by the rule between the chosen states in Υ at x_i^t. Phrased differently, it measures how much more was learned about x_i^t by knowing the states in Υ at the same instance than what was learned about x_i^t in total by knowing each member of Υ individually. When testing different information processing features as classification proxies for Wolfram's CA rule classes, the information integration or synergy was found to perform best [117]. Also larger orders over several time steps can be included. The more time steps are included, the more the synergy turns from a quantity based on local patterning information to a quantity based on global pattern information.

4. **Attractor based classifications**: The qualitative theory of nonlinear ordinary differential equations focuses on determining how the the solution of an initial value problem changes when changing the initial value [118], more specifically determining the fixed points and limit cycles, as well as their stability and basins of attraction [119]. For CA this can be approached in a similar [120] but more limited fashion due to the exponential growth of the number of possible patterns k^L with CA size. To quantify the complete dynamcis of a CA, a restriction to small system sizes is necessary. Then the dynamics can be visualized as a graph (fig. 3.5 (g)) with the vertices being the individual possible patterns and the directed edges the transitions between these patterns as specified by the rules [120]. Since the dynamics is deterministic, each vertex (pattern) has exactly one outgoing edge (i.e., the out-degree of every vertex is one), while each pattern can have different predecessors (pre-images), i.e., different in-degrees are possible. Since the state space is finite, the dynamics can either run into a fixed point (i.e., a pattern mapping to itself) or into a limit cycle. Thus, the graph separates into different components (basins of attraction) with an attractor (fixed point or limit cycle) each [120]. To specify the dynamics of a given rule for a fixed system size and boundary conditions, the number n_C and period T_C of attractors, the transient lengths T_T to reach the attractors, as well as the density of "Garden of Eden states" ρ_E (vertices without in-degree, i.e., unreachable from other states) [120] and the number of precursors (in degree) n_P can be calculated. The scaling of n_C, $\max_C(T_C)$, $\max_C(T_T)$, ρ_E and $\max_{Vertices}(n_P)$ can now be investigated as a function of system size for small system sizes. Note that these quantities are not independent of each other, as a large quantity of one of n_C, $\max_C(T_C)$, $\max_C(T_T)$ implies lower values of the other two.

Numerical values for the above classification quantities are summarized in table A.1.

3.4 Examples of Modeling of Biological and Artifical Systems with Cellular Automata

CA have been used previously to model a wide range of developmental and synthetic pattern forming systems. The applications range different model species, from plants like a model for the primary shoot meristems of *arabidopsis thaliana* [121], over insects like neurogenesis in *drosophila* [37] and vertebrae lizard skin scales [122], to mammalian cell-cell interaction in human and rat cells [123]. Also, a wide variety of different phenomena can be modelled, for example growth processes [124], adhesive cell interactions [124], cellular swarming behavior [124], stripe formation in pigment cells [124], tumor development [124], animal coat markings [125] and cell differentiation processes [126].

Apart from concrete systems, also more abstract analysis has been conducted using CA as modelling tool. For example, in a more general secrete and sense CA framework, the statistical dynamics of spatial order formation has been investigated with a focus on whether larger scale order, characterized by a large

Moran's I, appears [127]. CA are also used as an abstract toy model to investigate of unbound evolution and novelty generation by having 2 interacting CA, where the state of one CA influences the rule of the other [128, 129].

In summary, for special systems CA has been used to some degree as a successful modelling tool for specific systems and some cases of more abstract biological systems.

3.5 Modelling Assumptions, Advantages and Disadvantages of CA Models

3.5.1 Modelling Assumptions

CA are based on a specific set of assumptions, which make them applicable to some biological regimes. To map out theses regimes, in this section the assumptions are stated and discussed with regard to their biological implications. Some of these assumptions are also relaxed in later chapters.

1. **Discretization of cell states**: CA have discrete states, while in biological cells the question whether cell states are discrete or continuous is still open. Experimental cell type classification methods started from unspecific dyes and developed into specific antibody staining and later hybridization based techniques allowing for a detection cell type marker molecules [130]. Another possibility is the characterization via functional assays, defining cells states based on the cellular behaviour [130]. Classically, cells states were assumed to be discrete with cell transitioning through commitment points. Yet, this view has been challenged by modern day "-omics" suggesting a more continuous trajectory in gene expression space by looking at molecular characterization of cell types that are based on global profiling using high throughput methods [130].

   The traditional visual metaphor [131] for epigenetic dynamics is the Waddington landscape [132]. It shows (potentially multiple) supercritical pitchfork bifurcations (cp. fig. 3.6) as a function of some usually unspecified parameter. The idea can be visualized by the simple equation:

   $$\frac{dx}{dt} = sx - x^3 \tag{3.11}$$

   The fixed points of this equation are $x = 0$, which is stable for $s < 0$ and unstable for $s > 0$, and $x = \pm\sqrt{s}$ for $s > 0$ (stable). A cell moving in this landscape as the parameter s increases starting from negative values resides in the stable fixed point $x = 0$ until $s = 0$. Then, the cell has to decide to adopt one of two possible states $x = \pm\sqrt{s}$ as soon as $s > 0$ The viewpoint of a dynamical system can be extended further as proposed by Kaufmann by asserting that stable cell types correspond to attractors in a dynamical cell state landscape (fig. 3.7 (a)) towards which cells move [133]. Modifying the landscape can then lead to the adaption of a different attractor (fig. 3.7 (b-e)). Recently, more quantitative versions of this metaphoric picture based on experimental data was constructed by Saez et al. for mouse embryonic stem cells [134]. Such a picture, even if transitions between cell states are continuous, would support a modelling of cell states as discrete by identifying the attractors with the discrete labels.

   For the discrete modelling of the cell state transitions via the CA rule also exist established models with a similar spirit, e.g., the modelling of biochemical networks through boolean models [135] or the discretatization of Hill functions describing gene regulation through boolean like approaches which allow to view the integration of several signal via logic gates [89].

   Futhermore, it has been proposed that the exact inner workings of cells are irrelevant to development, rather important are the cells, output, response to external signals and biophysical properties [136].

2. **Reversibility of Cell State Transitions**: Principally, CA models place no restriction on which state can transition to which other state, i.e., cells switching state back and forth are allowed in this framework, and are thus more general than the often irreversible cell differentiation. Yet, cell plasticity

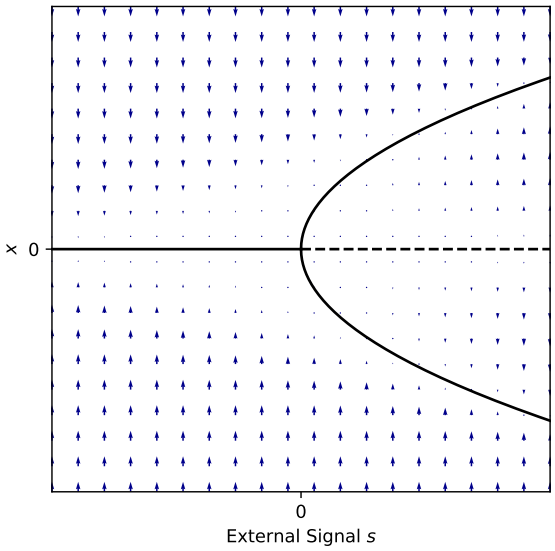


Figure 3.6 Prototype of the Waddington Landscape. Shown is a single supercritical pitchfork bifurcation [119]. For a parameter $s < 0$, a cell in this landscape moves towards the only stable fixed point $x = 0$. If the parameter is increased to $s > 0$, $x = 0$ becomes an unstable and $x = \pm\sqrt{s}$ stable fixed points. The metaphor of the Waddington landscape is to assert a cell the state x. When the parameter s is increased, from $x = 0$ onward the cell starts to follow one of two branches $x = \pm\sqrt{s}$, corresponding to two different cell states.

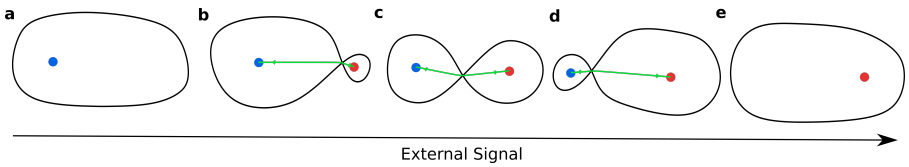


Figure 3.7 Schematic 2 State System with External Signal. Represented is a potential landscape of the cell type dynamic of a single cell: an equipotential (black line) of the dynamical system together with the steady states (blue and red) and dynamical trajectories (green) originating at the saddle point of the potential surface, all for different values of the external signal. When the external signal increases, the basin of attraction of the blue fixed point decreases and the red fixed point appears (b), which continues (b-d) until the blue fixed point vanishes and the red fixed point is the only one left (e). If the cell type is initially “blue” for low external signal, it will change to “red” for large enough external signal.

and de-differentiation is also an observed phenomenon [137], thus CA can in essence span the required range of cell state transitions in biological systems. Furthermore, as shown, e.g., in chapter 4, coordinated cell state transitions can sometimes be interpreted as cell movement or sorting.

3. **Immobility**: The cells in a CA are ordered on a fixed grid, which either amounts to assuming stationary cells forming a rigid tissue matrix in biological context [138] that is, e.g., the standard case for plants [8], or a patterning dynamics that is fast compared to topological changes in the tissue. In chapter 8 this assumption is relaxed by also including dynamic grids.

4. **Local Signaling**: The neighborhood determines the range of the signaling in the CA dynamics. In principle it is possible to include arbitrary connections, but the most common choices are neighborhoods that only include the nearest neighbors or next nearest neighbors. This allows for a modelling of local signals in the sense of the previous chapter, including signaling via portrustions.

5. **Deterministic Dynamics**: The standard CA dynamics is purely deterministic, despite the inevitably noisy nature of biological systems (see e.g., [63, 139, 140]). The discretization buffers some of the noise by coarse graining the many possible concentration combinations of bio-molecules into few discrete states, but the transitions between those states might still be noisy. Turning to a probabilistic CA allows for a modeling of said noise at the expense of introducing additional parameters. A simple possibility for turning a deterministic CA into an probabilistic CA for low noise is to assume that the CA follows the deterministic rule, but with a probability p it maps to a different state, with the same probability for all possible states. Thus, the deterministic rule is followed with probability $q = 1 - \sum_{i=0}^{k} p$. The exact modelling choice also greatly influences the dynamics and stochasticity of the system, e.g., a not modeled "hidden state" might turn a nearly deterministic transition into a stochastic one [141].

6. **Directional Sensing of Signals**: A general CA rule allows in principle to discriminate between the directions from which a signal in the neighborhood is coming from. From a biological perspective, this would require some kind of cell polarity. There are common restrictions on CA rule space where the signal direction is irrelevant, e.g., outer-totalistic or totalistic rules (see above), which can be enforced when the modelled phenomenon does not include directional signal processing.

7. **Discrete Time Dynamics and Synchronization**: CA dynamics assumes discrete steps in time, which can be justified as follows. From the perspective of continuous dynamical system, state changes can be seen as as transitions between concentration fixed points of molecules determining cell state. Assuming a discrete time dynamics together with discrete states is then consistent with a smaller timescale for the change between these fixed points compared to the retention time in the fixed points. Yet, the retention times between state transitions do not have to be equidistant in real time, allowing for an implicit modelling of phenomena like slowing down of cell cycle dynamics [142]. Another critical assumption is the synchronicity of the updates of the cells. The exact accuracy of synchronization needed may depend on the time it takes for the signal to be relayed between different cells and processed within the signal receiving cell, as well as the retention time between state transitions, compare chapter 6 for a more detailed analysis. Synchronization between biological oscillators is possible via only local interactions [13], which may allow for the ability of a synchronized patterning dynamics.

8. **"Markovian" Dynamics** The standard CA form calculates the new state of a cell only from state values of the previous time step, similar to a stochastic process that is Markovian [143]. In a biological context, this requires a degradation of the signaling molecules on a timescale shorter than the (real) time of a single update. CA implementations with memory usually use for the inputs into an update rule an exponential weighted sum of the past time steps, e.g., [144]

$$s_i^T = \frac{x_i^T + \sum_{t=1}^{T-1} \alpha^{T-t} x_i^t}{1 + \sum_{t=1}^{T-1} \alpha^{T-t}}. \tag{3.12}$$

This corresponds to a degradation of signaling molecule with a rate related to α that is small compared to the inverse of the (real) time between two updates. Additionally, to assume a “Markovian” dynamics when constructing the landscape of cell states on which the finite CA states are defined (cf. assumptions on discretization of cell states above), it must be ensured there are no hidden dimensions, i.e., types of molecules relevant to the state of the cell which are not included in the dynamical systems picture. Such an omission of relevant cell states could in principle lead to an identification of 2 different states $A1$ and $A2$ as a single state A, possibly introducing dependencies on the past for the dynamics of A since for one trajectory $A1$ may be the actual state, while for another it could be $A2$.

Some of the assumptions listed above are relaxed or their justification investigated in the following chapters by extending the basic CA model viewing CA as " [...] philosophy of modeling where one seeks a description in terms of simple but essential mechanisms" [145]. In chapter 4, an asynchronous model variant is studied during the analysis of the results, and in chapter 6, it is checked how local interactions can sustain global synchrony between cells. Chapter 4 starts a discussion about a changing grid by introducing cell division, chapter 8 extends this discussion by an extension of the basic CA model to dynamic grids allowing for cell movement, division, death and interactions between grid dynamics and CA dynamics. Chapter 7 couples local CA-type dynamics to long range signaling. The effects of noise on CA are investigated in chapters 4, 5, 6, and 7, fully switching to a probabilistic CA in chapter 9.

3.5.2 Disadvantages and Advantages of CA as Models

While CA as member of the class of agent based models seem like a natural framework to investigate cellularized systems with each cell corresponding to an “agent”, the previous chapter discussed the long list of assumptions that need to fulfilled to be able to employ them as models. While many of these assumptions can be relaxed as will be demonstrated throughout the thesis, developmental systems might not always fulfill this large list of eligibility criteria to be suite-able to be modeled by CA. Also adapting a CA model to a concrete case can prove to be difficult since CA usually have little flexibility to adjust parameters [145]. Even small changes in rule may lead to totally different outcome (as an example compare rule 180 and 181 in fig. A.1 differ only by one rule entry but show completely different behavior from seed initial conditions).

On the plus side, CA are usually relatively easy to implement and have a great numerical stability [145]. The discrete nature of CA partially allows for an exhaustive numerical search for solutions in CA rule space with found rules and dynamics often being easily interpretable and visuablizeable. CA can model both linear and non-linear systems, and the space of all CA rules contains a great variety of different dynamical behaviors (cf. section 3.3 and following sections). Finally, due to their “age” a large body of CA-theory is available which can be relied on.

4 How to Form Global Patterns with Local Interactions? - French Flag Problem Revisited

The main content of this chapter is the result of the collaborative work of the author of this thesis together with Gabriel Vercelli, which will be jointly published in [146]. Smaller parts where developed as collaboration between the author and Maria Molina Antón (section 4.11.3, [147]) and Lukas Zett (section 4.11.4, [148]) and are marked accordingly. All were supervised by Ulrich Gerland.

4.1 Global Patterns from Local Interactions: Forming French Flags

The analysis of pattern formation with local interactions is started by looking at one of the most simple model systems: a 1D grid with fixed boundary conditions. From a biological perspective, such a system may be of interest because biological systems are finite and thus usually have a boundary[1] - may it be cells of another tissue or the surrounding environment. Besides "real" one dimensional tissues and organisms [84, 149], 2D stripped patterns are ubiquitous in nature [150, 151] - due to the symmetry of the problem along the stripes a simplifying assumption can be that such patterning processes can be reduced to be modelled effectively in 1D as a first approach.

The dynamics of pattern formation is modeled by a cellular automaton with synchronous updates as described above. A pattern formation process described by this model has two inherent lengths scales: The interaction range of the cellular automaton, which will be set to $r = 1$ initially, and the extension of the grid itself.

On the one hand it is comparably easy to see how patterns with a length scale determined by the length scale of the interaction can be formed - for example to produce an alternating pattern in a $k = 2$ state system, it suffices to use rule 15 ($x_i^{t+1} = 1 + x_{i-1}^t$) which, after L steps, fills the pattern with alternations of the left boundary state (cf. fig. A.2). On the other hand, the problem of finding pattern formation techniques that construct patterns with wavelengths related to the grid size L seems less trivial. Such patterns are known to be formed using long range morphogen gradients [2], but here the goal is to investigate how to form them with short range interactions.

As a first simplification of the problem, a generic "target" pattern is specified which should be the stationary state that the cellular automaton reaches when starting from arbitrary initial conditions, namely patterns with s stripes with equal width for s states in the final pattern. For 2 states, this corresponds to the left half of the final pattern being 0, the right half being 1. For 3 states, it matches Wolpert's iconic French Flag pattern [20] with equal stripe widths. Different stripe width are investigated later.

To systematically screen rules ("genotypes") for their ability to form patterns ("phenotype") resembling the French Flags from arbitrary initial conditions, a mathematical criterion has to be defined - in this case the relative agreement with the target pattern (relative number of cells with the same state) after a fixed simulation time $2L$,

$$\Phi = \frac{1}{L} \sum_{i=1}^{L} \delta(x_i^{2L}, x_i^{\text{goal}}), \tag{4.1}$$

which is termed "fitness" of the rule (fig. 4.1). Other definitions of the fitness functions are possible and may be useful in specific contexts, e.g., when looking for applications where the location or width of a specific stripe is more vital than that of other stripes, or loss of biological function is not linearly associated

[1] it may be argued that in problems like the neural tube periodic boundary conditions are an appropriate modelling choice because of the eponymous tube geometry, so not all finite systems do necessarily have a boundary from a modelling perspective

with the number of cells in the wrong state, but depends on the phenotype in a more involved fashion. In the abstract context here, only the relative agreement is used. Since the final pattern and thus the fitness is not only a function of the rule but also of the initial condition, an average over different initial conditions is taken. Checking all initial conditions is usually not feasible since the number scales with system size as k^L. Therefore, if not stated otherwise, $N_{Sample} = 50$ initial conditions are sampled for a length of $L = 100$ and a maximum simulation time of $T_{max} = 200$.

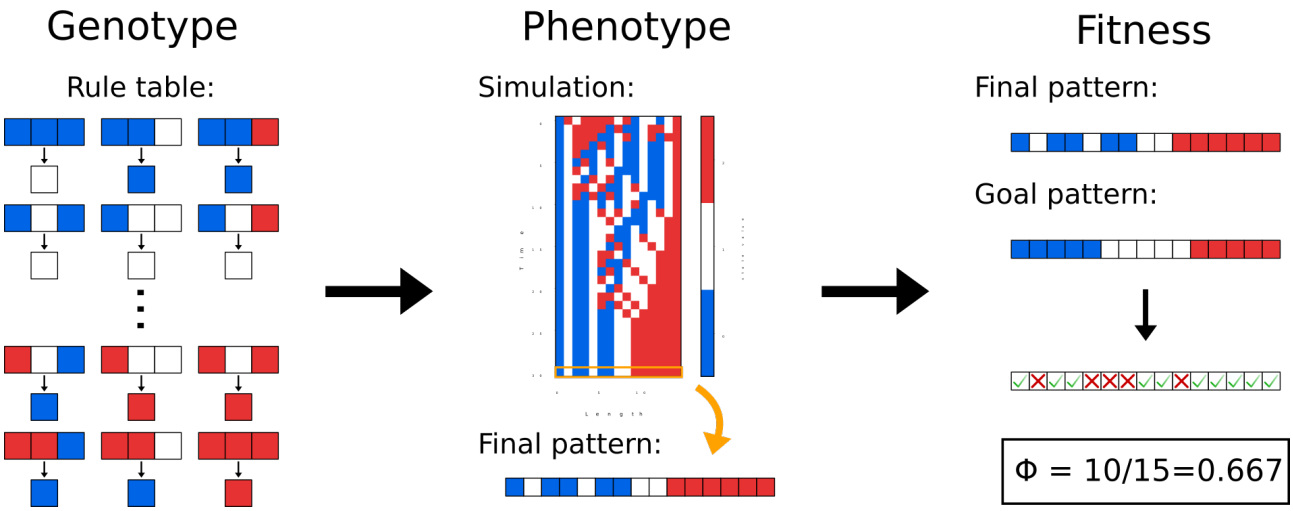


Figure 4.1 Genotype, Phenotype and Fitness. The outputs specified in the rule table can be interpreted as the genotype of a rule. Together with an initial condition they give rise to the phenotype visualized by the kymograph and result in a final pattern after a fixed amount of time. To determine the fitness associated with this final pattern, it is compared to a goal pattern and the fitness is calculated as the relative amount of matching states.

4.2 Only one Solution in 2 State Space

In the $k = 2$ state space, all rules can be explored exhaustively. While the fitness for most rules is close to the value for a random choice of states 0.5, Rule 29 (marked red) has a fitness close to 1 (fig. 4.2 (a)). The high fitness of rule 29 is not an artefact of the specific system size chosen for the fitness calculation in fig. 4.2 (a), but consistent across several orders of magnitude of system size (fig. 4.2 (b)). The mechanism by which rule 29 manages to achieve such a high fitness is illustrated by looking at a kymograph (fig. 4.2 c). If the pattern is checkerboard like, blue and white states just switch their place in the bulk of the checkerboard. Checkerboards expand as long as they do not hit a completely blue patch to the left or a completely white patch to the right. The blue (white) patches grow to the right (left) when hitting checkerboard like patterns, and stay stagnant when hitting the other colored patch (ensuring a stable steady state). This rough behavioural description allows to split up the rule table in the mappings responsible of conserving the steady state, and mappings generating and conserving the checkerboard pattern (fig. 4.2 (d)).

4.3 Searching Vast Rule-Spaces: Evolutionary Algorithms and Multiple Sequence Alignment

In the case of $k = 3$ states, exhaustively searching the whole CA rule space consisting of 7.63×10^{12} rules is hardly feasible, especially if not only a single target pattern is of interest but multiple under possibly different settings. Thus, an efficient sampling algorithm has to be used. For cellular automata, evolutionary or genetic algorithms (EA) have been employed in a similar setting trying to find rules that mimic early *Hydra* development [152]. EA are a computational exploration technique that employs motives found in natural evolution, but does not necessarily try to mimic real biological processes. The version used here

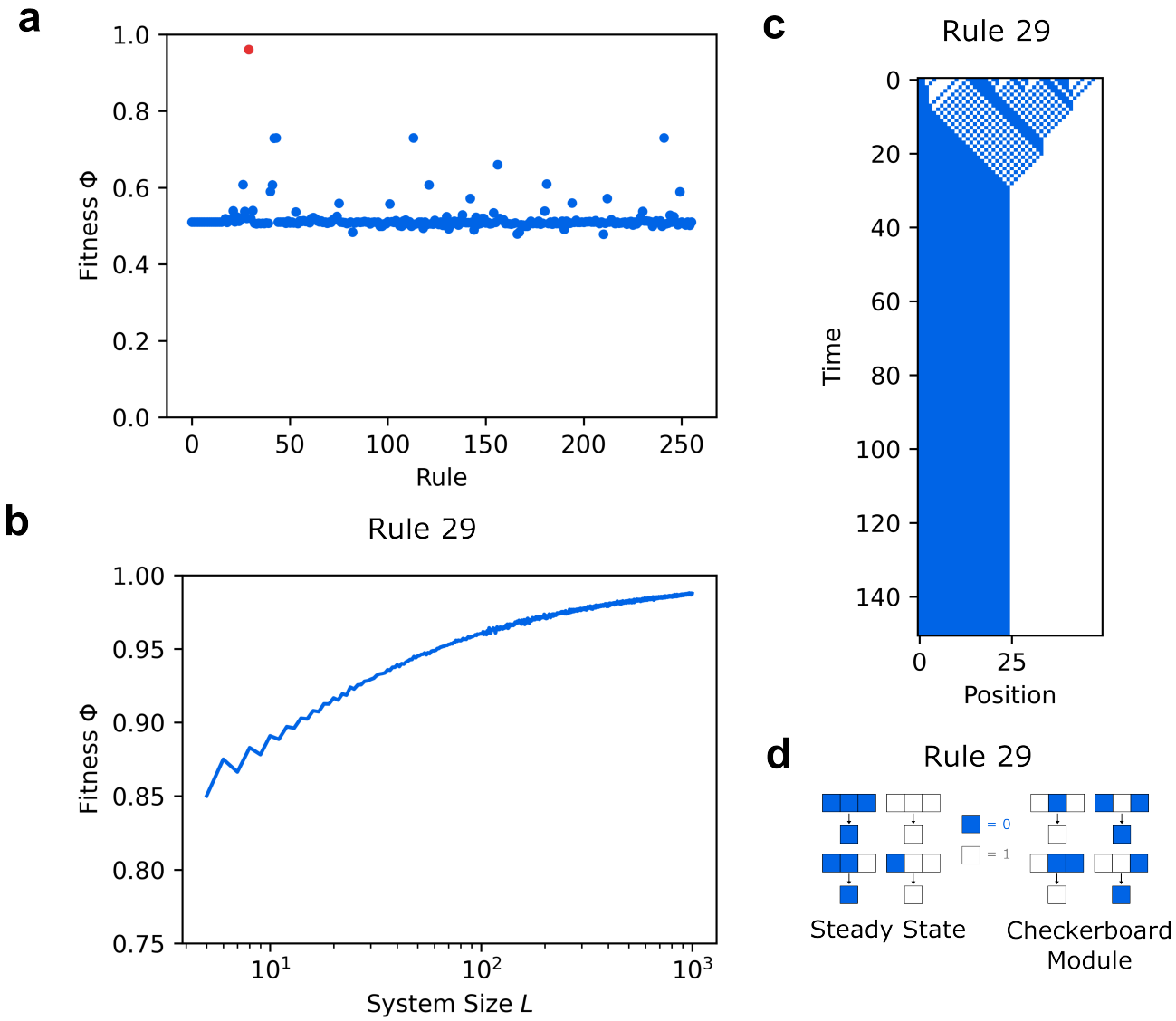
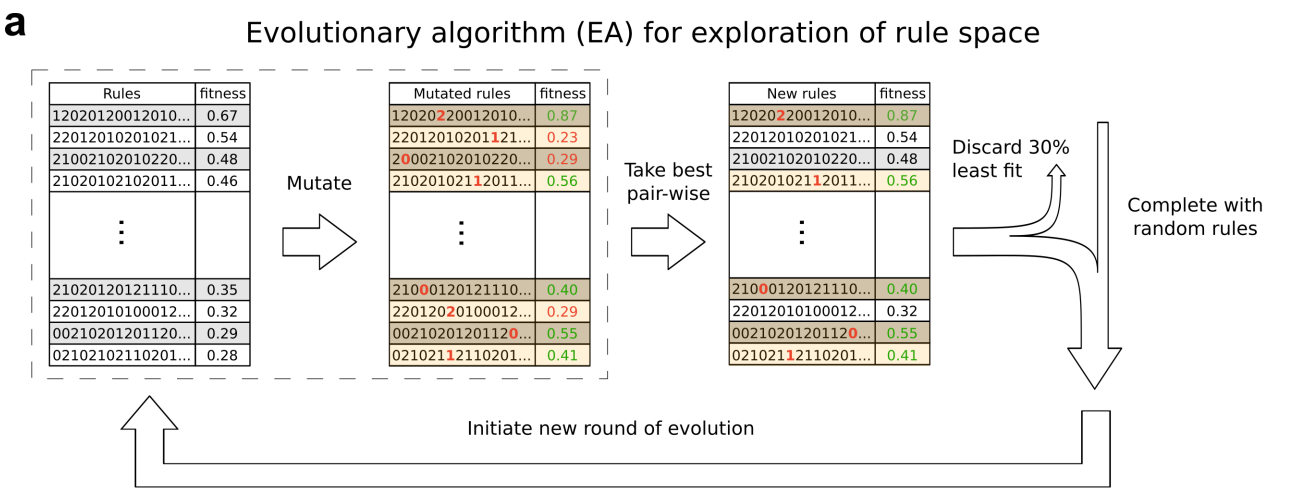
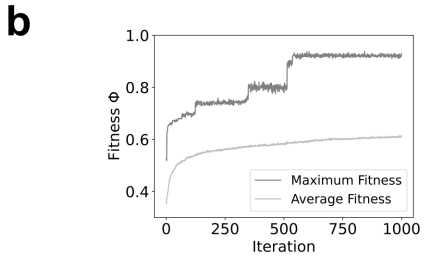
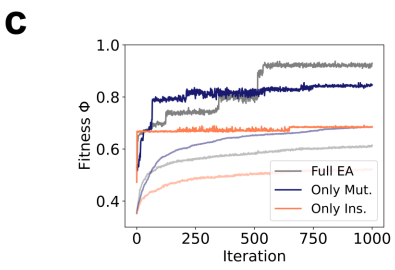
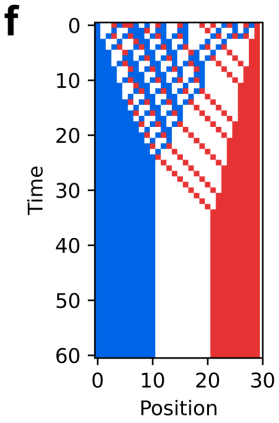


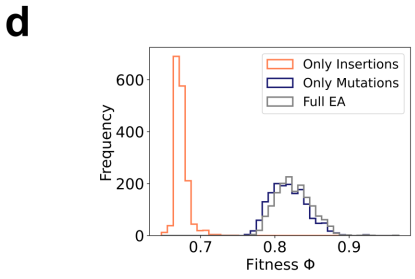
Figure 4.2 2 State Solution (a) Fitness for all 2 state rules trying to generate a half blue, half white pattern from a random initial condition. Fitness was calculated over $N_{Sample} = 1000$ for $L = 100$ and $T_{max} = 200$. (d) Fitness as a function of system size for rule 29. For $L \leq 20$ $N_{Sample} = 100000$, for $20 < L \leq 100$ $N_{Sample} = 10000$, and for $L \geq 100$ $N_{Sample} = 1000$ was used. (c) Exemplary kymograph for rule 29 starting from random initial conditions. (d) Rule table of rule 29, split in mappings necessary for a correct steady state and mappings creating the checkerboard pattern.

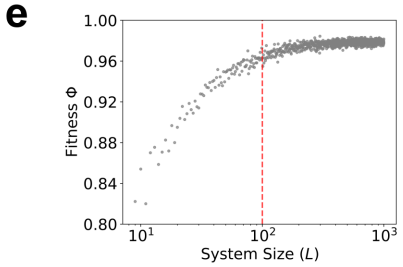

a
Evolutionary algorithm (EA) for exploration of rule space
Rules
fitness
12020120012010...
0.67
22012010201021...
0.54
21002102010220...
0.48
21020102102011...
0.46
21020120121110...
0.35
22012010100012...
0.32
00210201201120...
0.29
02102102110201...
0.28
Mutate
Mutated rules
fitness
12020220012010...
0.87
22012010201121...
0.23
20002102010220...
0.29
21020102112011...
0.56
21000120121110...
0.40
22012020100012...
0.29
00210201201120...
0.55
02102112110201...
0.41
Take best pair-wise
New rules
fitness
12020220012010...
0.87
22012010201021...
0.54
21002102010220...
0.48
21020102112011...
0.56
21000120121110...
0.40
22012010100012...
0.32
00210201201120...
0.55
02102112110201...
0.41
Discard 30% least fit
Complete with random rules
Initiate new round of evolution


b
Fitness Φ
Iteration
Maximum Fitness
Average Fitness


c
Fitness Φ
Iteration
Full EA
Only Mut.
Only Ins.


f
Time
Position


d
Frequency
Fitness Φ
Only Insertions
Only Mutations
Full EA


e
Fitness Φ
System Size (L)

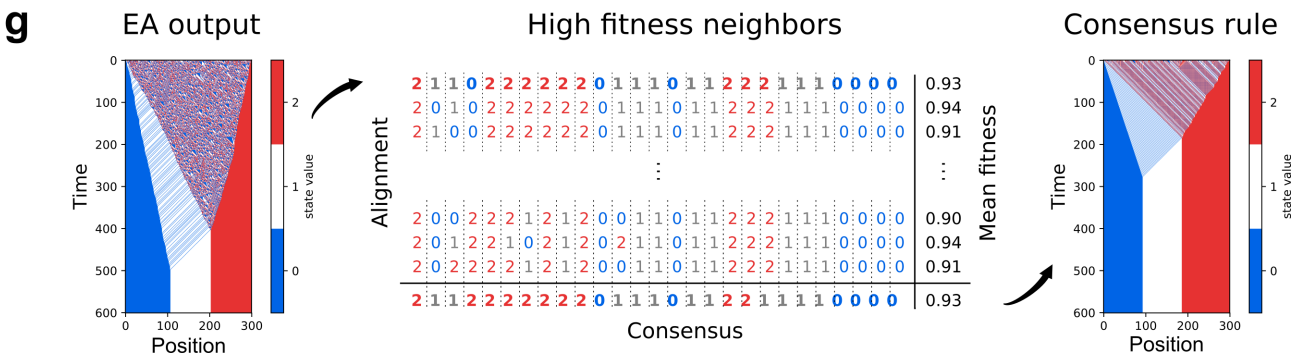

g
EA output
High fitness neighbors
Consensus rule
Time
Position
state value
Alignment
Consensus
Mean fitness

Figure 4.3 Evolutionary Algorithm and Consensus Procedure. (a) The evolutionary algorithms starts with a population of rules and calculates their fitness according to fig. 4.1. It then mutates each rule in the population and calculates the corresponding fitness. The mutated rules are compared their respective parent rules, and out of this pairwise comparison the rule with the higher fitness is chosen. From this new population, the 30% worst rules are eliminated and replaced by random rules and completes a single round of evolution, which gets repeated $N_{iteration} = 1000$ times. (b) Maximum and average fitness of a population of a single EA run as function of the iteration showing the increasingly slower increase with iteration. (c) Comparison of maximum and average fitness as a function of iteration for the evolutionary algorithm as described in (a), as well as versions without mutations or without injection of random rules. The version described in (a) outperforms the other two regarding the maximum fitness present in the population. (d) Distribution of maximum fitness for several evolutionary algorithm runs shows that while in general good solutions are found, for finding exceptional solution several runs may be needed. (e) Fitness as a function of system size for an exemplary best rule at the end of an evolutionary algorithm run. Fitness is not constant with system size, but may change, and is subject to stochastic variations due to the sampling of the initial condition. (f) Exemplary kymographs for said rule. (g) Consensus procedure taking the best rule in the rule set at the end of the evolutionary algorithm run and calculating the fitness for all rules at most 6 mutations distant, selecting the rules with fitness $\Phi > 0.9$, and then constructing a consensus by aligning the rule outputs and taking the majority output for each input.

(fig. 4.3 (a)) starts with a set of $N_{population} = 100$ randomly chosen rules. The fitness of the rules is calculated, each rule is mutated twice and the fitness of the mutated rule is again calculated. Now the fitness of the original and the mutated versions of the rule are compared and the lower performing is discarded. The resulting rule population again comprises of 100 rules, from which the worst 30% are also discarded and replaced by randomly chosen rules. This procedure is now repeated for $N_{Iteration} = 1000$ steps while the fitness of the best rule typically increases (fig. 4.3 (b)), yielding a final set of rules, the best of which are usually high performing (fig. 4.3 (b, d)). Both mutation and random injections are needed to efficiently search the rule space (fig. 4.3 (c, d)) - random injections alone would just be equivalent to random sampling, mutations alone can increase the average fitness of the rule set efficiently but not the maximum (fig. 4.3 (c, d))[2]. Since the interest here is only in the best performing rules, the combination of both mutation and random injection seems to be a sufficiently well performing combination to search the rule space[3]. The best rule found after several runs of the EA can reliably form a French Flag (fig. 4.3 (e, f)). Since it is expected that slight mutations to this rule yield in most cases also well performing rules, additionally the neighborhood of well performing rules in rule space is surveyed. This serves several purposes: First, knowledge of the well performing rules in the neighborhood allows for the construction of a Consensus rule consisting of the most abundant entries in high fitness rules (fig. 4.3 (g)) in accordance with motif search in DNA, RNA, and protein sequences [154]. Second, the evolutionary algorithm is not constructed to sample the local neighborhood of well performing rules systematically and it can be hypothesized that in this neighborhood equally well or better performing rules can be found.

The consensus procedure was carried out to include rules up to 5 mutations away from the rule found in the EA with a fitness of 0.9 or larger. Out of this set of rules, the most abundant rule entries for each input were selected to construct a “Consensus” rule, which can be seen in fig. 4.4 (a), (b). It is not sensitive to the specific choice the number of mutations or fitness threshold for the inclusion of rules as can be seen in table A.2 and is detailed in [146].

[2] It can be hypothesized that this is due to only finding local maxima in the fitness landscape and then being trapped inside them.

[3] Other common motives used in evolutionary algorithms are combinations of several rules, as well as fitness comparisons of the mutated rules with the whole rule set and not just the immediate predecessor rule [153] - both motives have also been explored during the early stages of the project in different combinations with the other motives but did not seem to yield satisfactory results.

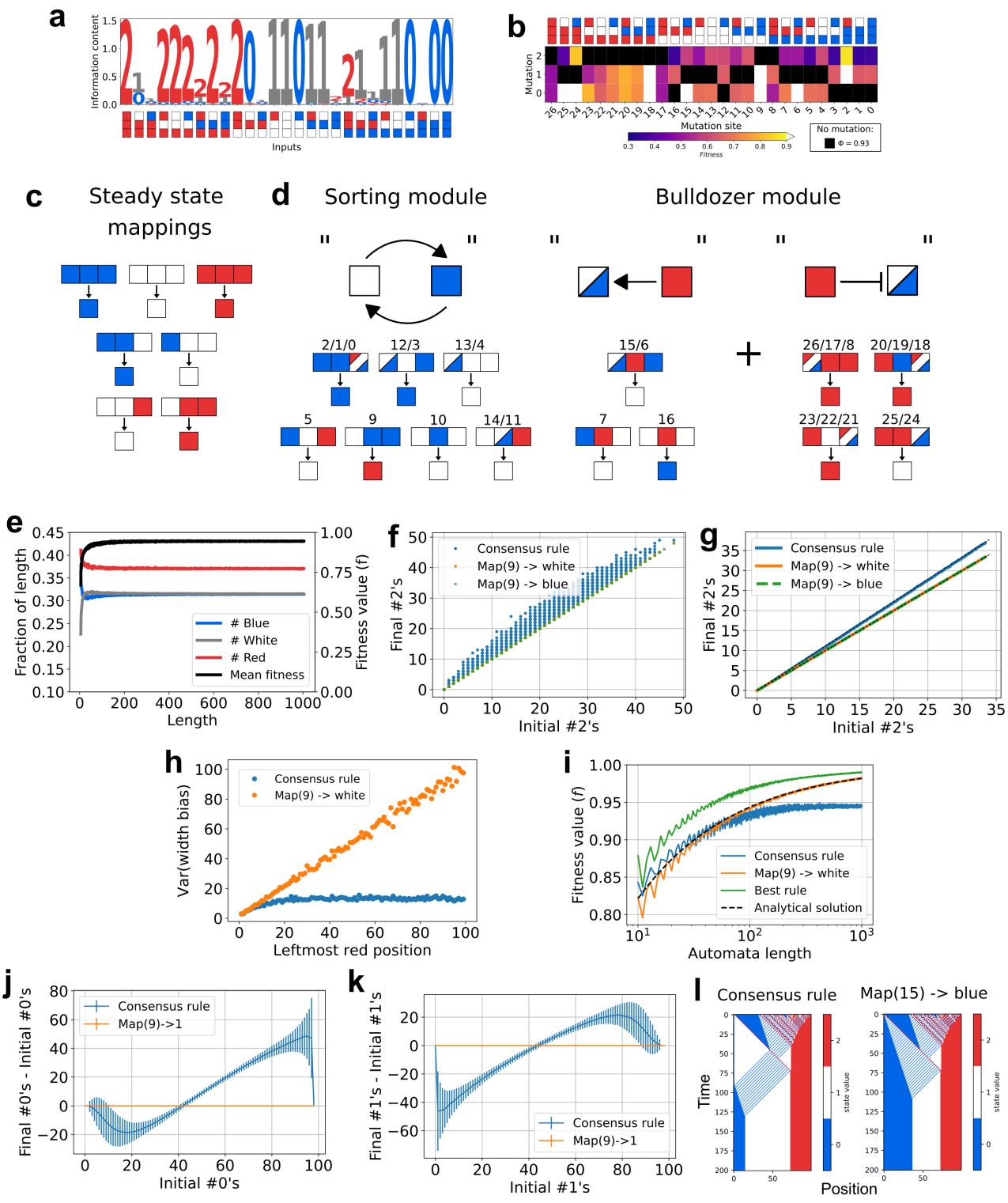

a
Information content
Inputs
b
Mutation
Mutation site
Fitness
No mutation:
Φ = 0.93
c
Steady state mappings
d
Sorting module
Bulldozer module
2/1/0
12/3
13/4
5
9
10
14/11
15/6
7
16
26/17/8
20/19/18
23/22/21
25/24
e
Fraction of length
Fitness value (f)
Length
Blue
White
Red
Mean fitness
f
Final #2's
Initial #2's
Consensus rule
Map(9) -> white
Map(9) -> blue
g
Final #2's
Initial #2's
Consensus rule
Map(9) -> white
Map(9) -> blue
h
Var(width bias)
Leftmost red position
Consensus rule
Map(9) -> white
i
Fitness value (f)
Automata length
Consensus rule
Map(9) -> white
Best rule
Analytical solution
j
Final #0's - Initial #0's
Initial #0's
Consensus rule
Map(9)->1
k
Final #1's - Initial #1's
Initial #1's
Consensus rule
Map(9)->1
l
Consensus rule
Map(15) -> blue
Time
Position
state value

Figure 4.4 Analysis of the Consensus Rule. (a) Logoplot of outputs of the rules with Hamming distance 5 around the Consensus rule which have a fitness $\Phi > 0.9$ showing how variable the entries of high fitness rules are. (b) Fitness of single point mutations of the Consensus rule, generally agreeing with the picture from (a). (c) Stationary state mappings. (d) Rule table of the "Consensus Rule" split into two modules: (1) the "Bulldozing" module consisting of all entries which have a red state in the middle of the input and seed new blue and white states, as well as all entries with red in the left input, in general erasing the middle entry and replacing it with red (2) and the "Sorting" module consisting of the rest of the entries leading effectively to an exchange of blue and white states. (e) Fraction of blue, white and red states in the final pattern as a function of system size. (f) Scatter plot of initial and final number of red states (2s) of the Consensus rule as well as the Consensus rule mutated at input 9 to blue or white, which lead to the conservation of the number of red states in the pattern. (g) As (f), but ensemble averages. (h) Variance of the final number of white and blue states as a function of the leftmost red position in the initial condition of the Consensus rule and the Consensus rule with mapping 9 mutated to white. Not having this mutation as in the Consensus rule can generate an additional red bulldozer far to the left which guarantees a lower variance in the blue and white proportion. (i) Fitness as a function of the system size for the Consensus rule, the best found rule, the mapping 9 to white mutation of the Consensus rule and the analytical solution [146]. (j, k) Difference between the final and initial number of 0s (1s) as a function of the initial number of 0s (1s) for the Consensus rule and the Mapping 9 to white modification of the Consensus rule showing the Conservation achieved by the mutation. (l) Kymographs of the Consensus rule and the Consensus rule with mapping 15 mutated to blue.

4.4 Solutions for 3 State CA

4.4.1 Importance of Contributions

By looking at the rule, it is not clear which (combination of) rule entries are important in the sense that mutating them would lead a rule with considerably lower fitness. Knowing the well working rules in the local neighborhood allows for selection of the relevant rule entries. Thus, the fitness for all rules up to Hamming distance 5 around the Consensus rule is calculated and all rules with fitness $\Phi > 0.9$ are selected. Then, the frequencies of each output in the input-output $(x_{i-1}^t, x_i^t, x_{i+1}^t) \to x_i^{t+1}$ map in this set are viewed as a probability $p_{x_i^{t+1}}^{(x_{i-1}^t, x_i^t, x_{i+1}^t)}$ and the associated entropy

$$S_{(x_{i-1}^t, x_i^t, x_{i+1}^t)} = -\sum_{x_i^{t+1}=0}^{2} \log_2\left(p_{x_i^{t+1}}^{(x_{i-1}^t, x_i^t, x_{i+1}^t)}\right) p_{x_i^{t+1}}^{(x_{i-1}^t, x_i^t, x_{i+1}^t)} \tag{4.2}$$

for the distribution of each rule contribution is calculated. While the entropy can be interpreted as the amount of information gained from determining the exact entry instead of a choice from the distribution of the associated random variable, subtracting the entropy from the maximum entropy gives the information content

$$I_{(x_{i-1}^t, x_i^t, x_{i+1}^t)} = \log_2(3) - S_{(x_{i-1}^t, x_i^t, x_{i+1}^t)} \tag{4.3}$$

plotted in fig. 4.4 (a) in form of a logo plot. A set of contributions can be immediately singled out, the steady state mappings (fig. 4.4 (c)) which map all inputs that can occur in a successfully formed French Flag on the state of the middle input. These mappings guarantee that the the French Flag is a steady state of the dynamics generated by the rule. It is intuitive that these contributions have a large information content in contrast to other contributions, namely $2, 6, 9, 15, 18, 20, 24$ and 25 (enumeration as in fig. 4.4 (b, d)), which are highly variable and can be chosen almost at will without leaving the realm of the high fitness neighborhood. This is also confirmed by the changes of the fitness by single mutations (fig. 4.4 (b)), which show large negative changes for the steady state mappings, but only comparably small changes for the contributions identified as highly variable.

4.4.2 Patterning Modules

By close inspection, the rule can be split up in two parts termed modules whose distinct dynamical behaviors can be characterized (fig. 4.4 (d)). The first module is termed "sorting module" and is comprised of all

inputs which have a no red state in the middle or left part of the input of the mapping. Roughly speaking, this module exchanges any blue cell that is to the right of a white cell, sorting the blue and white states into the correct order. The second module is the "bulldozer module", consisting of all cells with a red cell in the middle or left input. It moves red cells to the right if there is no other red cell and leave blue and white cells behind in an alternating fashion. Both modules work together to construct the French Flag. The bulldozer module shifts all initially red states to the right, while the sorting module sorts any left behind blue and white states as well as all blue and white states to the left of the leftmost red state in the correct order. Since the number of red states in the initial condition is on average $\frac{1}{3}$ and the blue and white states are left behind in equal numbers, the resulting pattern can be expected to be close to a French Flag.

This high level description of the hypothetical rule dynamics of course needs to be confirmed by computer experiments. It allows to formulate predictions about the expected behavior of the rule:

1. The initial number of red states should be conserved and thus the same as in the final French Flag.
2. For an initial condition without red states, the number of blue and white states should also be perfectly conserved since they are only sorted.
3. The fitness should converge towards 1 for large system sizes L, $\lim_{L\to\infty} \Phi\left(L\right) = 1$.

Checking the conservation of red states (fig. 4.4 (e), (f)), it is clear that strict conservation does not hold. Additionally, the number of red cells in the final pattern is larger than the number of blue and white cells (fig. 4.4 (g)). Inspecting the sorting module of the Consensus rule (fig. 4.4 (d)), it can be hypothesized that entry 9 produces additional red states. This hypothesis is verified by mutating this entry to either blue or white, in which case the numerical exploration shows a perfect conservation of red states (fig. 4.4 (e), (f)). It can be speculated why a red entry 9 is more prevalent in the found high scoring rules used to construct the Consensus rule. Having this entry potentially produces additional red states close to the left boundary, which in turn reduces the "stochastic component" in the number of blue and whites in the final pattern since the blue and whites left of the first red state keep their distribution when being sorted into the final pattern. This can be quantified by first defining the difference between the number of blue and white states in the final pattern as variable and then calculating the variance of that variable as a function of the position of the leftmost red in the initial condition (fig. 4.4 (h)). If entry 9 is mutated this scales roughly linearly as expected for the difference between two independent binomial random variables. If it is not mutated, the variance approaches a constant value determined by the average position of the leftmost red state created by entry 9.

Conservation of the number of white and blue states for white and blue only initial condition does also not hold true (fig. 4.4 (i), (j)), again solely be because of entry 9 (fig. 4.4 (i), (j)).

The convergence of the fitness to 1 does also not hold (fig. 4.4 (k)) exactly. Assuming a uniform distribution of all rule inputs in the initial state and no new creation of inputs 9 during the rest of the simulation, a fraction of $\frac{1}{3} + \frac{1}{27}$ reds and $\frac{1}{3} - \frac{1}{27}$ blues can be expected (fig. 4.4 (g)), which explains the missing convergence to 1. An analytical model constructed from the above hypothesized assumptions about the dynamics imposed by the Consensus rule supplemented by the effect of mapping 9 constructed by G. Vercelli predicts for $1 - \Phi$ a scaling with system size proportional to $L^{\frac{1}{2}}$ and a limit of $\lim_{L\to\infty} \left(1 - \Phi\left(L\right)\right) = 1 - \frac{1}{18}$ [146], which fits the numerical results (fig. 4.4 (j)). Mutation of mapping 9 leads to the prediction of a limit of 1, which is also confirmed by numerical experiments (fig. 4.4 (k)) [146].

As a final note, mapping 15 would be expected to map to blue instead of white for a perfect bulldozing module. However, mapping 15 influences the final pattern primarily when there is a large blue patch to the right of the leftmost red cell, which is unlikely for random initial condition (fig. 4.4 (l)). Thus this mapping does not effectively influence pattern formation.

4.4.3 Coverage of the Rulespace

The evolutionary algorithm and following consensus procedure might be successful in finding solutions to the French Flag problem, but there is inherently no guarantee that all qualitatively different types of

solutions are found with this procedure. For a single automaton length and a single target pattern, it is in computational reach to calculate the fitness of all 3 state rules by using an adaptive fitness calculation algorithm. In contrast to the normal algorithm, the fitness of a rule is only evaluated for $N_{Sample} = 10$ different initial conditions for $L = 30$, and only when the fitness calculated in this way exceeds 0.66666, additionally 40 initial conditions are sampled for a total of $N_{Sample} = 50$ initial conditions. To conserve memory, only rules with fitness exceeding 0.85 are finally used. For visualization purposes, a tSNE algorithm [155, 156, 157] preceded by a principle component analysis of a 27 dimensional vector containing the rule outputs is used to project the rules pace down to 2 dimensions (fig. 4.5). To check whether the same modules as in the Consensus rule are used, the rules are colored by their relative agreement to the relevant parts of these modules, where relevant parts is defined as having an information content that is at most one bit less than the maximum information content (fig. 4.5 (a)-(d)). It shows that the relevant parts of the sorting and bulldozing module are prevalent in nearly all of the found rules with high fitness (fig. 4.5 (e)) either in the red bulldozer and blue-white sorting or in the blue bulldozer and white-red sorting combination, with the exception of some rules on the top. These rules have, in contrast to all other rules, not all entries necessary to yield the French Flag steady state which is also shown by coloring the rules by their measured time to steady state (fig. 4.5 (g)) and direct visualization of the kymographs (fig. 4.5 (k)), and therefore are indeed an artefact of the chosen simulation time when calculating the fitness. The crosses illustrate the prototypical Consensus rule (fig. 4.5 (h)) and its color symmetric counterpart (fig. 4.5 (i)).

4.5 Solutions for 4 State CAs

In this part, the combinations of the modules found in 3 state space in 4 state space as proposed by G. Vercelli are explored. The fourth state is labeled black in the following.

The simplest way to generalize the Consensus rule to 4 states is to just take the entries of the 3 state rule for all inputs that do not have a black state, let the bulldozer state also erase the black state, and evenly distribute the rule outputs over the blue, white and red colors for inputs that contain a black state (fig. 4.6 left column). Additionally, the entries that let the Consensus rule diverge from the simple model of a pure red bulldozer and a pure blue-white sorting like for example rule entry 9 are now set such that the simple model assumptions of red bulldozing and blue-white sorting are fulfilled (fig. A.3). This rule will be called Mixed rule for purposes which will get clear later. In itself, the construction of the Mixed rule is not particularly interesting because it behaves essentially like the Consensus rule (fig. 4.7 first and second row), but it can serve as a comparison and benchmark for the following constructions.

With the additional fourth state, the bulldozing state which erases the initial pattern and generates the rest of the states with equal proportion can be shifted on this black state, while all other states just sort themselves starting from this states generated by the bulldozing state (fig. 4.6 middle column, fig. A.3). Intuitively, this has the advantage that the relative proportion of the red state does only weakly depend on the initial condition, namely on the position of the leftmost black state and the proportion of reds left to the leftmost black state (fig. 4.7 third row).

Finally, it can be attempted to construct a rule that purely sorts blue, white and red states and converts all the black states in the initial condition immediately into blue, white or red states. A rule working in this way would, for the width of the stripes of the French Flag, solely rely on the statistics of the initial condition. The problem in attempting to do this is that with 3 states it is unclear how to process the reverse French Flag (red, white, blue) input, because red and blue need to be swapped to the same place and on the neighboring tiles two white states are created (fig. 4.8 a). Having a fourth state allows to solve this problem by using the fourth's state as a temporal superposition of blue and red state that also gets sorted and resolved as soon as the states around it are in a suitable condition (fig. 4.6 right column, fig. A.3, fig. 4.8 (c) and (d)). The module that performs this action is termed resolution module (fig. 4.8 (b)). The rule itself is termed Bubble rule because of its resemblance of the equally named parallel sorting algorithm [158].

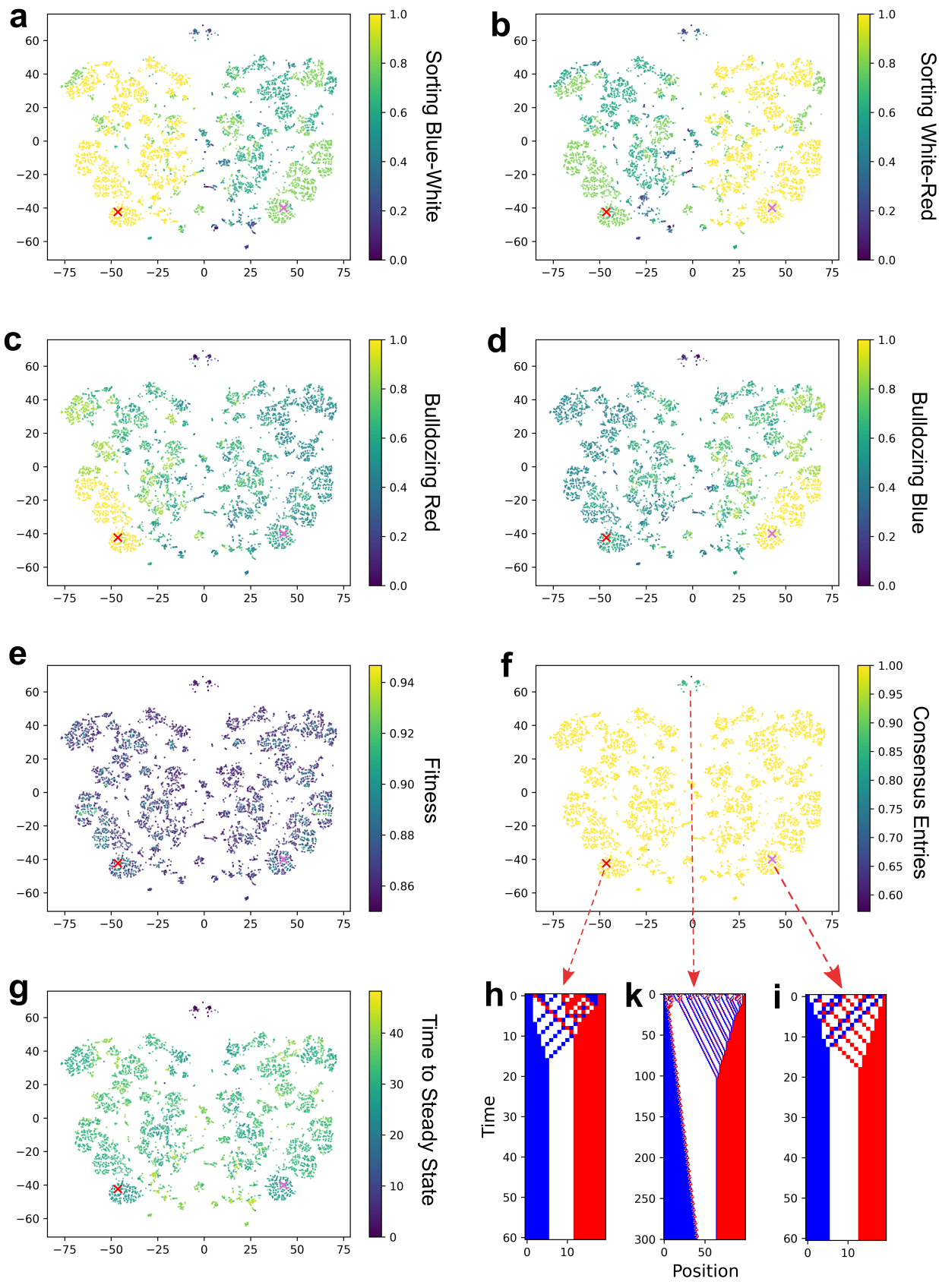

a
Sorting Blue-White
b
Sorting White-Red
c
Bulldozing Red
d
Bulldozing Blue
e
Fitness
f
Consensus Entries
g
Time to Steady State
h
k
i
Time
Position

Figure 4.5 Screening of Entire Rule-space. For all rules in 3 state rule-space, the fitness is calculated, all rules with a fitness $\Phi > 0.85$ are selected and for visualisation purposes projected down from the 27 dimensional genotype space to a 2 dimensional space using first principle component analysis followed by t-distributed stochastic neighborhood embedding. The colors represent relative amount of relevant entries (entries with an information content larger $log_2(3) - 1$) with the same output as the Consensus rule for a (a) blue white sorting module (b) white-red sorting module (c) red bulldozer module (d) blue bulldozer module; (e) relative amount of steady state entries (f) fitness Φ (g) time to steady state. Example kymographs for (h) Consensus rule (i) its symmetric counterpart (k) and an example of the small batch of rules not reaching a steady state during simulation time.

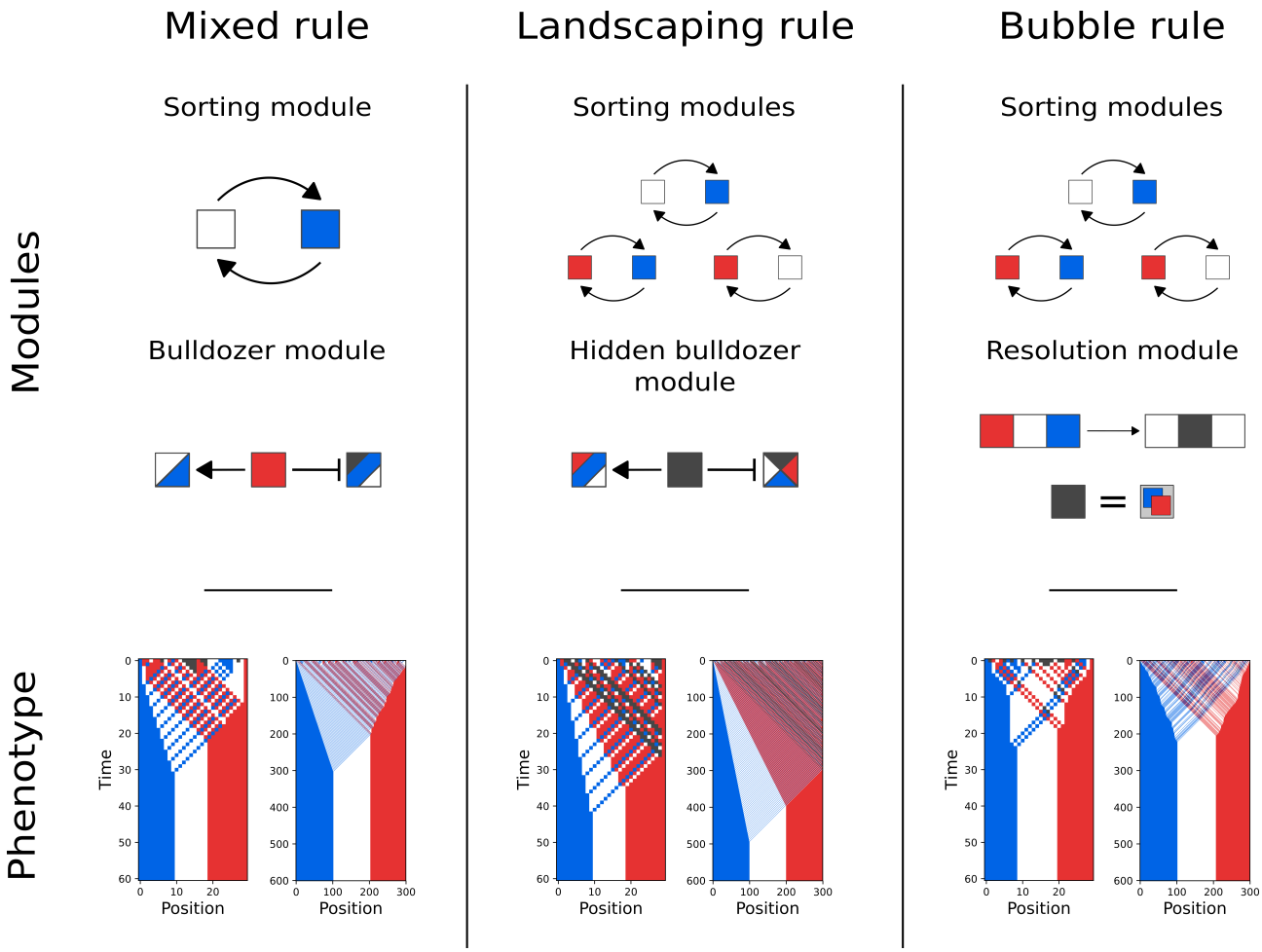


Figure 4.6 Construction of Rules in 4 State Space. High level models of rules constructed in 4 state space, for kymographs and detailed mapping see fig. 4.7 and fig. A.3. The Mixed rule is the 4 state extension of the Consensus rule working similarly but now with the red state also erasing all black states. Landscaping rule uses the black state to erase the preexisting pattern and seed blue, white, and red states alternatingly, which subsequently get sorted. The Bubble rule sorts the initial pattern using a resolution module (fig. 4.8) to resolve inverse French Flag inputs.

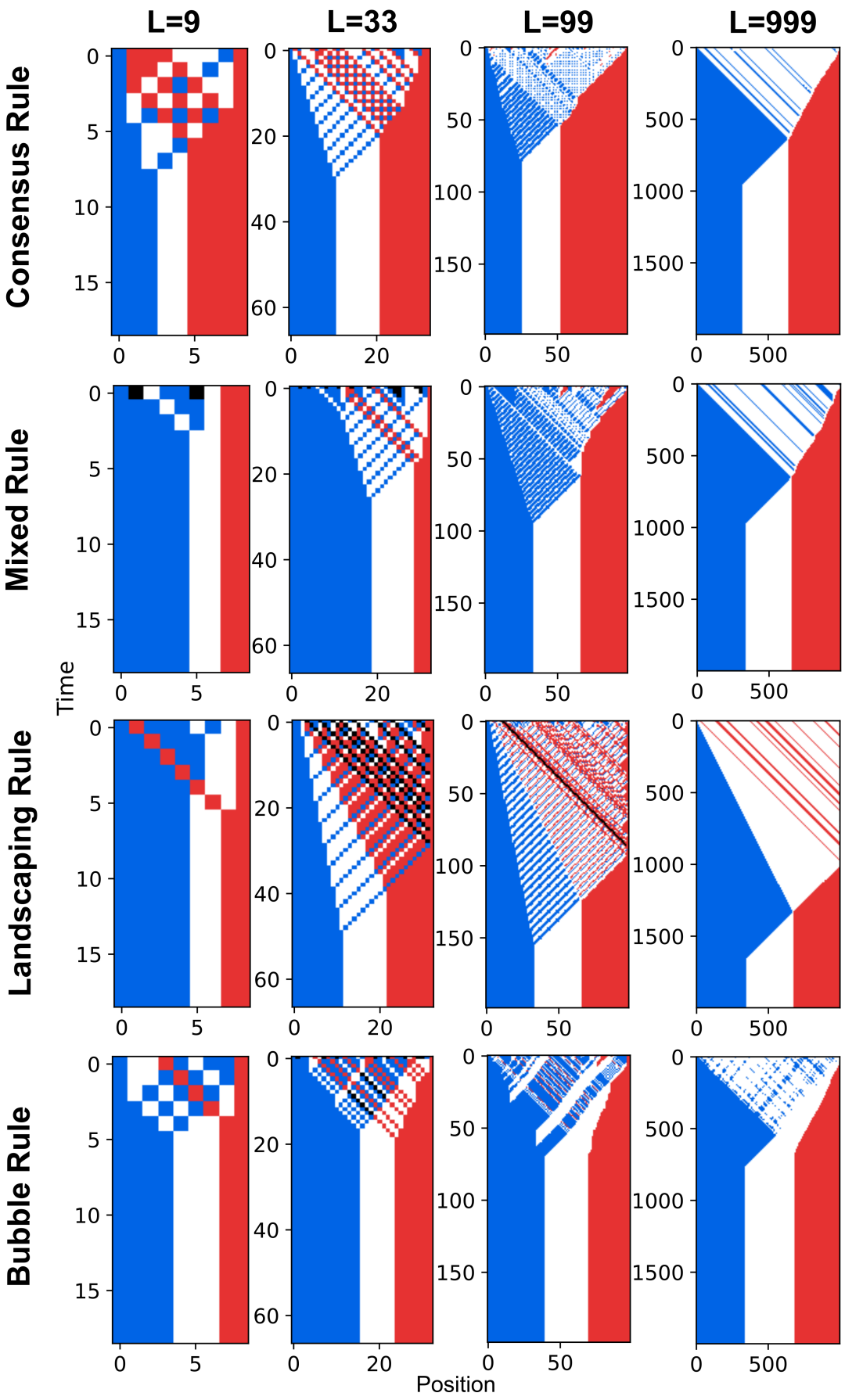


Figure 4.7 Kymographs for Consensus, Mixed, Landscaping, and Bubble Rule for Different System Sizes.

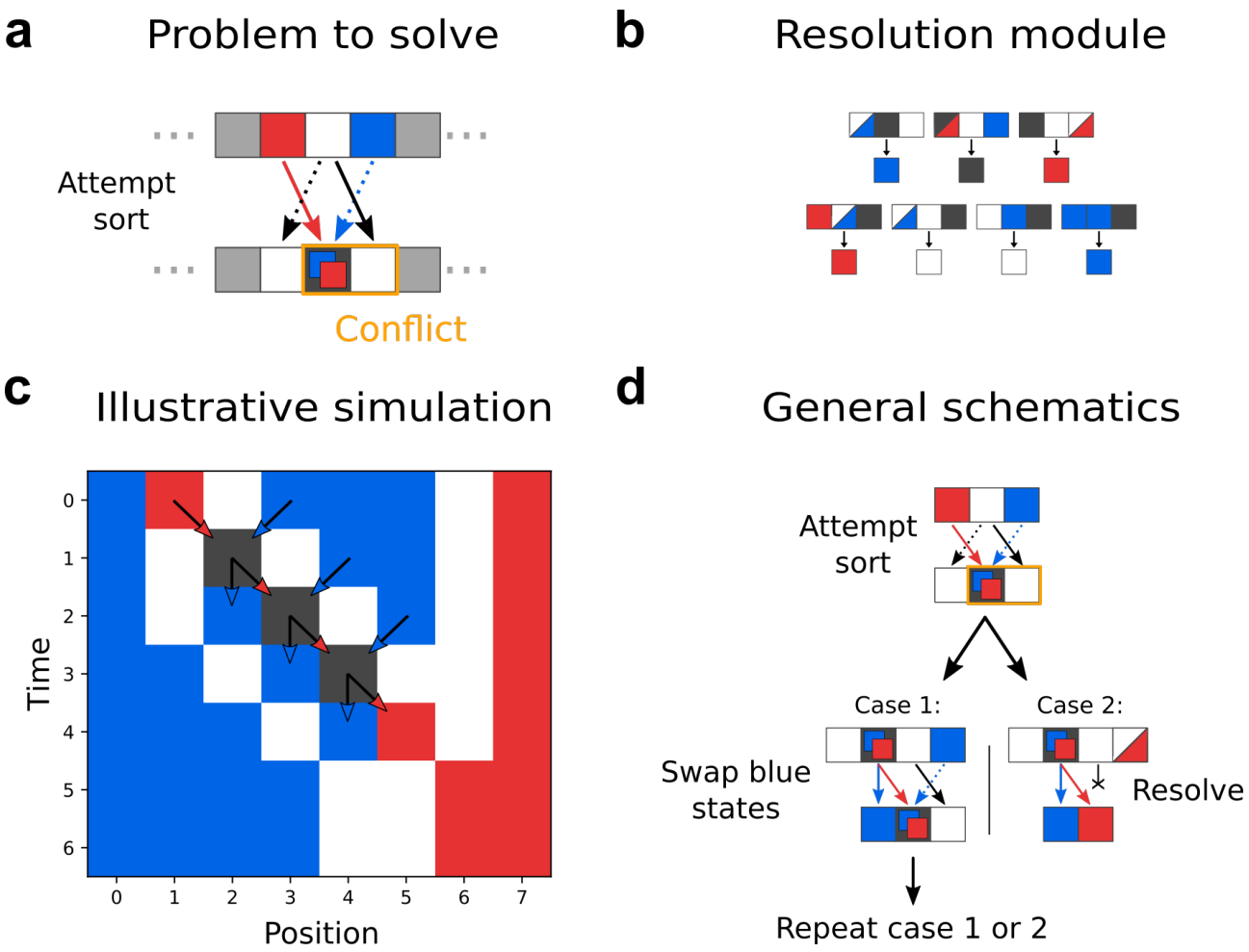


Figure 4.8 Resolution Module (a) The resolution module solves the problem to which state the inverse French Flag input should be mapped to. (b) Mappings of the resolution module (c, d) Illustrative simulation and general scheme show how the resolution module resolves the inverse French Flag. It maps it to a black state creating one black and one white state in the process. The black state moves to the right until there is no blue state any more 2 cells to its right and then annihilates with a white state creating a blue and red state in the process.

4.6 Scaling of the Fitness Function

One of the most notable conditions in the French Flag problem is that the pattern is scale-able, i.e., it adapts to the system size. While it is intuitive from the simplified models stated above describing how the rules work that the functioning should be independent of the system size, that the actual rule and not only the simplified conceptual model need to be shown explicitly. Indeed, numerically calculating the fitness for Consensus, Mixed, Landscaping and Bubble rule over a range of system sizes shows that in all cases the fitness is large and slightly increasing with system size (fig. 4.9). The curves exhibit a high frequency oscillations due to the discreteness of the system which only allows proportions of exactly 1:1:1 if the system size is divide-able by 3. The curves can be fitted well in a log-log plot (fig. 4.9 inset), resulting in linear scaling coefficients of roughly -0.5 for the Mixed and Bubble rule and -1 for the Landscaping rule. A detailed mathematical model developed by G. Vercelli [146] shows that it is possible to predict these exponents using the simplified conceptual models of sorting and bulldozing together with a uniformly state distribution in the initial condition. In a nutshell, since both Mixed and Bubble rule rely on (for at least a part of the stripes) purely sorting the initial condition, the fitness difference to 1 is dominated by the absolute difference of the number of states in the initial condition from the average $1/3L$. As shown in the publication, this difference scales as the inverse square root of the system size, reproducing the fitted coefficients. The Landscaping rule on the other hand uses the bulldozer state to completely erase the pattern and perfectly reconstruct it, thus all fitness deficits originate from the part of the initial condition left of the leftmost bulldozer state in the initial condition. The size of this part is roughly independent of the system size, i.e., its relative size shrinks inversely proportional to the system size. Thus, it creates a fitness deficit scaling roughly inversely proportional to the system size explaining the measured coefficient of -1.

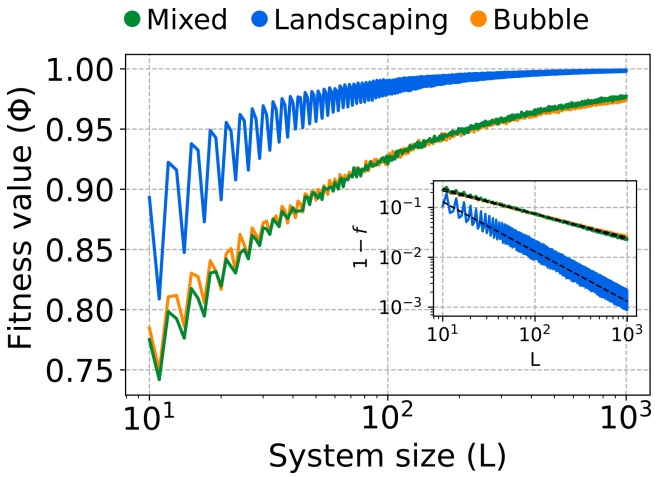


Figure 4.9 Scaling of the Fitness Function. Fitness as a function of the system size for Mixed, Landscaping and Bubble rule. High frequency oscillations are due to the non unique definition of the target pattern if the system size is not divisible by 3. The Landscaping rule outperforms Bubble and Mixed rule, which have a similar graph. Inset: 1 - Φ as a function of the system size in a double logarithmic plot together with linear fit functions illustrating the power law scaling of the fitness function when approaching 1 in the limit of large systems.

4.7 Kinetics of Pattern Formation

In developmental systems, the time to form a pattern until the next step of development is often rather limited [146]. Thus, a fast paced pattern formation mechanism can be imperative. A measure for the speed of pattern formation is the average time it takes until the pattern reaches its steady state T_{SS}, expressed as a function of the system size. Computationally measuring this parameter and using a linear regression to fit T_{SS} yields linear scaling coefficients of 0.74 for the Bubble rule, 1 for the Mixed rule and

1.67 for the Landscaping rule fig. 4.10. The numbers have again also been derived using mathematical models based on the simplified conceptual models of these rules [146]. The Bubble rule just sorts the initial condition, which for large system sizes should for the furthest states should naively take at most $2/3L$ steps to reach their respective states - yet the resolution model delays the sorting with each of its applications by a step, thus increasing this number slightly to 0.74. The Mixed rule first needs time to sort all red states to the right $2/3L$ and then sorts the last created blue state from the $2/3L$ position to the $1/3L$ position, in total $2/3L + 1/3L = L$ steps. For the Landscaping rule, this increases to L steps for the bulldozer state to move completely to the right and then subsequently $2/3L$ steps for the last created blue state to reach the blue stripe, yielding $L + 2/3L = 5/3L$ steps necessary.

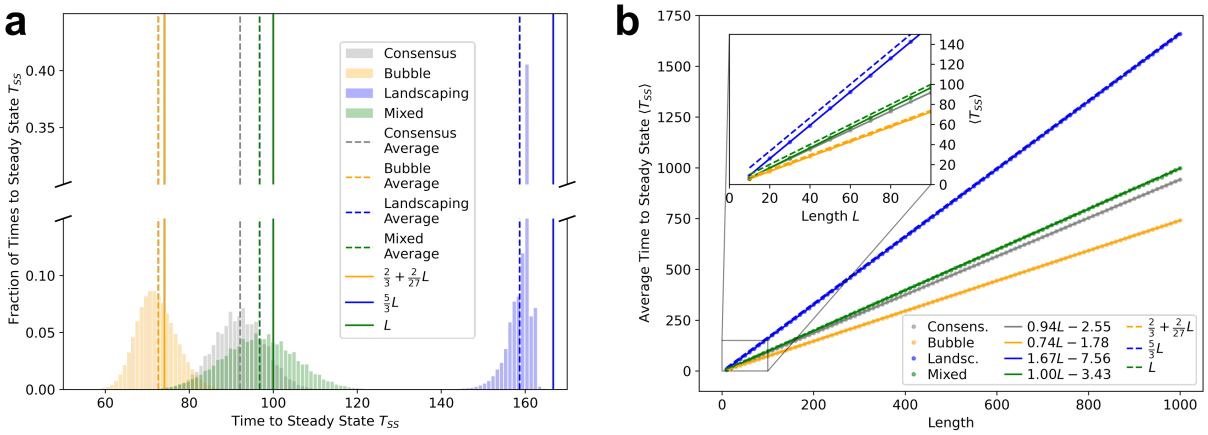


Figure 4.10 Time to Steady State. (a) Distribution of the time to steady state for Consensus, Mixed, Bubble and Landscaping rules, together with their averages and theoretical values for a fixed system size. (b) Scaling of the time to steady state as a function of system size for the prototypical rules together with linear fits and theoretical estimates.

4.8 Leaving the perfect world: Noise, Growth and Asynchrony

So far, the setting of the model was deterministic, except for the varying initial conditions, and the number of cells was constant.

4.8.1 Noise

In more realistic scenarios like developing organisms or synthetic signaling, noise is usually ubiquitous [140]. It may influence transmission, processing and storing of the information about the states. Here, noise will be modeled by a probability p that during a CA update of a single cell the dynamics will not follow the CA rule but instead choose one of the states at random. This single parameter models noise during the transmission of information, the processing of the received and stored information, or during the storage of information itself. Figure 4.10 (a) gives a visual impression of how pattern formation with Mixed, Landscaping and Bubble rule changes in the presence of noise for different noise strength. The Bubble rule seems a lot less affected than the Mixed and Landscaping rule, from which the latter seems to perform worst. Intuitively, this can be understood as a creation of a bulldozer state through noise far left in the pattern basically restarts pattern formation nearly from scratch - for large enough noise p the average time between establishing two new bulldozer state through noise is then smaller than the time it takes for the pattern to form, never ending the pattern formation effectively. For the Bubble rule on the other hand, the general pattern is not disturbed too much by the noise since individual errors just get sorted, but do not “spread” throughout the pattern. Note that a width of $L/3$ for each stripe is a steady state under noise

for the Bubble rule assuming that it just sorts: If a stripe has currently length l and there are 4 states in total, i.e., 3 flag states (the black state gets changed to the others relatively quickly), on average $lp\frac{2}{3}$ of the mutations in the stripe change their state to a different one, while on average $(L-l)p\frac{1}{3}$ change to this state in the other stripes - equating these two processes results in $l = \frac{L}{3}$. To quantify the stability of the pattern formation process, in fig. 4.10 (b) the average boundary positions as a function of the noise are shown, where boundary position is defined as the first occurrence (looking from the left) of a consecutive triple of cells with at least two cells in a state of the corresponding stripe, confirming that the Bubble rule indeed is stable against noise in contrast to Mixed and Landscaping rule.

4.8.2 Growth

In development, pattern formation often occurs during growth of the tissue itself. To check whether Mixed, Landscaping and Bubble rule are able to form the pattern during growth, a simple homogeneous growth is introduced: every cell has a probability q of dividing itself per step ("growth rate per cell"), with both daughter cells adopting the same state as the mother. In fig. 4.10 (c) again kymographs for different growths rates per cell are depicted, showing that for low growth rates per cell, patterns form reasonably well, while for larger growth rates per cell the patterns get distorted. To quantify this more, the average fitness as a function of the initial system size for a growth rate per cell of 0.002 is shown in fig. 4.10 (d). This is used as a measure since the average number of new cells per time step is qL, i.e., increasing either the initial size or the growth rate both increases the total growth rate. For smaller initial sizes, the fitness increases, while for larger sizes it starts to decrease again. This can be explained by the combination of two realizations: A steady state pattern is, for short times, only mildly affected by growth since if the stripes roughly have a relative width of $1/3$ each, each of the grows with the same rate. Once the total growth rate exceeds roughly one cell per time step, growth happens faster than pattern formation, and thus the latter cannot succeed if it is not finished yet. To test this explanation, the initial length at which the exponential growth is faster than the pattern formation L_{crit} is estimated using $qT_{SS}L_{crit} = 1$, which approximately marks the length at which the fitness starts to dip again. The exception is the Landscaping rule, which creates new bulldozer states for the input $(2, 1, 0)$ (reverse French Flag) - a input that does usually not appear during pattern formation behind the last bulldozer, but can due to the system growth (fig. A.7 and fig. 4.10 (c)). Mutating this entry to a non bulldozer state changes the form of the curve to be similar to the curves of the other rules (fig. A.4 (f) and fig. A.7). In summary, all prototypical rules are able to form patterns during growth, with their ability decreasing the longer the pattern formation takes favouring the Bubble rule over bulldozer module based rules.

4.8.3 Asynchrony

A central assumption of our model is the synchronous update of all cells. While there are examples of pattern formation where the expression of certain genes is synchronized due to local interactions [29], there is in general no guarantee that cell state changes happen synchronously across different cells. To test how much an asynchronous update influences pattern formation with the prototypical rules, the probability of a cell to perform an asynchronous update per time step p_A is defined. At each time step, a set of cell is chosen according to this probability to update asynchronously. All other cells first perform a synchronous update, then the cells of the asynchronous set update in a randomly chosen order. The fitness as a function of p_A is plotted in fig. 4.10 (e), showing that Mixed and Bubble rule outperform Landscaping rule and generate pattern with a relatively high fitness even when on average 30% of the cells update asynchronously. The low performance of the Landscaping rule can partially be explained by entry 36, which is the only entry generating new black states without black inputs. Under synchronous operation of the rule, input 36 does not appear left of the leftmost black state after the initial condition is processed, but the asynchronous update can generate this input and thus new black cells, which restart the patterning process (fig. 4.10 (f)). Mutating said entry to white improves the fitness significantly (fig. 4.10 (e)). In conclusion, the Mixed and Bubble rule can withstand a significant portion of asynchronous update, while the Landscaping rules needs slight modifications.

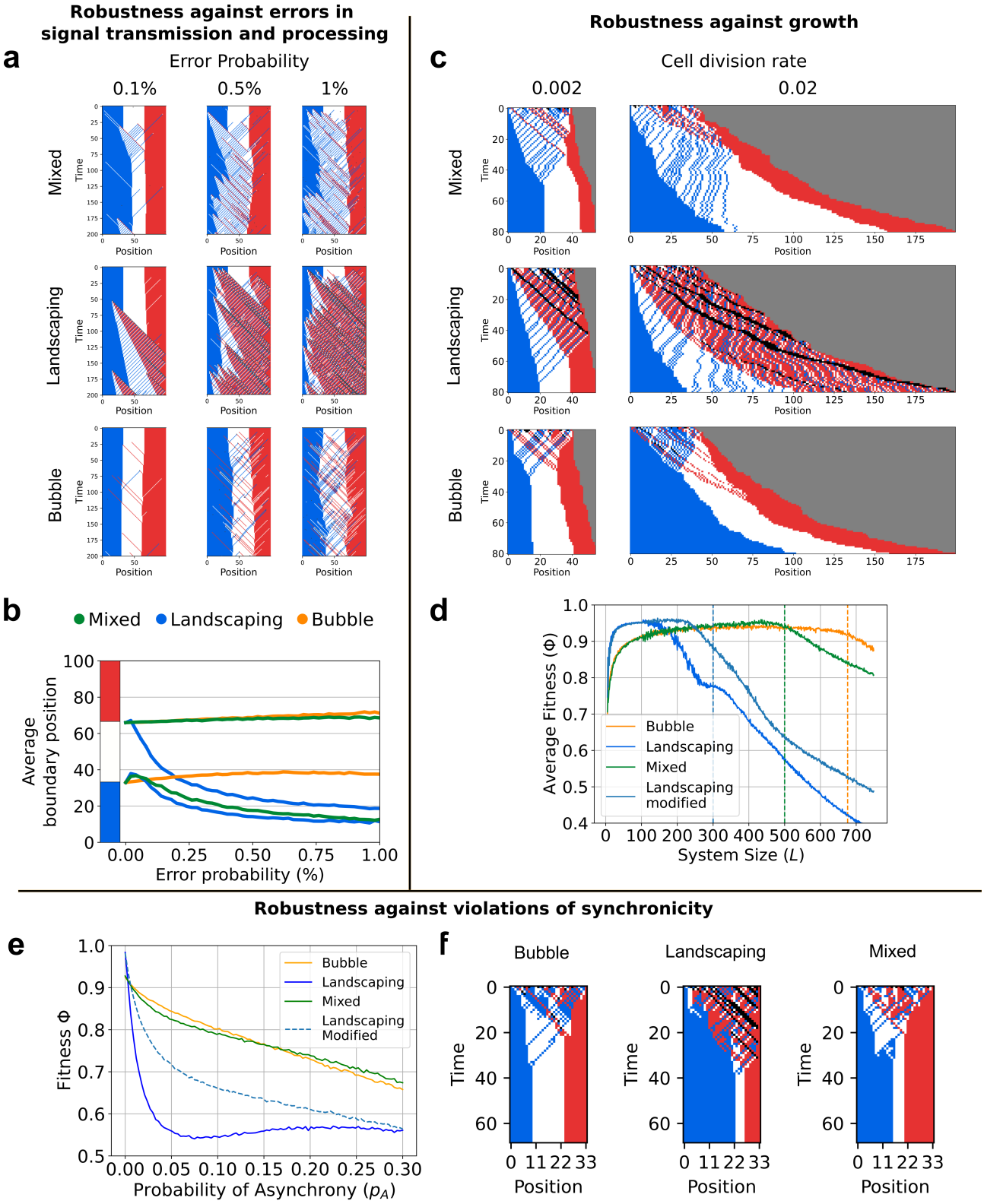

Robustness against errors in signal transmission and processing
a
Error Probability
0.1%
0.5%
1%
Mixed
Landscaping
Bubble
Time
Position
b
Mixed
Landscaping
Bubble
Average boundary position
Error probability (%)
Robustness against growth
c
Cell division rate
0.002
0.02
d
Average Fitness (Φ)
System Size (L)
Bubble
Landscaping
Mixed
Landscaping modified
Robustness against violations of synchronicity
e
Fitness Φ
Probability of Asynchrony (p_A)
Landscaping Modified
f
Bubble
Landscaping
Mixed
Time
Position

Figure 4.10 Noise and Growth (previous page). (a) Kymographs for the dynamics of Mixed, Landscaping and Bubble rule for different probabilities of mutating the states of an individual cell to a wrong state at each time step. Generation of new bulldozer modules (Mixed and Landscaping rule) drastically impacts the pattern formation, while the effects of errors in the dynamics of the Bubble rule only have local effects. (b) Average boundary position, defined as the first occurrence (looking from the left) of a consecutive triple of cells with at least two cells in a state of the corresponding stripe, as a function of the error probability underpinning the stability of the Bubble rule. (c) Kymographs for the dynamics of Mixed, Landscaping and Bubble rule for different rates for each cell to divide at each time step. (d) Fitness as a function of system size for Mixed, Landscaping and Bubble rule together with estimates of the system size which on average has one cell division per time step in the whole grid. (e) Fitness as a function of the probability of asynchrony for Bubble, Landscaping and Mixed rule as well as the Landscaping rule modified at position 36 away from black. (f) Exemplary kymographs for pattern formation with asynchrony for $p_A = 0.1$ and $L = 34$.

4.9 Changing the Target Pattern

While the French Flag serves its purpose as a conceptual target, its symmetry with regard to stripe width raises the question if the techniques used to find solutions to the French Flag problem as well as the solutions themselves are generalizeable to different but related problems. In this section, first the evolutionary algorithm and consensus procedure are used to find solutions for different target ratios. Second, it is shown that re-engineering the Bubble rule allows for nearly continuous adjustment of the stripe within a range. Third, using the patterning modules of Landscaping and Mixed rule, solutions for flags with more than three stripes are constructed.

4.9.1 Using Evolutionary Algorithms and Consensus Procedure to Search for Rules Creating Different Target Patterns

The evolutionary algorithm and consensus procedure is in principle agnostic to the target patterns which is specified by choosing the target in the fitness function. To look for target patterns with a ratio of blue:white:red equals (1:2:1), (2:1:1) or (3:1:1), respectively, the target pattern was set accordingly. Setting the target pattern to different ratios, in this cases blue:white:red equals (1:2:1), (2:1:1) or (3:1:1), respectively, and modifying the consensus procedure to a smaller coverage radius of 3. Since the found solutions were riddled with special solutions for the used simulation time $T_{max} = 2L$ which did not reach the steady state yet, the fitness function was adapted to average comparisons after $T_{max} = 2L$ and $T_{max} = 5L$, as well as looking for solutions for different lengths ($L \in \{40, 100, 200\}$). From several runs of EA and consensus procedure, the best solutions were picked and are displayed in fig. 4.11. Visually inspecting the kymographs (fig. 4.11 middle column), all rules seem to use variants of the sorting module, with solutions to (1:2:1) and (3:1:1) sorting red and white instead of blue and white. Using special initial conditions (fig. 4.11 right column), it is possible to shine a bit more light on the additional mechanism of each solution. In case of the 1:2:1 rule, a single blue cells seems the be used as bulldozer producing red and white cells at approximately 1:2 ratio. For the 3:1:1 ratio, the blue bulldozer is moving slower than 1 cell per time step across the pattern while producing white and red at a 1:1 ratio, while for 2:1:1 the bulldozer seems to act in a more stochastic manner.

4.9.2 Making the French Navy Happy: Re-engineering the Bubble Rule that Create French Flags with Different Scales

When sailing at sea, the French navy is faced with the problem of a fluttering flag, which when having originally 1:1:1 ratio will appear distorted viewed from a distance. To counter this problem, it chose a ratio of 30:33:37 for the flag colors giving it a more even appearance when dancing in the wind [159]. Taking this as a model problem, it can be asked if it is possible to modify our already found solutions in a fine grained manner to achieve precisely tune target ratios. In order to solve this problem, the Bubble rule is reinspected here. When being fed with an initial condition which contains only blue, white and red states,

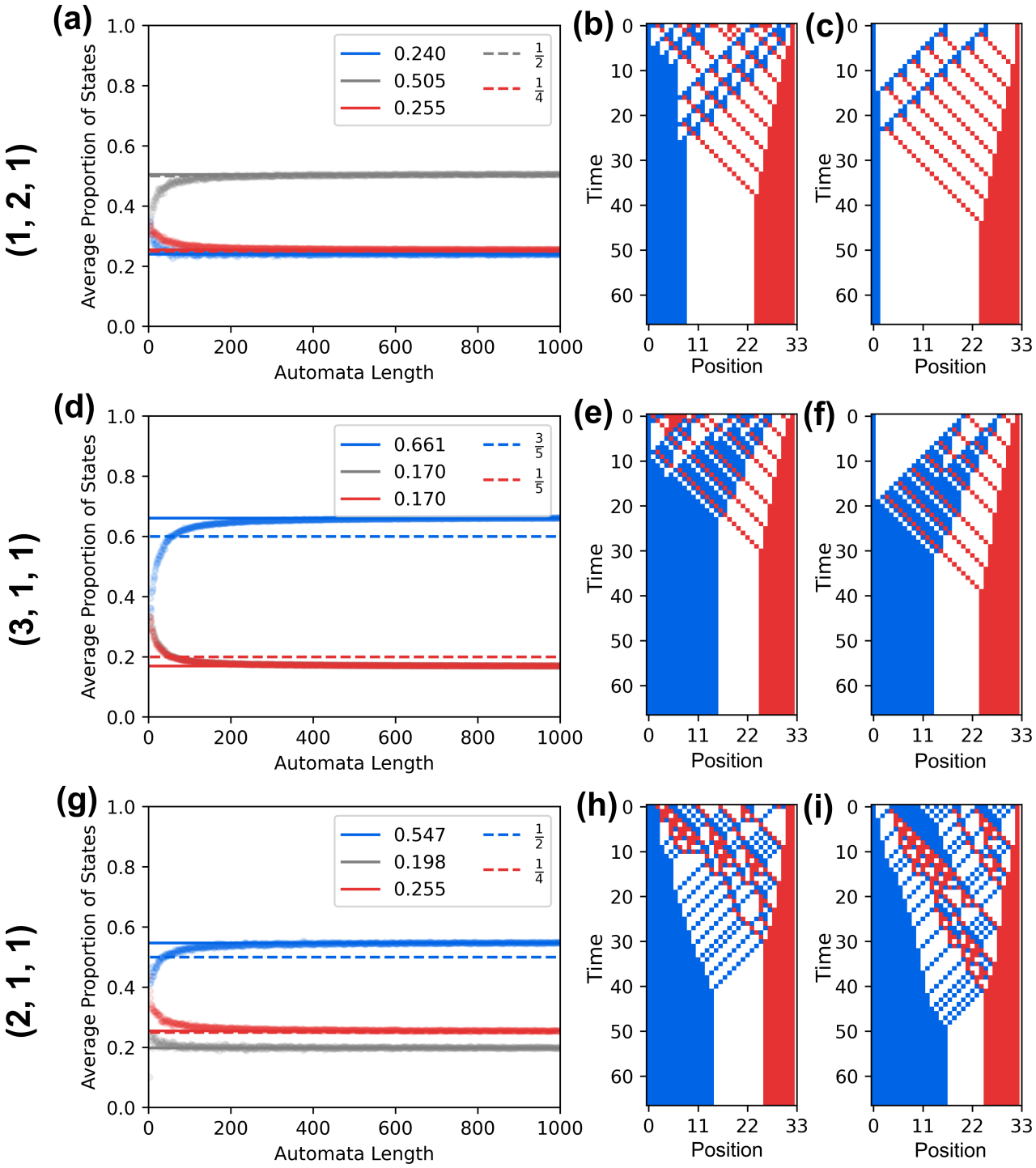


Figure 4.11 Solutions for Different Target Ratios Found by Evolutionary Algorithm Search. Average proportion of each state as a function of the automata length (a, d, g) as well as exemplary kymographs from random (b, e, h, j) as well as white with perturbation (c, f) initial conditions for the blue:white:red stripe width target ratios of 1:2:1 (a-c), 3:1:1 (d-f), and 2:1:1 (g-i).

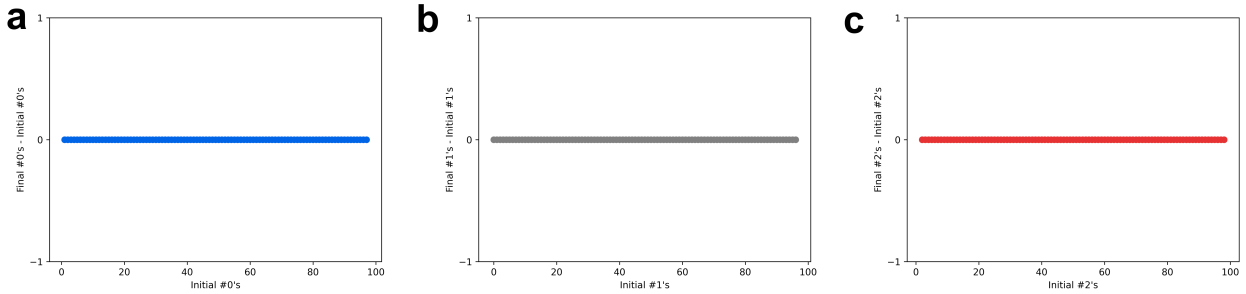


Figure 4.12 Conservation of States for the Bubble Rule. Difference between initial and final number of 0s (1s, 2s, respectively) as a function of the initial number of 0s (1s, 2s, respectively), demonstrating the conservation of the number of states through the Bubble rule dynamics.

the Bubble rule conserves their relative proportions (fig. 4.12) and merely sorts them. A simple solution to our problem would therefore just be a modification of the distribution of the initial states. Assuming this is not possible, it can additionally be used that the resolution module which resolves sorting conflicts only uses parts of all available rule entries with inputs containing black states, i.e., a lot of the inputs containing black states are just chosen to distribute any blacks in the initial condition evenly to the other colors for subsequent sorting. Changing the even balance of this distribution to lean more towards a desired state can thus be expected to change the color ratio in the final flag when provided with a evenly distributed initial condition. Testing this prediction slowly mutating the rule entries with black inputs which are not in the sorting module (fig. 4.13), it can be seen that the average proportion of states can indeed be tuned in a fine grained manner for any color of choice.

4.9.3 10 Shades of Blue: Combining Bulldozing and Sorting Modules to Generate Stripes with More Target Patterns

So far, the number of states in the target pattern was kept constant and their number simply motivated by the original French Flag problem. From a conceptual viewpoint it is interesting to reflect if the previously obtained rules can be modified to create more stripes. Mixed and Landscaping Rule are a good candidate for this since their operating principle is to erase the pattern using one of their states and then recreating it by seeding the other states in an alternating fashion with a period of 2 (Mixed) or 3 (Landscaping) states for the original French Flag and subsequently sorting them. Simply increasing the period to a larger number of states and generalizing the sorting modules to any combination of states proves successful in producing target pattern with more stripes with approximately equal ratio (fig. 4.14). In general, patterns with N stripes formed by rules following the principle of the Mixed (Landscaping) rule are expected to require the use of N ($N + 1$) states.

4.10 Changing the Initial Pattern

Whilst a random initial condition with a flat distribution over the states is probably the most unbiased initial condition, in biological systems cells often start in an undifferentiated state and change later to a more differentiated one. In the model here, this can be mimicked by interpreting the black state as undifferentiated, i.e., apart from the fixed boundaries, all other cells are initially black, while the goal of forming a French Flag as target pattern stays invariant. Applying the Landscaping rule to this initial condition solves the French Flag problem perfectly, apart from the usual fitness oscillations if the system size is not dividable by 3 and thus the target pattern not completely specified (fig. 4.15 (a)). Alternatively to applying the Landscaping rule, other solution strategies can be searched by evolutionary algorithm similar to above, with the modification of only allowing sampling rules with the correct steady state mappings. The rules all use the same strategy of having a uni-colored front moving with speed of on average $\frac{1}{2}$ cell per time step

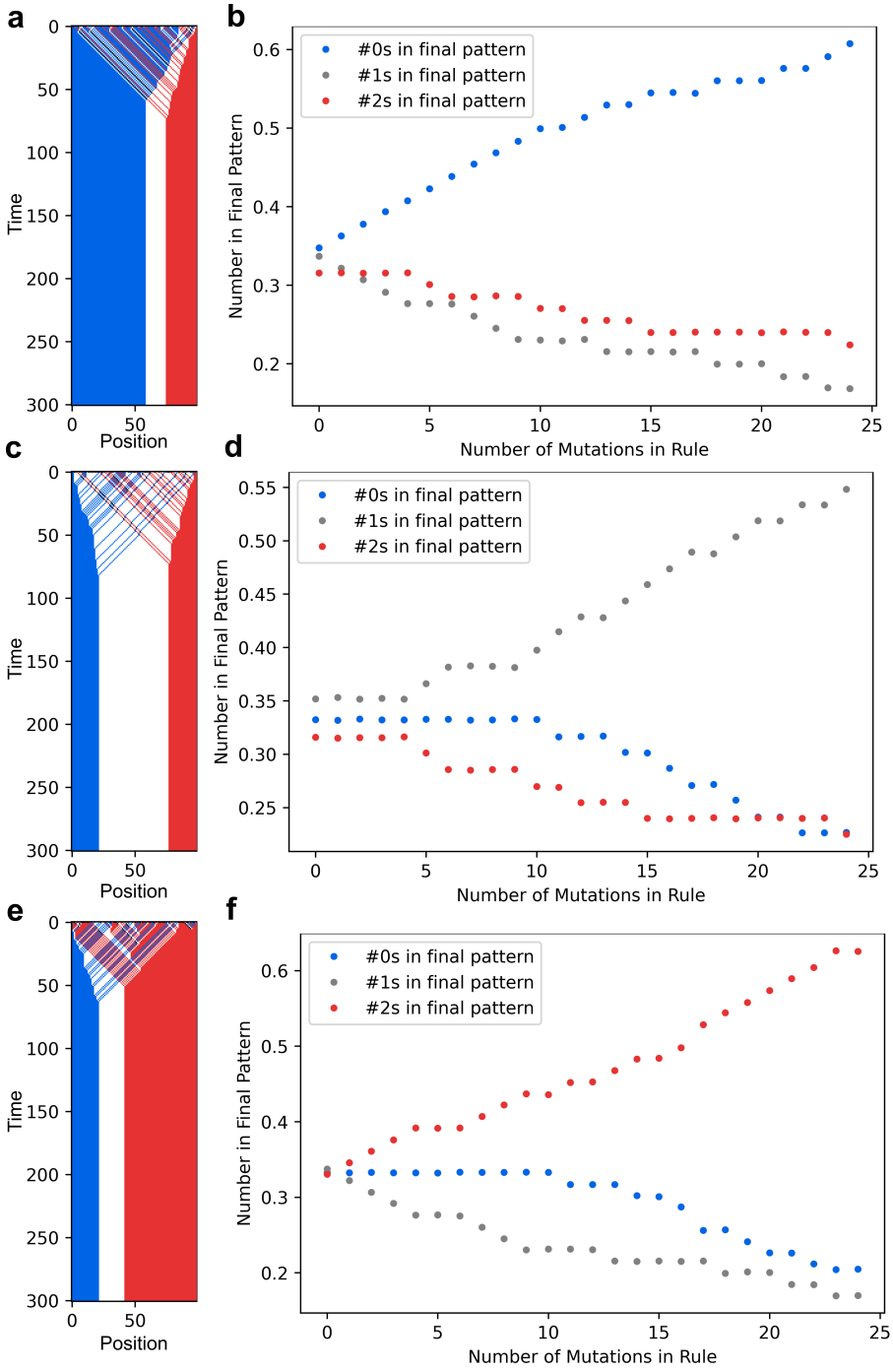


Figure 4.13 Modifications of the Bubble Rule. Exemplary kymographs (a, c, e) and number of 0s (1s, 2s) in the final pattern as a function of the number of mutations in the Bubble rule (b, d, f), demonstrating the possibility of gradually changing the outputs of inputs with black states that do not belong to the sorting resolution module lead to a nearly continuous change in the width of the stripes.

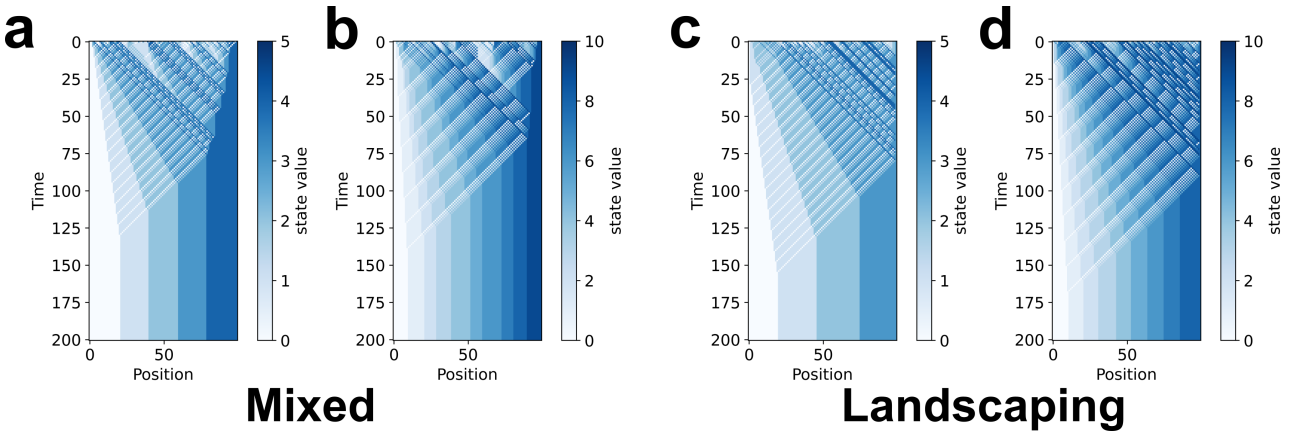


Figure 4.14 More States. Exemplary kymographs for generalizations of the Mixed (a, b) and Landscaping rule (c, d) to more than 4 states.

to the middle, meeting with a front producing white and the leftover color and moving at speed 1 (fig. 4.15 (b, c)). The states produced by the faster front get sorted accordingly.

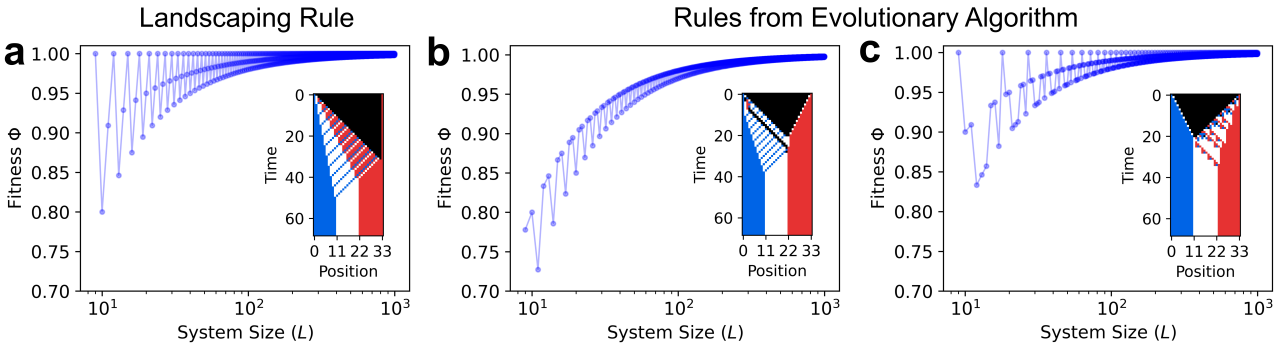


Figure 4.15 Homogeneous Initial Pattern Fitness as function of the system size and exemplary kymograph starting from a homogeneous initial condition of (a) Landscaping rule and (b, c) exemplary rules found by using evolutionary algorithms. The EA only used rules with the correct steady state mappings.

4.11 Discussion and Outlook

4.11.1 Summary

In order to solve the French Flag problem of axial pattern formation with only local interactions, a cellular automaton model was employed. With evolutionary algorithms, consensus procedures and engineering approaches, solutions in 3 and 4 state space based on two different patterning strategies were found. One patterning strategy is to sort a prexisting distributions of states in the correct order, the other is to erase the existing pattern and create states in the right proportion. Combining these two strategies allows for the construction of qualitatively different solutions. It was shown that the Bubble rule solution basing on sorting states is robust against noise, growth and asynchrony, as well as its stripe width tune-able. The Landscaping rule based on erasing the preexisting states and sorting the newly created ones is robust against changes in the initial condition and its fitness converges faster to 1 as a function of the system size.

4.11.2 Comparison with previous results

From a method point of view, the work of Rohlf and Bornholdt is related concerning several aspects [152, 160]. They used a 3 state CA and evolutionary algorithms to successfully find a solution to the problem of dividing an initially random 1D pattern into two compartments with sizes $0.3L$ and $0.7L$ where the left, small compartment comprising of only cells in state 2 and the right compartment comprising only of cell not in state 2. They showed scalability of the pattern with system size and robustness under noise and growth similar as demonstrated above. Further, they investigated growth at the boundaries, robustness to "cuts" and errors in directional signaling. Compared to the results presented in this thesis, their problem was less strictly defined (only a condition on one of the the compartments) with just one special solution that was discussed, which did not allow the investigation and characterization of more general patterning principles for axial pattern formation.

From a results point of view, Ancona et al. [161] demonstrated in a general message passing model different solution algorithm for solving the French Flag pattern. One of their solutions uses a variant of counting the passed on messages to determine the (relative) position in the pattern and assign a color based on that relative position. Another solution utilizes a sequential version of the Landscaping pattern formation idea, where first the correct proportions of the colors are generated across the system, followed by a Bubble sort to bring the colors in the correct order. All proposed solutions have a slower kinetics, and similar or larger memory requirements then solutions found here.

4.11.3 Increased Interaction Range

The model used in this chapter included only interaction between directly neighboring cells. Yet, interactions between non neighboring cells are possible, for example trough filopodia extending from one cell to a more distant cell (see [4, 56] and fig. 2.1 (d)). To include those interactions, in an exploratory project conducted by María Molina Antón supervised by the author of this thesis (personal communication, [147]), the interaction range of the CA was increased. Naturally, this vastly increases the rule table and complicates the analysis. Using a similar methodology as described in this section, rules that could solve the French Flag problem were found. The kymographs for different carefully chosen exemplary initial conditions from one of the found rules are shown in fig. 4.16. A number of hypothesis can be made from observing the kymographs: The blue stripe is generated by red states hitting it with white to the right (a-d). While red moves with a speed of one cell per time step (a, d, e, f), white can move with larger speed to the left (b, c) and right (f). The red stripe is generated similarly by blue states hitting the red stripe, while the blue seems to move right slower than 1 cell per time step on average (d). The findings hint at more elaborate mechanisms that can work for larger interaction ranges, partially using the possible larger pattern formation speed. A slightly different approach was chosen by T. Ramalho by investigating 1D system in discrete time and space, but continuous state [162]. The update function was of the form of a recurrent neural network, and solving for a 2 state partition of the cells showed traveling wave solutions, yet was unable to find a steady state [162].

4.11.4 2D Topology

A simple yet severe restriction of the above model was the one dimensional topology, which may be model certain aspects of axial pattern formation, but aspects of increased fitness and error correction in axial patterns via the second dimension or patterns different from axial patterns cannot be investigated. An ongoing project with Lukas Zett [148] uses a split up of the 2D rule in a horizontal rule and a vertical rule both subsequently acting on a cell and its 2 immediate neighbors in horizontal and vertical direction, respectively (first the vertical then the horizontal rule). When using the Consensus rule in horizontal direction, an evolutionary algorithm can be used to optimize the fitness with respect to the rule in vertical direction. The result are rules outperforming a vertically uncoupled version or a version coupled by the majority rule which always takes the local majority of states as an output for the vertical direction [148]. The coupling of a Consensus rule in horizontal and a majority rule in vertical direction significantly improved

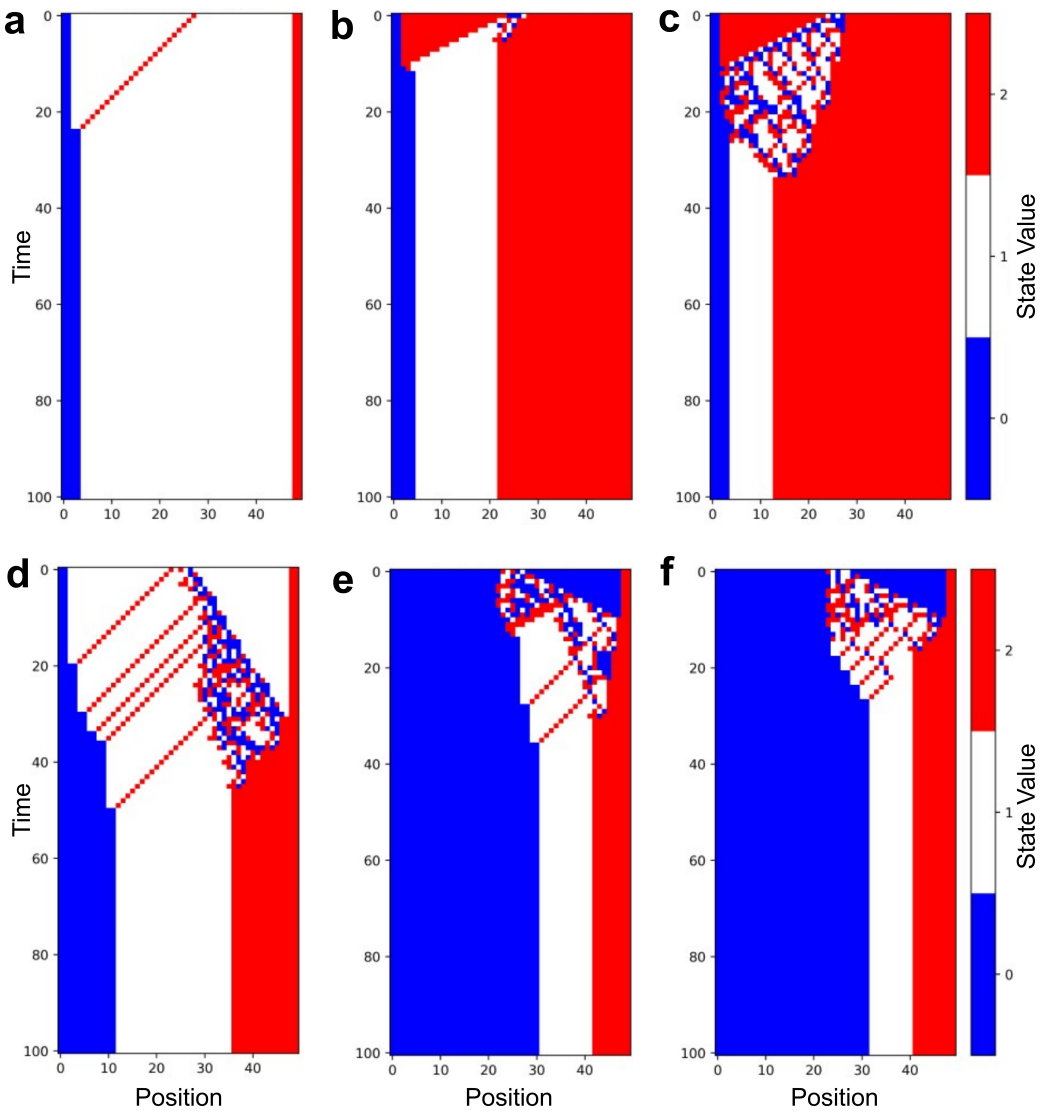


Figure 4.16 Increased Interaction Range Kymographs for an exemplary rule with interaction range $r = 2$ for exemplary initial condition chosen in order to reveal the basic mechanisms of the rule. Figure adapted from Maria Molina Anton with permission [147].

the fitness once noise was present, but was strongly susceptible to errors due to asynchrony, with a reversed behavior of a horizontal Consensus rule coupled to a with regards to fitness optimized version of the vertical rule [148]. Another possible solution found was to use a rule in vertical direction that creates a nearly homogeneous pattern, which is then patterned by a horizontal rule specialized in forming this intermediate pattern to a French Flag [148]. Additionally, solutions for triangular, hexagonal and irregular topologies were found. The former two generally outperform uncoupled horizontal 1D Consensus rules with regards to fitness, while for irregular grids the fitness is lower for large systems [148].

4.11.5 Increasing the Search Quality

Most extensions previously discussed in this chapter require the search of a rule-space which is often many magnitudes larger, e.g., for 3 states 1D $r = 1$ CA can have $3^{(3^3)} \approx 7.62 \cdot 10^{12}$ different rules, $r = 2$ CA $3^{(3^5)} \approx 8.72 \cdot 10^{115}$ different rules, and 2D CA with a Moore neighborhood $3^{(3^9)} \approx 1.51 \cdot 10^{9391}$ different rules. It is unclear if evolutionary algorithms are able to find well working rules in such vast spaces, thus other approaches might be needed. If there are other features which need to be illuminated like the time to steady state, mapping elites has shown to be a promising algorithm outperforming evolutionary algorithms [163]. Its basic principle is instead of an unstructured list of rules it uses solutions over a low dimensional feature space to construct new rules. If such a rule is a better solution than the already existing solution in the corresponding sector in feature space it replaces the old rule [163]. Other algorithms exist that do not rely on a low dimensional feature space, but instead work in the high dimensional rule space and create a structured list of well performing and diverse rules by adding new rules to the list if they are either more performant than rules in their close vicinity in rule space which are already in the list or "novel" compared to the rules already in the list [164, 165]. Combinations of both conditions are also possible [165]. New rules ("offsprings") are created by mutation or crossover from parent rules chosen for the rules in the structured list and the recently created rules [165]. If the offsprings are added to the structured list, their parents get awarded with a higher "curiosity" which influences the likelihood of becoming a "parent" again [165].

4.11.6 Evolutionary Perspective: Fitness Landscapes

Another interesting avenue of research was proposed by Gabriel Vercelli, who opted for seeing the model system in 4 state space as example for a possible space of all solutions a evolutionary process (in the sense of biological evolution, not as evolutionary algorithm as mere search tool in this rule space) could explore. This opens up questions like if there exists a path of continuous mutations between prototypical rules which contains only high fitness rules.

5 How to Steer Global Pattern Formation With Local Interactions? - Programmability

This chapter consists of work started by Hao Wu [166] and Tiago Ramalho [162], extend in [167] and in this thesis by the author, and cumulatively published in [168]. Here, a larger emphasis is given to the points developed during the time of this thesis[1] while other parts necessary for a coherent story are presented in a more concise manner partially omitting proves or calculations already presented in [166], [162] or [167]. All were supervised by Ulrich Gerland. The author thanks Kilian Vogele and Friedrich Simmel for fruitful discussions.

5.1 A Model for Programmability

In the previous chapter, finding a rule for the formation of a specific pattern from an unknown arbitrary initial conditions without any additional inputs using local, nearest neighbor signaling was discussed. Here, the focus is shifted to programmable pattern formation in the sense that the target is to find interaction logics that are able to form any desired pattern from any given initial condition given that 1 or 2 cells at the boundary or in the middle of the pattern supplement the appropriate inputs by changing their state, again only relying on local, nearest neighbor signaling.

This target is motivated by organizer cells in developmental biology which instruct their neighboring cells via cell-cell signaling while they themselves do not react to any cues recieved from the outside. The most famous example of an organizer region is the Spemann-Mangold organizer inducing the main body axis in early *Drosophila* development [2, 83]. A single "anchor cell" also organizes *C. elegans* vulva development with a following signal relay by local interactions [28, 84]. Further, it has been demonstrated that the evolution of formed patterns can occur through minor changes in cis-regulatory sequences which control gene transcription [2, 169]. From a synthetic perspective, optogenetic manipulation of opto-Delta, an optogenetic allel of Delta, opens a perspective of manipulating the expression programs of individual cells effectively creating analogues to organizer cells [68].

As a modeling tool to investigate this research question, CA are chosen. In this CA, three types of cells are used (fig. 5.1 (a)): bulk cells, which all follow the same CA rules with synchronous updates; fixed boundary cells, which have a constant state; and organizer cells, which follow a protocol. All cells can have one of two possible states, 0 or 1 (fig. 5.1 (a)). They are arranged linearly in a line with either two organizer cells at both ends or one organizer cell at one and a fixed boundary cell at the other end. Alternatively, they are arranged in a ring with an organizer cell embedded (fig. 5.1 (b)). Starting from an arbitrary initial condition, the bulk cells are updated according to the CA rule, if necessary taking into account the states of organizer or fixed boundary cell (fig. 5.1 (c)). The goal is now to find CA rules which allow for any initial pattern and any target pattern a choice of organizer sequence that produces the target pattern.[2]

5.2 Limited Set of Rules Enables Programmability

As first attempt to solve this question exhaustive sampling of small systems is used. For a given system topology and system size, a network/graph is generated whose nodes are all possible patterns of the bulk

[1] in particular: the parts about the degree statistics and standard classification quantities in section 5.2; section 5.3 except for the part about microcanonical entropy; the comparison of different pattern complexities in section 5.6

[2] "Pattern" in this sentence can be any combination of 0s and 1s, not necessarily an ordered one.

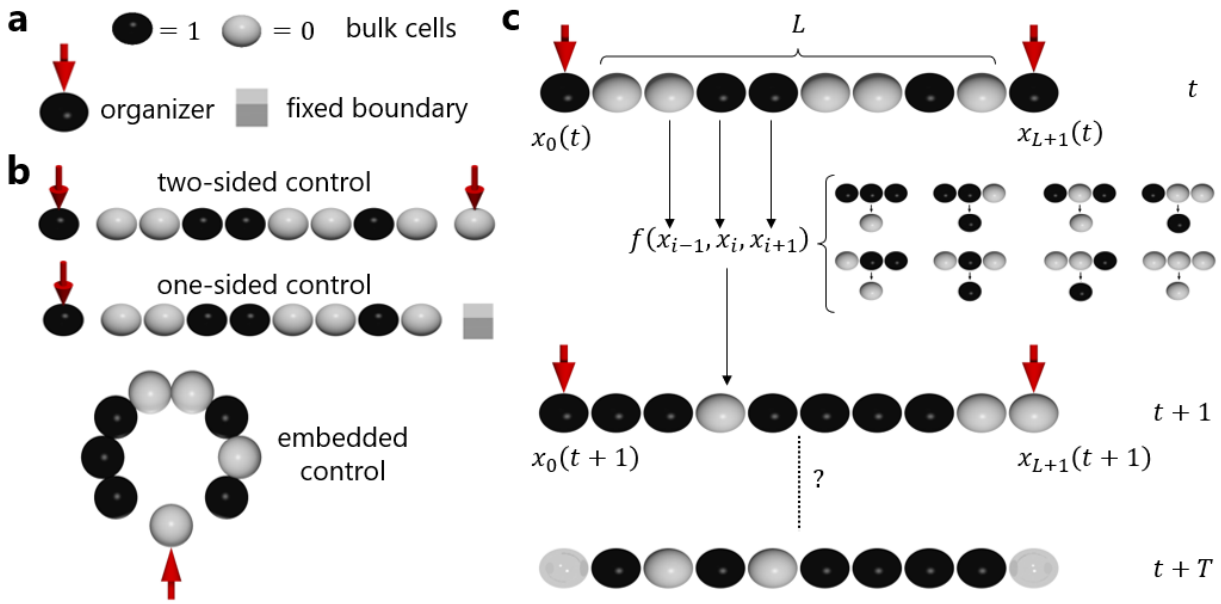


Figure 5.1 CA Model for Programmable Pattern Formation. (a) The model features 3 types of cells: bulk cells, following a CA update rule, organizer cells (marked by a red arrow), which follow a protocol, and fixed boundary cells (square), which have a constant value. All cells can have either state 0 or 1, marked by color. (b) Three topological setups are considered: A linear, 1D array of bulk cells with organizer cells on both sides (two-sided control), or with an organizer cell on one side and a fixed boundary cell on the other side (one-sided control), as well as a circular, 1D array of bulk cells with a single embedded organizer cell (embedded control). (c) The bulk cells follow a CA dynamics, while the organizer cells control the patterning process by following a protocol of state changes. A given CA rule is programmable if it allows the organizer cells to steer the pattern formation to any target pattern starting from any initial pattern using an appropriate sequence of organizer signals. *Reproduced from [168] under the Creative Commons Attribution 4.0 International License [170].*

(fig. 5.2 (a)-(c)). For each such pattern and each possible organizer input(s), the CA is executed once synchronously and then an directed edge is draw between the initial pattern and the pattern after a single step (fig. 5.2 (b)-(c)). For a given system size and system topology, this network now contains all possible transitions between patterns and will be called "patterning graph". Thus, checking whether it is possible to transition from any initial to any target pattern in the patterning graph is equivalent to checking if it is possible to transition from any node of the patterning graph to any other node along the directed edges, i.e., checking if the patterning graph is strongly connected [171]. Using standard algorithms implemented in Igraph [172] or NetworkX [173], respectively, it is possible to check which rules generate strongly connected patterning graphs for a given system size and topology and are thus “programmable”.

The number of representatives from the CA equivalence classes (section 3.3.1) which are programmable decreases monotonically as a function of the system size and seems to reach a stationary value for system sizes larger than $L = 8$ (fig. 5.2 (d); for the “one-sided control” scenario a organizer cell on the left was chosen). The representatives of the equivalence classes are tabulated in fig. 5.2 (e).

In order to explain how the programmable rules work, as a first measure the degree statistics of the corresponding patterning graphs is investigated. For a given number of possible combinations of the states of the organizer cells, the out-degree of each node, which is the number of edge-tails which begin at that node, corresponds to that number - i.e., for a single organizer the out degree is 2, for two organizers and thus 4 possible combinations of organizer states the out degree is 4. The in-degree distribution, which is the distribution of edge heads ending at a node, can in principle be any distribution conserving the cumulative degree and depends on the rule, but notably for programmable rules it is a distribution with a single nonzero value of the corresponding out degree in contrast to nearly all other rules (fig. 5.2 (f) and fig. A.8). The only other two rules with a similar in-degree distribution are the *identity* and *complement* rule which are both trivially not programmable. The form of the in-degree distribution is somewhat surprising since a lot of different degree distributions would allow for a strongly connected network, e.g., a single path of edges connecting all nodes in a circular structure with all other edges pointing any desired nodes. It also insinuates that programmable rules may be distinguished from non-programmable ones by sampling the local structural statistics of their patterning graph which is also numerically feasible for larger patterning graphs.

With the same goal in mind, also the values of standard classification quantities (chapter 3) was calculated for all 256 elementary CA rules, highlighting the programmable rules amongst them (fig. 5.3). Starting from the top, Langton’s λ is both for the 0 or 1 as quiescence state for all programmable rules at $\frac{1}{2}$ - which is not surprising since in the table in fig. 5.2 (e) the given algebraic expressions for the rules indicate that all of them are linear in at least one of their arguments, making a change in that argument change the output of the rule and thus leading to an equal 0 - 1 output distribution of the rule. The μ-sensitivity does not show a particular structure which can again be explained from the algebraic expressions which show that while the programmable rules do depend at least in some inputs linearly, most either do not depend on all inputs or do depend on them in a nonlinear fashion. The estimate for the Kolmogorov complexity of the rule table does also not reveal any particular structure, yet both spatial and temporal Kolmogorov complexity estimates show comparably large values for the programmable rules, indicating that without a specific external protocol the space of patterns does not get compressed by the programmable rules, consistent with the in-degree distributions discussed above and the values for spatial and temporal single point entropies. The information integration or Synergy on the other hand does not show a particular structure with the values being distributed over both large and low values, indicating that some rules introduce dependencies between the inputs, while others do not which again becomes apparent by looking at the algebraic expressions on the rule table. Summed up, the analysis above indicates that computing estimates for the spatial and temporal Kolmogorov complexity and single point entropy might, together with a sampling of the in-degree distribution, hint which rules to check for programmability in larger rule spaces.

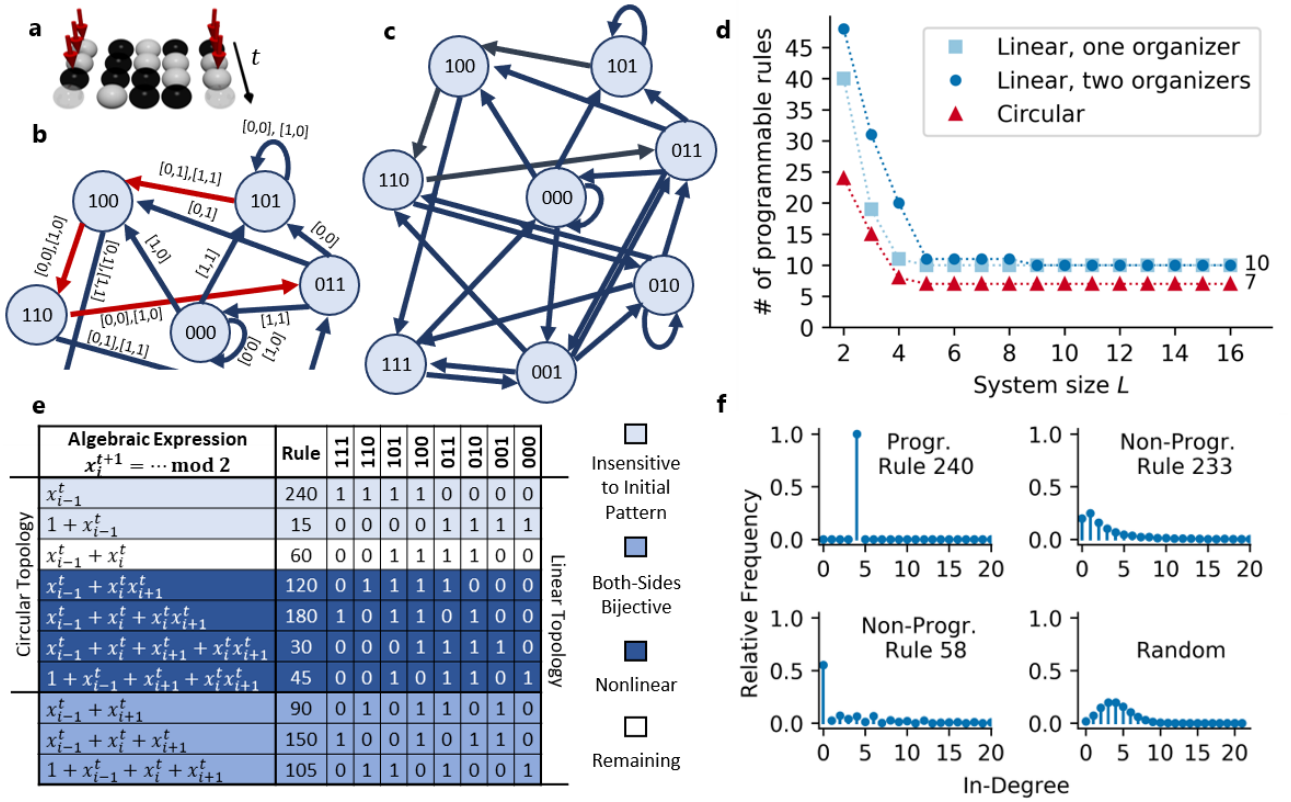


Figure 5.2 Patterning Graphs. (a) Exemplary system with $L = 3$ bulk cells following rule 86 and two organizer cells leading the pattern from 101 to 011. (b) Excerpt of the patterning graph of the system depicted in (a) ($L = 3$, rule 86), defined nodes being patterns with $L = 3$ and directed edges being possible transitions for rule 86 and the possible organizer states, with the red path corresponding to the patterning dynamics in (a). Organizer states that corresponds to a specific transition are noted next to the directed edges. (c) Whole patterning graph for $L = 3$ and rule 86 and two-sided control. A rule is called programmable if transition form every bulk pattern to any other bulk pattern is possible, i.e., if the graph is strongly connected. (d) Number of programmable rules as a function of the system size for different system topologies, calculated by exhaustive generation of the patterning graphs and checking for strong connectivity for one member of each equivalence class (section 3.3.1). The exact number of a linear system with one organizer varies depending on the choice of the representative rule (not shown). For system sizes $L > 8$ the number reaches a plateau value. (e) Programmable rules which comprise the plateau in (d) - algebraic expression, rule number and input-output mappings, as well as indications about the sensitivity on the initial pattern, the bijectivity of the rule (see below) and non-linearities. (f) Examples of in-degree distributions for different rules and a random assignments of directed edges to a graph with the same number of nodes and edges as the rules. Programmable rules have in general a histogram with only a single nonzero value. *Reproduced from [168] under the Creative Commons Attribution 4.0 International License [170].*

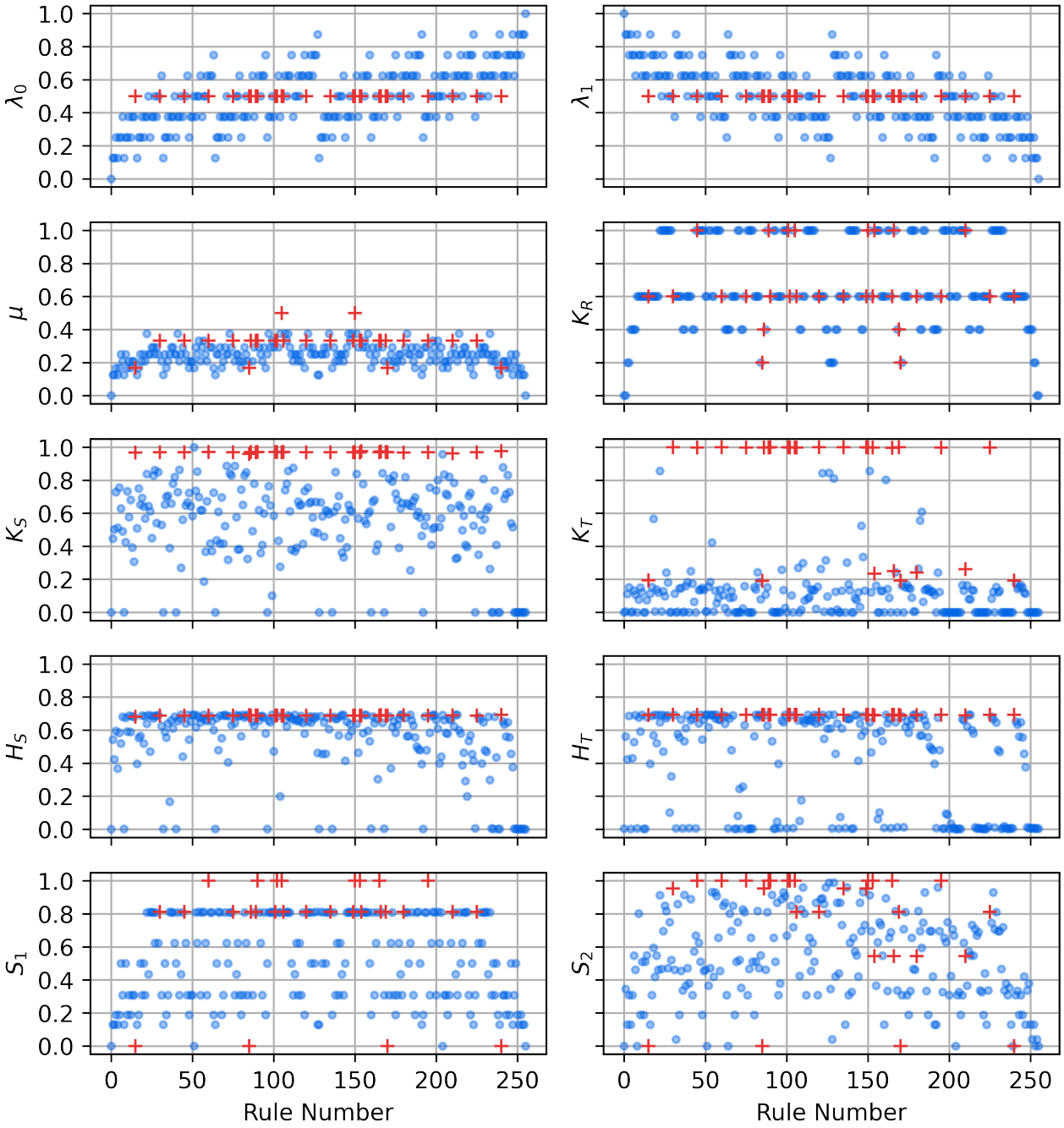


Figure 5.3 Classification of Programmable Rules by General Schemes. Langton's λ with 0 (λ_0) and 1 (λ_1), respectively, as "quiescent" state; μ sensitivity; Kolmogorov complexity of the rule table K_R, of the pattern along the spatial axis K_S (length $L = 100$), averaged over 1100 time steps and 10 samples with random initial conditions, and of the pattern along the temporal axis K_T averaged over the spatial axis with similar parameters as above; Single site entropy H calculated along the spatial H_S and temporal axis H_T, respectively, and averaged over the other, with same parameters as above; Numerical calculation of the the synergy for 1 or 2 time steps S_1 and S_2, values from M. Bojer [116]. Programmable rules are marked with a red $+$.

5.3 Rules Enabling Programmability Form Patterns in Different Ways: Patterning Strategies and Patterning Kinetics

After having determined the programmable rules and contextualized their values for some classification quantities, in this section a closer look on which mechanisms the individual rules use to form the desired target patterns is taken. In fig. 5.4 (a)-(h), it is exemplified how different programmable rules reach desired target patterns (most of the time 10011100), starting from an all 0 initial condition except for (g). For (a)-(c), examples are given with one organizer cell on the left and a fixed boundary cell on the right. The probably simplest example is rule 240 ($x_i^{t+1} = x_{i-1}^t$) (fig. 5.4 (a)), which just moves the whole pattern one cell to the right and the target pattern is thus plainly written into the left organizer cell (except for the 0 part of the target pattern to the right). This is reminiscent of the clock and wavefront mechanism of vertebrate somitogenesis [13] in the sense that a temporal signal gets written into a spatial pattern. The temporal signal may also be processed, e.g., 15 (fig. 5.4 (b)), shifting the pattern to the right and simultaneously inverting the color at each step, requiring a different organizer protocol. The processing might be more complex, exemplified by rule 105 (fig. 5.4 (c)), generating also parts of a complex pattern without the external input and, depending on the initial and final pattern, shortening the time it takes to form the target pattern. This can also be achieved by using organizer cells on both sides, which allows some rules to systematically shorten the time to reach the target pattern by processing information from both sides symmetrically (rule 90, fig. 5.4 (d)). Some rules may also process this information asymmetrically, only partially using one of the inputs from one of the organizer cells (rule 30, fig. 5.4 (e)). At last, in fig. 5.4 (f)-(h) the organizer cell is embedded in bulk cells in a linear arrangement. The pattern information might again be only transmitted to one side (rule 240, fig. 5.4 (f)). It may also in some cases be computed directly from an initial pattern (rule 30, fig. 5.4 (f)) or the information might be pushed in both directions, creating an interference like phenomenon which eventually produces the target pattern.

To substantiate the loose claims made above about the patterning kinetics, it is useful to measure how much the pattern space gets "compressed" by the action of the organizer cells. In order to do so, the following thought experiment is undertaken: After selecting a target pattern, a little walker is placed at every node of the patterning graph. All walkers move towards the target pattern along the shortest possible path on the patterning graph. The amount of node still occupied by walkers, in the sense of statistical physics the occupied "state space", is recorded as a function of time. The logarithm of this quantity corresponds to a microcanonical entropy-like quantity $S(t)$. For a fixed system size and a single target pattern S is plotted as a function of time for a circular topology with an embedded organizer (fig. 5.4 (i)) and a linear topology with 2 organizers (fig. 5.4 (j)) for the corresponding programmable rules. For the circular topology (fig. 5.4 (i)), rules that only transfer information into one direction (rules 15, 60, 240, cp. fig. 5.4 (f)) follow a linear law ($S(t) = L - t$), while rules that transmit information in one direction linearly and in the other non-linearly (30, 45, 120, 180, cp. fig. 5.2 (e) and fig. 5.4 (h)), have a non-linear dependency on time. For two organizers, rules that process information linearly from both sides (rules 90, 105, 150, cp. fig. 5.2 (e)) form patterns faster ($S(t) = L - 2t$) than the other programmable rule. These fall into two sub-classes. Rules that depend only linearly on the left entry and not the right entry show a perfectly linear behavior with slope -1 ($S(t) = L - t$). Rules that depend on the left entry linearly and non-linearly on the right entry show only a roughly linear behavior with slope -1 ($S(t) = L - t$)).

In conclusion, rules which process information linearly in both outer arguments can utilize two organizer cells to form patterns twice as fast compared to using just one organizer cell or having two organizer cells but rules which do not take inputs from one of their outer arguments. Rule with non-linear processing of one of the outer arguments and a linear processing of the other can utilize two organizer inputs only slightly better compared rule which only process one outer argument and ignore the other, but take longer on average to form the pattern when placed in a circular topology because the same signal gets processed twice in a different fashion.

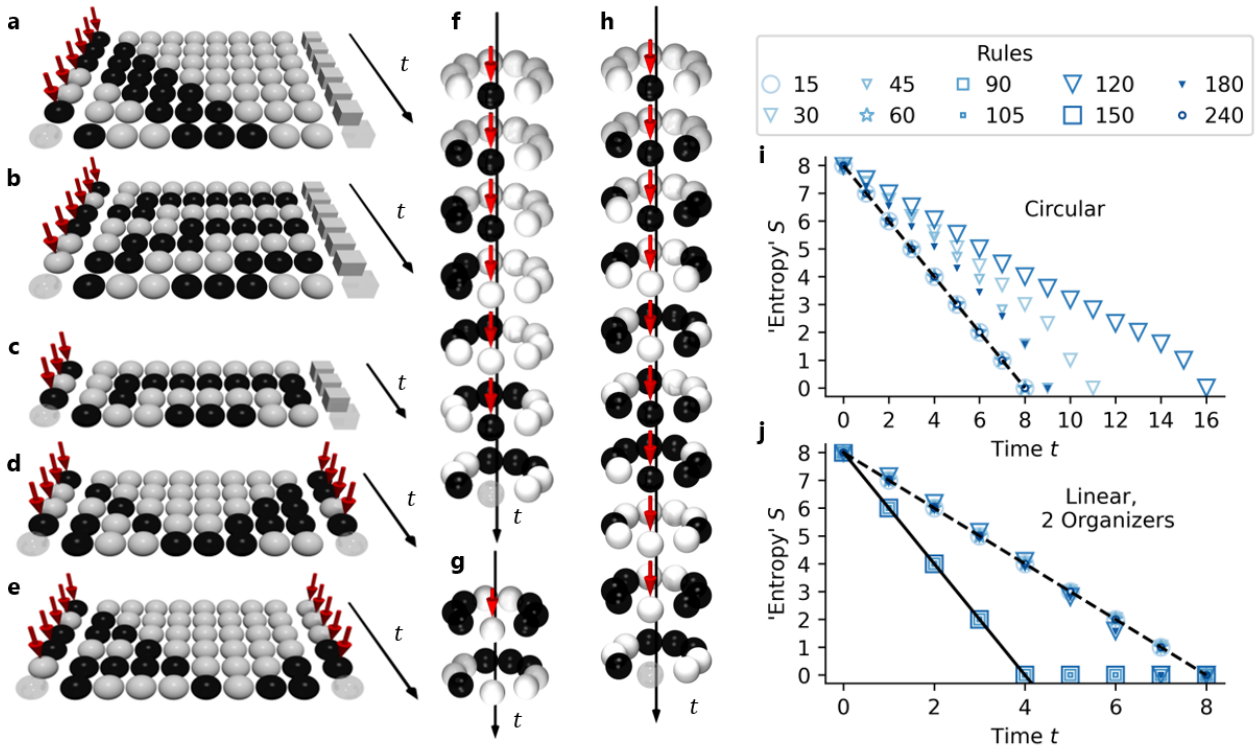


Figure 5.4 Patterning Strategies and Kinetics. (a)-(g) Exemplary kymographs for different rules and topologies. **Linear topology, one organizer cell:** (a) Rule 240 copies the organizer sequence 1:1 into the pattern. (b) Rule 15 copies and additionally inverts the color at each step (c) Rule 105 shows a more complex information processing, leading to a much quicker formation of the desired pattern from the same initial condition. **Linear topology, two organizer cells:** (d) Rule 90 uses inputs from both sides equally, while rule 30 (e) fully uses inputs from one side, but only partially inputs from the other side. **Circular topology:** (f) Rule 240 functions similar in circular topology (g) Rule 30 can use information from both sides, which due to the interference of the signals from the inputs may lead to a long pattern formation time (h). (i, j) Entropy S (occupied rule space, defined as the logarithm with base 2 of the number of occupied patterns when starting from every initial pattern and moving towards a defined final pattern) as a function of time for the programmable rules. Dashed lines correspond to $S = -t + L$ and $S = -2t + L$, indicating a linear decreases for all rules (linear, 2 organizers) or only linear rules (circular), with rules with linear inputs on both sides being the fastest, while non-linear rules being slower in circular topology. *Reproduced from [168] under the Creative Commons Attribution 4.0 International License [170].*

5.4 Bijectivity Is a Sufficient Criterion for Programmability

The previous section already hinted that programmability has a connection to the linearity of the algebraic expression. In this section, this notion is formalized with the concept of bijectivity [174].

Definition: *A rule is called left bijective, if the mapping* $x \rightarrow y$ *with* $y = f\left(\cdot, x_i^t, x_{i+1}^t\right)$ *is bijective for each combination of* x_i^t, x_{i+1}^t *values (fig. 5.5 (a)-(b)). Right side bijectivity can be defined analogously.*

This means for a left bijective rule, the output x_i^{t+1} can be set to any desired value by chosing x_{i-1}^t appropriately. Translating this argument back in time and space until an organizer cell is reached allows control of x_i^{t+1} (fig. 5.5 (c)). The argument made in fig. 5.5 (c) allows for the following conjectures:

- If a rule is right side bijective, the same argument can be made with an organizer cell on the right.
- The argument is constructive in the sense that it produces a recipe for calculating the organizer sequence $\mathcal{O}^t$ to reach the target pattern given the initial pattern and the rule.
- It explains the patterning kinetics discussed in the previous section: rules that can take inputs from only one side need at most L steps to form the target pattern, rules that can take inputs from both sides at most $\frac{L}{2}$ (or $\frac{L+1}{2}$ if $\frac{L}{2}$ is not of integer value).
- It also explains part of the discussed pattern formation mechanisms: For simple rules like 240 and 15, the organizer sequence can in principle be chosen independent of the initial state, although taking the initial state into account might increase the speed of pattern formation. For more complex rules, the initial state will influence the pattern formation and has to be taken into account.
- The bijectivity argument does neither use a specific number of states nor a specific system size.
- This also sets the lower boundary on number of programmable rules to $k!^{k^2}$.

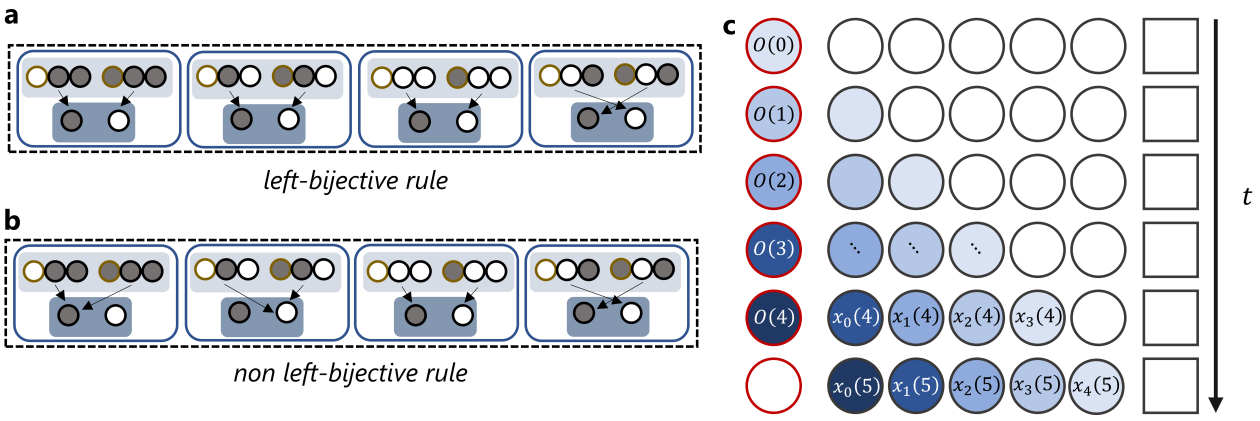


Figure 5.5 Organizer Sequence Construction for Bijective Rules. (a) Example of a left-bijective rule which maps, for same middle and right input, to different outputs for different left inputs. (b) Example of non left-bijective rule. (c) Illustration of the organizer sequence construction for a left-bijective rule. The values of all white cells are not influenced by the organizer sequence. To set the state of $x_4(5)$, the state of $x_3(4)$ must be at the correct value, while left-bijectivity of the rule guarantees that such a state exists. To set $x_3(4)$, $x_2(3)$ must be at the correct value. This argument can be carried on up to the organizer $O(0)$, which can per definition take any desired value and thus determine $x_4(5)$. The same argument can now be made for $x_3(5)$ and $O(1)$ and so forth. Thus, bijectivity of the update rule is a sufficient condition for programmability. *Reproduced from [168] under the Creative Commons Attribution 4.0 International License [170].*

5.5 Expanding to More Dimension Allows Buffering of Errors

In realistic settings, the communication between cells, storage of the cell state, and processing of the information are all erroneous. To model this, a probability p is introduced with which at each step a cell changes its value to the opposite value similar to section 4.8.1. As observable, the probability of not having an error in the final pattern $Prob\,(noerror)$ is chosen. For rules that are only bijective in one of their outer arguments it takes at most L steps to reach the target pattern and thus, by the construction in fig. 5.5 (c), at most errors in $\frac{L(L-1)}{2}$ updates can influence the final pattern, leading to

$$Prob\,(\text{no error}) \sim (1-p)^{\frac{L(L-1)}{2}} \approx 1 - \frac{L\,(L-1)}{2}p + \mathcal{O}\left(p^2\right). \tag{5.1}$$

This estimate together with its Taylor expansion to the first order is shown in comparison to the numerically sampled values for the programmable rules in fig. 5.6 c, confirming that the estimate indeed reproduces the general behaviour of the model and showing the great sensibility towards errors.

In a one dimensional system with limited range, a correction of the errors is not directly possible since there is not information redundancy. To introduce the possibility of such a redundancy, the system is extend to two dimensions. The goal is to have a striped system in the sense that all cells in the y direction are supposed to having the same state (fig. 5.6 (a)). The cells are arranged in a tube topology with periodic boundary conditions along the y axis, with a total system size of $L \cdot K$ cells. The redundancy of the y direction offers a possibility of increasing the reliability of pattern formation by locally averaging over this direction, effectively equivalent of applying a majority voting rule similar to section 4.11.4 and [175], before applying the CA rule (fig. 5.6 (b)). Using a combinatorical argument for the possible arrangements of errors and their influences it is possible to show that

$$Prob\,(\text{no error}) \approx 1 - 2L^2Kp^2 + \mathcal{O}\left(p^3\right), \tag{5.2}$$

confirmed again by numerical sampling [167, 168]. Thus, the redundancy is able to reduce the sensitivity of the programmable rules to errors drastically. There is some variability among the programmable rules, which can be explained by the fact that there are rules which only shift the information (240, 15) and thus not spread any introduced errors, whereas rules that also process the information in a complex way also have the tendency to spread the errors and thus perform worst (rule 90, 105, 150). When changing the boundary conditions on the top and bottom to be fixed, the error correction process is jeopardized noticeably (fig. 5.7), with the exception of rule 15. As explanation it can be hypothesised that at the fixed boundary the error correction breaks down since only two cells in the majority voting are updated by a dynamical rule and the third is fixed - if those two cells have a different value correction is not possible. The exception is for rules that invert the patterns at each step as for rule 15, allowing the averaging majority rule also to have some effect on errors at the boundary cells.

5.6 Robustness Against Organizer Signal Timing

Up to now, the timing of the organizer signal had to be in precise lockstep with the updates of the bulk and the target pattern did not have to be a stationary pattern of the update rule. The former can be achieved by subjecting the organizer cells to the same synchronization mechanism as the bulk cells. The latter has possibly no practical influence since the time an update takes is never specified in the model and does not have to be the same for each step of the dynamics, and additionally transient patterns are a common feature of developmental systems [176]. However, it may still be interesting to ask what kind of patterns can be formed if these assumptions are relaxed. To this end, the model is changed insofar now it is always waited until the dynamics reaches a stationary pattern before the organizer inputs change, but at the same time also the rule is changed (fig. 5.8 (c)). The rule change would, in a developmental system, be equivalent to changing the information processing of the same signals a cell receives, as, e.g., exemplified by the Toll signaling pathway in *Drosophila* [169]. A combination of organizer inputs and rule is now referred

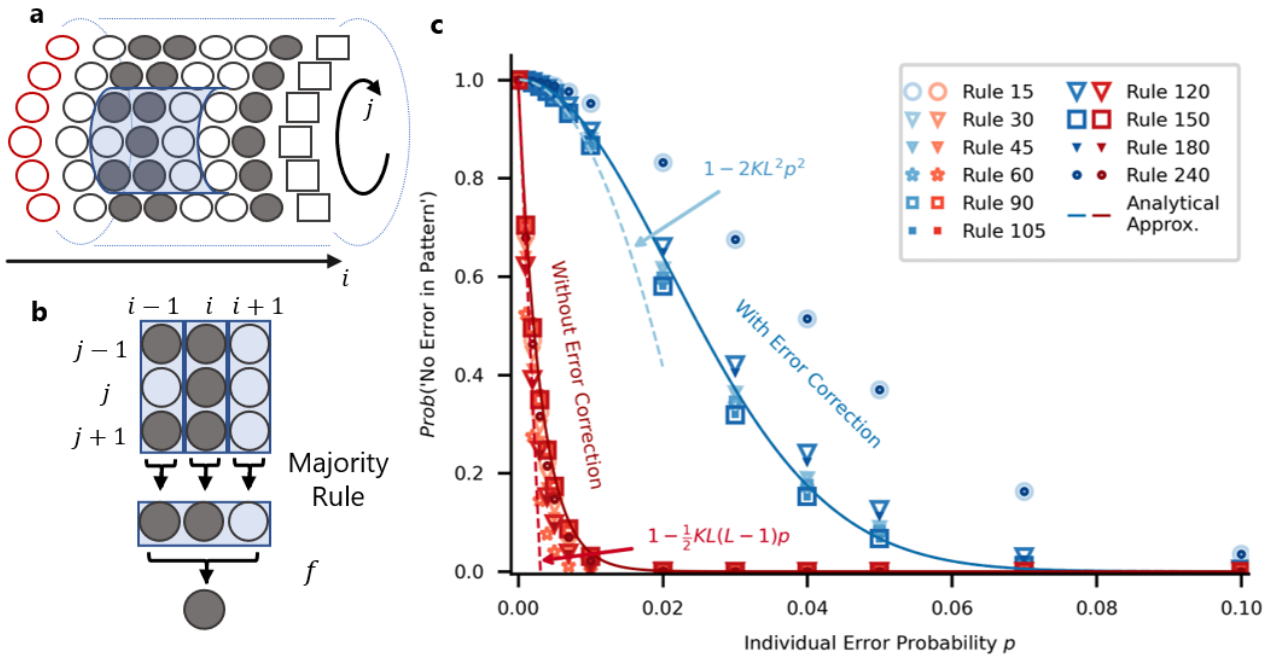


Figure 5.6 Robustness Against Errors and Error Correction (a) To investigate robustness against errors cylindrical grid is used with organizer cells on the left and fixed boundary cells on the right. (b) An update step of the Moore neighborhood comprises of an application of the majority rule in vertical direction, followed by a CA update in horizontal direction. (c) To quantify the susceptibility to errors, the probability of not having an error in the final pattern as a function of the probability of each cell mapping to the wrong state per time step p is calculated from simulations for a $9x9$ grid for the programmable rules (symbols). Solid lines give the values of an analytic theory with dashed lines representing the corresponding Taylor expansions up to the first non constant order. Red for systems without error correction, blue including error correction. Error-corrected systems are less susceptible to errors, leading to a higher order of p dependency for small p. *Reproduced from [168] under the Creative Commons Attribution 4.0 International License [170].*

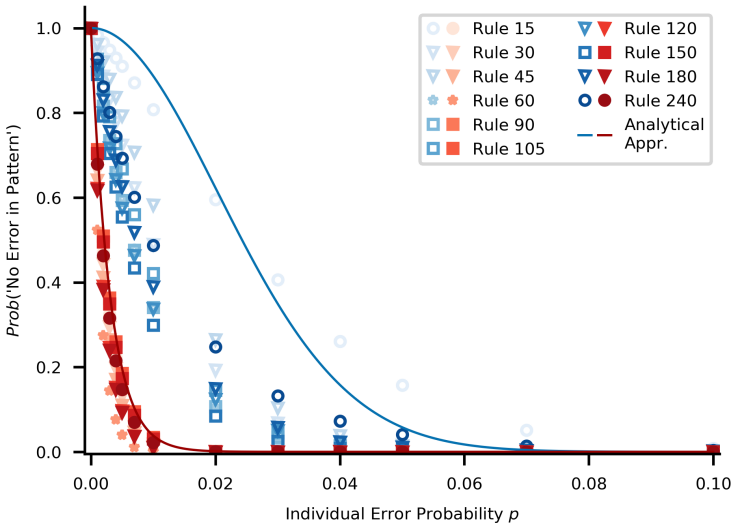


Figure 5.7 Robustness Against Errors and Error Correction for Fixed Boundary Conditions. Same as in fig. 5.6, but for fixed upper and lower boundary conditions instead of periodic. Correction works noticeably worse due to the disruptive influence of the fixed boundary conditions. Only for rule 15 and its color switches (cp. fig. 5.4) it is possible to correct errors more effectively (cp. [167]). *Reproduced from [168] under the Creative Commons Attribution 4.0 International License* [170].

to as instruction. To capture this newly defined dynamics, the patterning graph of each rule is mapped to a new graph with the nodes still being the patterns, but the edges now representing the transitions from a pattern to (if existing) the stationary pattern to which it is lead by the rule (fig. 5.8 (a), (b)). Adding these graphs for all rules together by keeping the nodes copying the edges into a common graph constructs the "attractor graph" for a given system topology and system size. As initial state, a homogeneous bulk state is chosen. Form other initial conditions, this initial condition can always be reached by the 0 rule

Since not all patterns are stationary states to some rules (except for the identity rule), some patterns are not reachable. Calculating the number of actually reachable patterns by exhaustive exploration of the pattern space shows that it can be well approximated by an exponential growth (fig. 5.8 d), which is consistent with an analytical approach showing that the number of attractors of finite CA grows with the exception of the identity rule more slowly with system size L than 2^L [177]. Even though the number of reachable patterns grows in exponential fashion, the number of instructions necessary still follows a linear relationship (fig. 5.8 (e)), similar to the previous patterning scheme.

Ramalho looked at the pattern formation process in the sense of this section from the viewpoint of Kolmogorov complexity by viewing the pattern formation process as means to compress the information needed to generate a specific pattern [162, 168]. The number of instructions is then up to a constant an upper bound for the Kolmogorov complexity of the pattern and it can be hypothesized that, since elementary cellular automata are a minimal model for algorithmic generative models, it is a good upper bound to universal complexity [162]. Here, it is compared with the pattern complexity calculated as described in section 3.3.4. Figure 5.9 demonstrates that both are only weakly related in the sense that low values of one measure do not occur together with high values of the other measure, but otherwise no additional structure can be seen. The first aspect can be explained by the fact that pattern which needs the lowest number of instructions exhibit some periodicity as naively expected from patterns which are easily compressed [162, 168]. That otherwise no structure is evident is not a contradiction of two measures for a Kolmogorov complexity, since both can only be seen as upper limits and are up to a constant determined by the underlying algorithm.

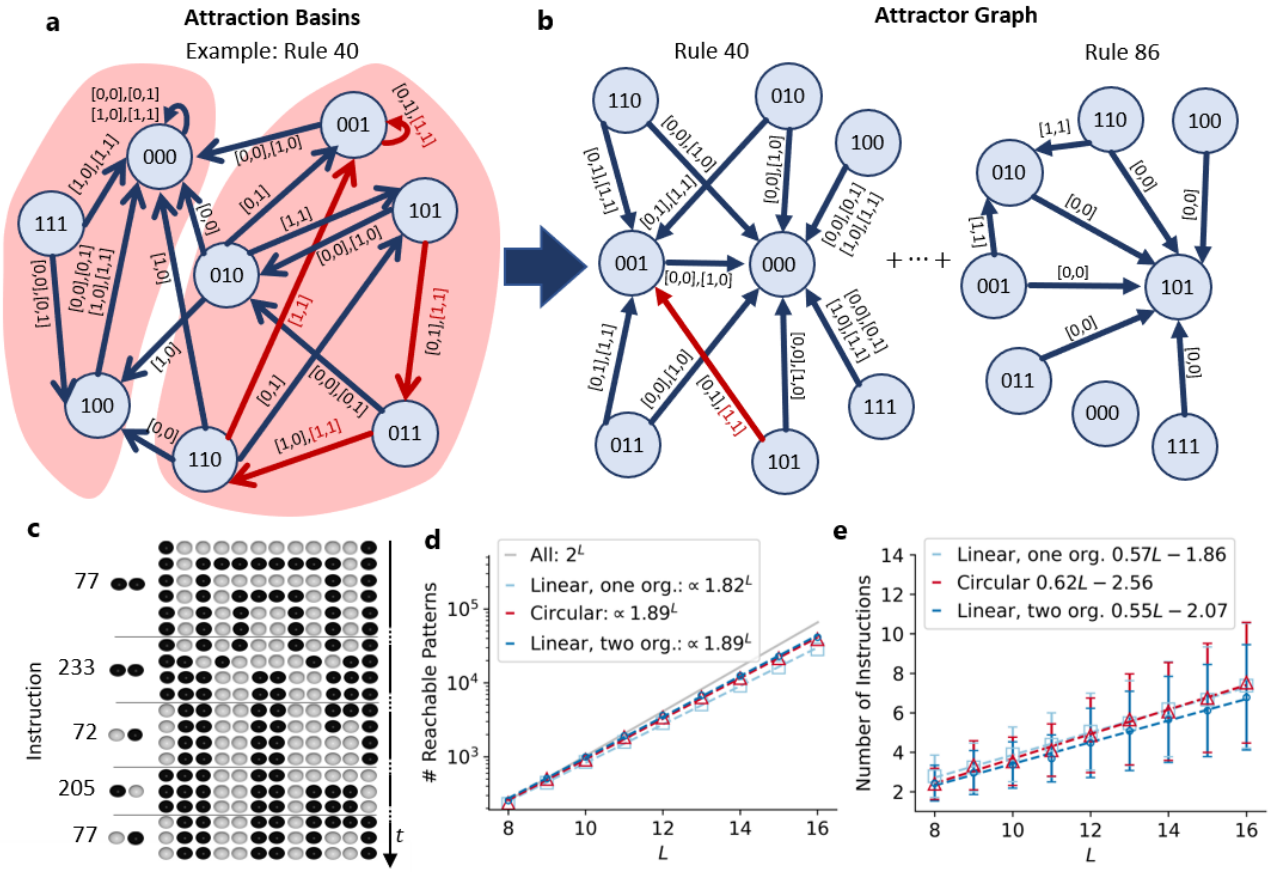


Figure 5.8 Attractor Graphs. (a) Attractor graphs are generated by calculating the attraction basins of fixed points of patterning graphs for fixed organizer inputs. (b) Each element of the attraction basin is then directly mapped to the fixed point in the contribution to the attractor graph of the specific rule. As an example, the transition marked in red arrows from pattern "101" to pattern "001", which is stationary for rule 40, using $[1, 1]$ as organizer inputs, is equivalent to the red arrow in (b). The attractor graph comprises contributions from all CA rules and organizer inputs for corresponding topology. (c) Exemplary trajectory in the attractor graph, changing rules and organizer inputs (instructions) only after having reached a fixed point. Pattern 011001101001 is formed from a homogeneous initial pattern. (d) Number of reachable patterns from a homogeneous initial condition as a function of the system size from exhaustive calculations (symbols) as well as exponential fits (dashed lines), which all have a base smaller than 2. (e) Number of instructions necessary to reach a given state calculated exhaustively (symbols) with linear fits. *Reproduced from [168] under the Creative Commons Attribution 4.0 International License [170].*

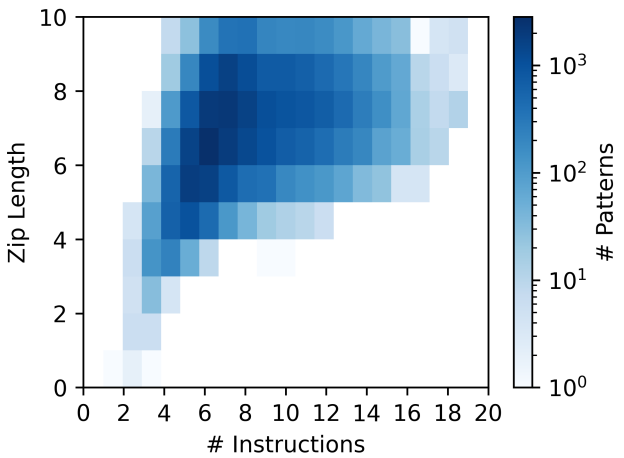


Figure 5.9 Pattern Complexity Number of patterns with a zip length and number of instructions needed to reach all reachable patterns for two organizer cells and $L = 16$. Zip length is calculated by applying the zlib module of python to a string representation of the pattern and then subtracting the minimum resulting compressed size, compare section 3.3.4.

5.7 Discussion and Further Avenues of Research

In this chapter, a theory for programmable pattern formation was developed. First, a programmable rule was defined as allowing to steer a patterning process starting from any initial pattern to any target pattern by appropriately choosing organizer inputs. A computational approach by exhaustive analysis of the space of patterns for small system sizes using network theory identified a limited set of rules that were programmable. Estimates for the spatial and temporal Kolmogorov complexity as well as single point entropies of the rules alongside with the degree statistics of the patterning graphs were identified as possibly useful filtering criteria for the search of programmable rules in larger rule spaces. The patterning principles employed by the rules where discussed, ranging from a Chinese's whispers or Mexican wave [178] type of information relay over complex information processing to interference type rules for circular topologies. Bijectivity was identified as a sufficient criterion for programability of rules independent of system size or number of states. Much like in a game of Chinese whispers, errors are a common feature of biological pattern formation and have been addressed subsequently by introducing redundancy through extending the system in a second dimension over which local averaging procedures have been performed. Finally, the precises timing of the organizer protocol was relaxed at the expense of changing rules and it was shown that under these conditions, it was still possible to reach an, as function of the system size, exponentially growing amount of patterns in an, as function of the system size, linearly growing time.

The equivalent question in the theory of continuous systems is usually phrased as control theory [179, 180, 181]. In control theory, the notion of controllability is formalized similarly to programmability here as the abilities to steer a dynamical system to any desired state from any initial state. Recent advances focused on controllability of systems with complex topology yet simple linear dynamics [181]. In contrast, we used a simple topology but a more complex, highly nonlinear dynamics in an open loop system. For a discrete CA setting, a similar problem was addressed independently [166, 162, 167] in the framework of regional control of CA. In a deterministic setting, Boolean derivatives [112] and peripheral linearity have been employed as a conceptually similar vehicle to bijectivity to prove programmability of CA rule [182, 183], as well as a graph theoretical approach was undertaken to computationally investigate the question [184] which finally was generalized also in a probabilistic setting [185, 186].

How does the here discussed pattern formation mechanism via organizer cells compare with other mechanisms in the literature [176] like, e.g., the clock and wavefront mechanism [13] or traveling and phase waves [187]? For example, in the clock and wavefront mechanism the difference between exemplary investigated wild types and mutants seems to be the wavelength of the pattern, while the principal form seems to be invariant [29]. As demonstrated here, with organizer cells the full capability of complex, information processing subunits can be leveraged leveraged to achieve a complete adaptability of the pattern formation process to freely chose-able target patterns and not only changes in the pattern wavelength.

An example for a programmable dynamics may be the pattern formation during *C. elegans* vulva development, where an anchor ("organizer") cell steers the fate of 6 precursor cells [84]. Reverse engineering a possible CA rule from the proposed high level models yield a partially specified rule which is compatible with a bijective rule [167].

The results of this research can possibly be transferred to other fields ranging from locally communicating robots [188, 189] over DNA based CA implementations [167, 190], to nanoscale robotic arms [46]. For the latter, as proposed by Simmel [48], controlling an organizer cells could be achieved through small enough electrodes addressing only a single arm.

For future research, it seems promising to investigate the concept of programmability in 2D, yet an attempt for a direct transfer of the concepts developed turned out to be not successful [167]. Thus, it might be interesting to turn either to more relaxed forms of programmability, e.g., focusing only on a subset of cells that need to be in the correct state or surrendering the all-to-all mapping as exemplified in section 5.6. On the other hand, the external steering capacity could be strengthened by increasing the number of organizer cells and/or decreasing their mutual dependency. Other promising avenues might be investigating the influences of prepatterns [12, 116] and long range signaling.

6 How to Synchronize Local Oscillations Globally? - Pattern Formation and Synchronization

The main content of this chapter was the work of the author of this thesis with critical feedback by Mareike Bojer and supervision by Ulrich Gerland. Section 6.6.1 was developed together with Johannes Harth-Kitzerow, Torsten Enßlin, and Ulrich Gerland.

6.1 Introduction - Synchronization and Pattern Formation

With the minor exception in section 4.8.3, until now all used CA had synchronized updates, i.e., all cells updated exactly at the same time. While this is the common assumption, which is, e.g., used by Wolfram [42, 191] and Conway [41], the effects of asynchrony a CA has also been investigated - for a recent review see for example Fatès [192]. The space of possible versions of (a)synchronous updates in CA stretches between completely synchronous updates on one side and completely random updates, i.e., at each step in time a cell is picked at random and updated. In between these extremes, several possible update schemes are discussed in literature which usually vary

- the amount of cells synchronously updated at the same time,
- the amount of order (subsets) of cells follow when updating sequentially,
- how cells are assigned to the updating subset, e.g., with which probability,
- how long it takes to communicate the update to the neighbors, i.e., introducing a communication delay, and with what probability the update is communicated at all or to which subset of neighbors.

Exemplary models from literature [193, 194] employing different update schemes are presented in appendix A.6.1, another example in the context of this thesis can be found in section 4.8.3. The pattern formed can be strongly affected by the chosen type of update scheme and even undergo qualitative changes akin to a phase transition as function of, e.g., the probability of communicating the update to the neighbors or choosing the subset of cells which will be updated [194]. Thus, with the goal of reliably forming a pattern, it is interesting to look at how synchronization could be achieved in CA. Assuming that all cells have imperfect internal clocks and, as the focus is on pattern formation by local communication, no global external clock signaling them when to update, the question can be asked to what extent it is possible to keep CA globally synchronous just by local signaling.

6.2 A Model for Pattern Formation with Information Transmission Delay

6.2.1 Model Description

The model description is started by describing how a continuous oscillation model can be discretized in a biological setting and how this can be emulated with a discrete stochastic update. Then, the integration of such an update model with a CA is specified and a delay introduced.

To adapt a continuous time synchronization to a discrete CA setting, a continuous time oscillating clock is discretized into a pulse like behavior (fig. 6.1 (a)). In this effective model, only at discrete time intervals, representing, e.g., the maximum of the continuous phase model, the clock “ticks”. In a biological setting, reacting only to a specific part of this oscillation could potential be realized by highly nonlinear signal

processing functions [89], effectively acting as a filter mechanism. Also, an integrate and fire oscillation [195] like proposed for firefly flashing, where a variable increases autonomously over time with additional jumps added by firing neighbors until a threshold is reached at which the clock "fires" and resets itself, could be possible. To simplify this more complex discretization model, it is assumed that successive cycle times τ are drawn from a distribution $P(\tau)$ (fig. 6.1 (a)) to emulate the potentially noisiness in the oscillation phase. For $P(\tau)$ it is assumed that the cycle time is originating from a sum of random variables specifying the time of individual chemical reaction which in total comprise a full oscillation. Thus, a Gaussian distribution with mean μ and standard deviation σ is chosen using the central limit theorem as justification. Each cell is modeled with one of these clocks. The number of ticks a clock has performed will be labeled as generations.

At each tick of the clock in a cell, the cell updates its state according to a cellular automaton rule processing the states of the neighbors and its own state. If the information about the cell state update is transmitted instantaneously, even extremely small temporal differences lead to cells already taking the "new" neighbor states into account when updating. Consequently, the CA dynamics is effectively strongly asynchronous even if the first "tick" of all clocks is synchronized due to the synchronous initial condition.[1] This is not a desirable property for a predictable pattern formation close to a synchronous setting, instantaneous information transmission is additionally not a realistic assumption in a biological setting since the transfer of molecules within a cell to its boundary in relevant concentrations takes time. Thus, a constant temporal delay of the magnitude t_{delay} (fig. 6.1 (b, c)). It models the time for information processing and transmission between neighboring cells, is introduced.

6.2.2 Observables and Parameter Choices

After having defined the model, the next step is to define appropriate observables. From the position of pattern formation in CA it is of interest at which point the formed pattern diverges from a pattern formed by an identical rule and initial condition but synchronous update. This point in time can be determined by comparing the kymographs of a CA with synchronous update and a CA with an update scheme as defined in the previous section - the first generation at which those two kymographs differ is defined as

$$t_{First\ Difference} = \arg\min_t (x_i^{sync}(t) \neq x_i^{async}(t)). \tag{6.1}$$

with x_i being the state of cell i. While the CA dynamics might be effectively asynchronous, this might not be reflected in the kymograph. As an example, for rule 0, the kymograph will after the first generation always be 0, regardless if the updates happen in a synchronous or asynchronous manner. To quantify the point where the dynamics is first asynchronous, $t_{First\ Asynchrony}$ is defined as the first time the temporal difference between the update of neighboring cells exceeds the delay,

$$t_{First\ Asynchrony} = \arg\min_t (\Delta t(t) > t_{delay}) \tag{6.2}$$

with Δt being the temporal difference between the update of neighboring cells.

As simulation parameters the following values are used if not stated otherwise:

- Gaussian distributed update time with mean $\mu = 1$ and noise $\sigma = 0.016$
- Temporal delay of $t_{delay} = 0.1$
- 1D system with periodic boundary conditions and system size $L = 20$
- 2 state random initial condition of the pattern

[1] If the mean of the update distribution μ is large compared to the standard deviation σ, the resulting update scheme can be expected to initially be a variant of a cyclic update scheme in which all cells perform the update of a generation before any cell performs the update of the next generation. The order of update within a generation can be expected to neither be random nor fixed, but have some finite temporal correlation.

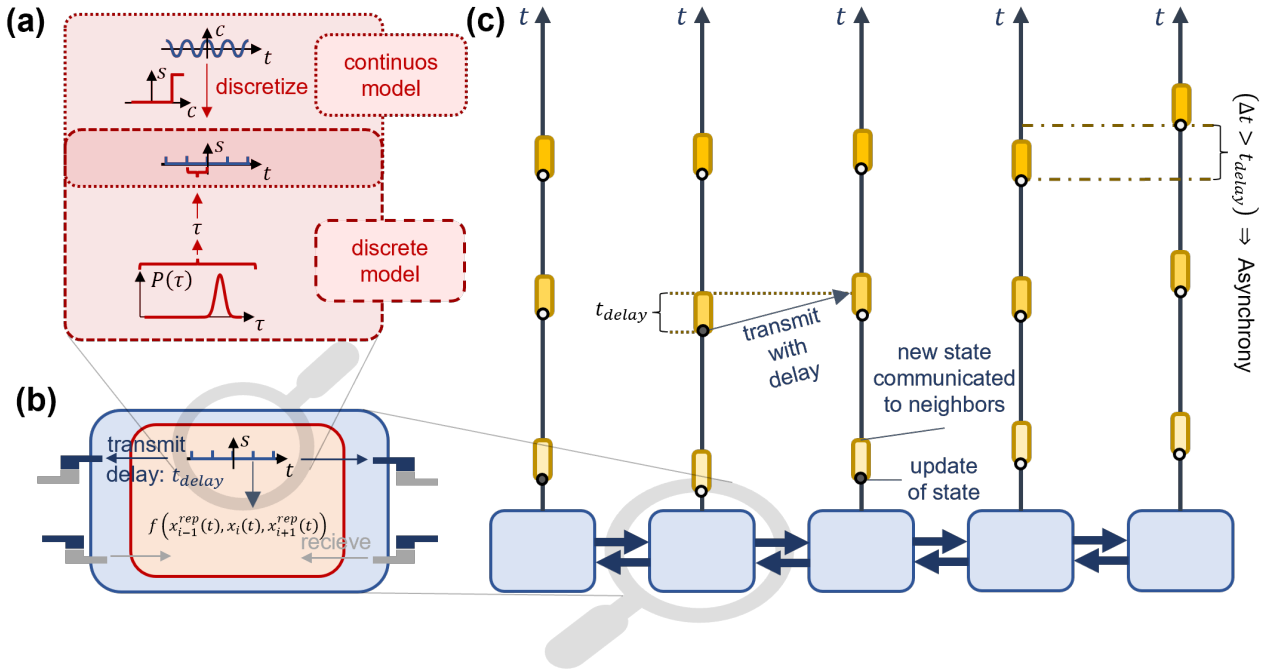


Figure 6.1 A Model for Synchronization in Pattern Formation (a) **Inside of a cell.** The oscillations of gene expression is usually modelled a continuous oscillation of a concentration c as a function of time (top). This oscillation may have an average mean frequency with fluctuations around this mean. If a signal s is emitted only for large concentrations or only detected for large concentrations, equivalent to applying a high pass filter $c\left(s\right)$, the continuous model is effectively discretized in a pulse like behavior representing the "ticks" of the oscillation clock (centre). This pulse like behavior can also be directly produced by a discretized model drawing the times of the pulse from a probability distribution $P\left(\tau\right)$ (bottom). (b) **The cell and its neighbors.** The ticks mark the update times of the cellular automaton. The state of the cellular automaton is transferred with a delay t_{delay} to the neighbors of the cell. The CA update $x_i\left(t+1\right) = f\left(x_{i-1}^{rep}\left(t\right), x_i\left(t\right), x_{i+1}^{rep}\left(t\right)\left(t\right)\right)$ is conducted with the current state of the cell i itself as well as the last states received from the neighbors $\left\{x_{i-1}^{rep}, x_{i+1}^{rep}\right\}$. These states might not be the current states of the cells $i+1$ and $i-1$, but due to the delay cell i has not yet received an update. (c) **Tissue context and observable.** Depicted are, for a set of cells (bottom), events as a function of time (vertical axis for each cell). Each yellow rectangle with rounded corners symbolizes an update with the lower edge corresponding to a ticking of the clock and the execution of the CA and the upper edge corresponding to how long it takes to communicate the state change to the neighboring cells (t_{delay}). The dark or light colored circle at the lower edge of the yellow rectangles records the state of the cell right after the update. If the temporal difference between the updates of neighboring cells of the same generation Δt is larger than the delay t_{delay}, then the update of the CA is effectively asynchronous.

- All cells will start in perfect synchrony, i.e., the main question discussed will be if synchronization is conserved.

As a first estimation, the temporal difference between two neighboring cells after a single temporal update step follows a Gaussian distribution with mean 0 and standard deviation $2\sigma = 0.032$. Thus, the probability that 2 cells are synchronous after a single step in time is 98.7%. Each run is conducted for 1000 steps in time. If the system is still synchronous and/or there is no difference between the synchronous and asynchronous pattern, $t_{First\ Asynchrony}$ and $t_{First\ Difference}$ are set to 1001. For both measures, averages over 2000 independent simulation runs are calculated.

6.2.3 Simulation Results

The aim of this section is to investigate how the delay can buffer against asynchrony due to the noise in the update timing, to analyze which CA rules are robust in the context of asynchrony and identify the common properties of those rules.

To quantify the buffering effects of the delay, in fig. 6.1 (a), the ensemble averaged time to the first asynchrony $\langle t_{First\ Asynchrony} \rangle$ is plotted as a function of the standard deviation of the update time σ for different delay times t_{delay}. As can be expected, the larger the noise and the smaller the buffering effect of the delay, the shorter the time until $\Delta t > t_{delay}$ for a pair of neighboring cells, i.e., the time until the system is asynchronous.

More interesting is how different CA rules are affected by a possibly asynchronous dynamics. In fig. 6.2 (b)-(f), the average time to the first difference in the patterns between the potentially asynchronous and completely synchronous dynamics $\langle t_{First\ Difference} \rangle$ is shown for all elementary 2 state CA rules for all heuristic Wolfram classes (cp. section 3.3.2) in fig. 6.2 (b), as well as separately for the individual classes in fig. 6.2 (c) - (f). Class 1 rules, which consist of rules which quickly converge to a homogeneous final state, have in general large values for $\langle t_{First\ Difference} \rangle$ (fig. 6.2 (c)), as can be expected because reaching a steady state quickly protects the dynamics from asynchronous updates. Class 2 rules, which exhibit "a simple stable or periodic structure" [43, p. 17] in the long term behavior, partially also have large values for $\langle t_{First\ Difference} \rangle$ (fig. 6.2 (d)), while the chaotic class 3 rules and class 4 rules typically with "complicated, localized structures, some propagating" [43, p. 17] in general have low values for $\langle t_{First\ Difference} \rangle$. The behavior for different classes is consistent with the expectations stated when defining our observables.

To substantiate these expectations further, the values for typical classification quantities (cp. section 3.3) as well as typical attractor properties as a function of $\langle t_{First\ Difference} \rangle$ are shown in fig. 6.3 and fig. 6.4. While properties depending only on the rule-table itself (λ_0, λ_1, μ, K_R) show no obvious tendencies, both temporal Kolmogorov complexity and temporal entropy need to be low for rules with large $\langle t_{First\ Difference} \rangle$, which is again consistent with a steady state behavior. Since both spatial Kolmogorov complexity and spatial entropy have a range of different values for large $\langle t_{First\ Difference} \rangle$, the patterns formed with rules which are stable against asynchrony may appear simple or complex. The synergies S_1 and S_2 show no obvious dependencies, suggesting that the integration of information does not play a significant role regarding synchrony. Looking at the attractor properties of each rule (fig. 6.4) confirms the picture already discussed: Rules with small average attractor sizes, hinting at steady states and not limit cycles, and small transient length are potentially robust against asynchrony, while for large average attractor sizes and large transient length this robustness can not be achieved (fig. 6.4 (a)). The few rules which have average attractor sizes of around 10 and a large value for $\langle t_{First\ Difference} \rangle$ have a larger number of attractors, between 100 and few 1000s, also consistent with the possibility of having several fixed point attractors despite the large average attractor size (fig. 6.4 (b)).

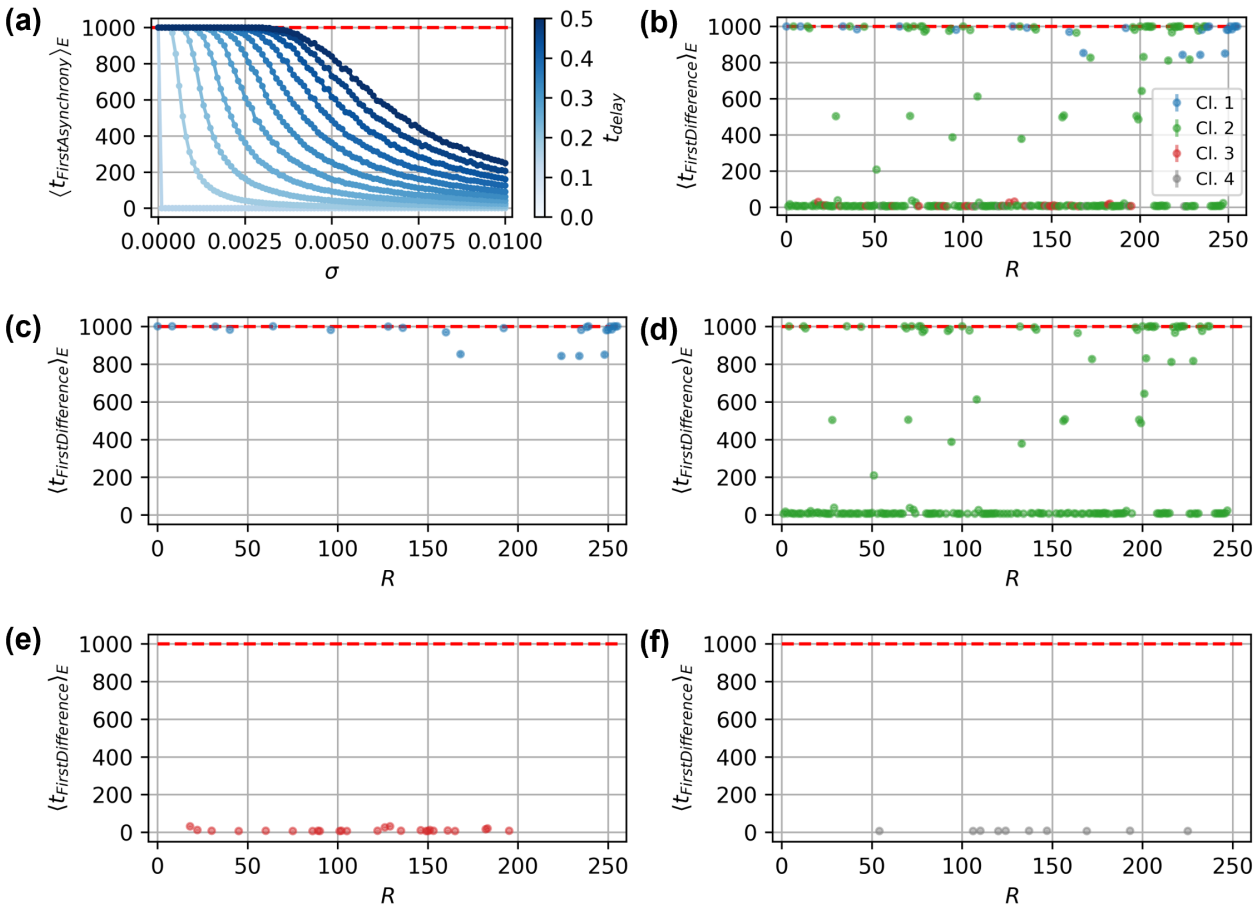


Figure 6.2 Synchronization with Delay - Dependency on Noise, Delay and Rule. (a) Mean time to the first asynchrony $\langle t_{First\ Asynchrony} \rangle$ as a function of the standard devation of the Gaussian update distribution for different delays t_{delay}. (b) Average time to the first difference $\langle t_{First\ Difference} \rangle$ as a function of the rule for different Wolfram classes (blue class 1, green class 2, red class 3, grey class 4). (c)-(f) same as in (b), but only class 1 (c), class 2 (d), class 3 (e), class 4 (f) shown per panel.

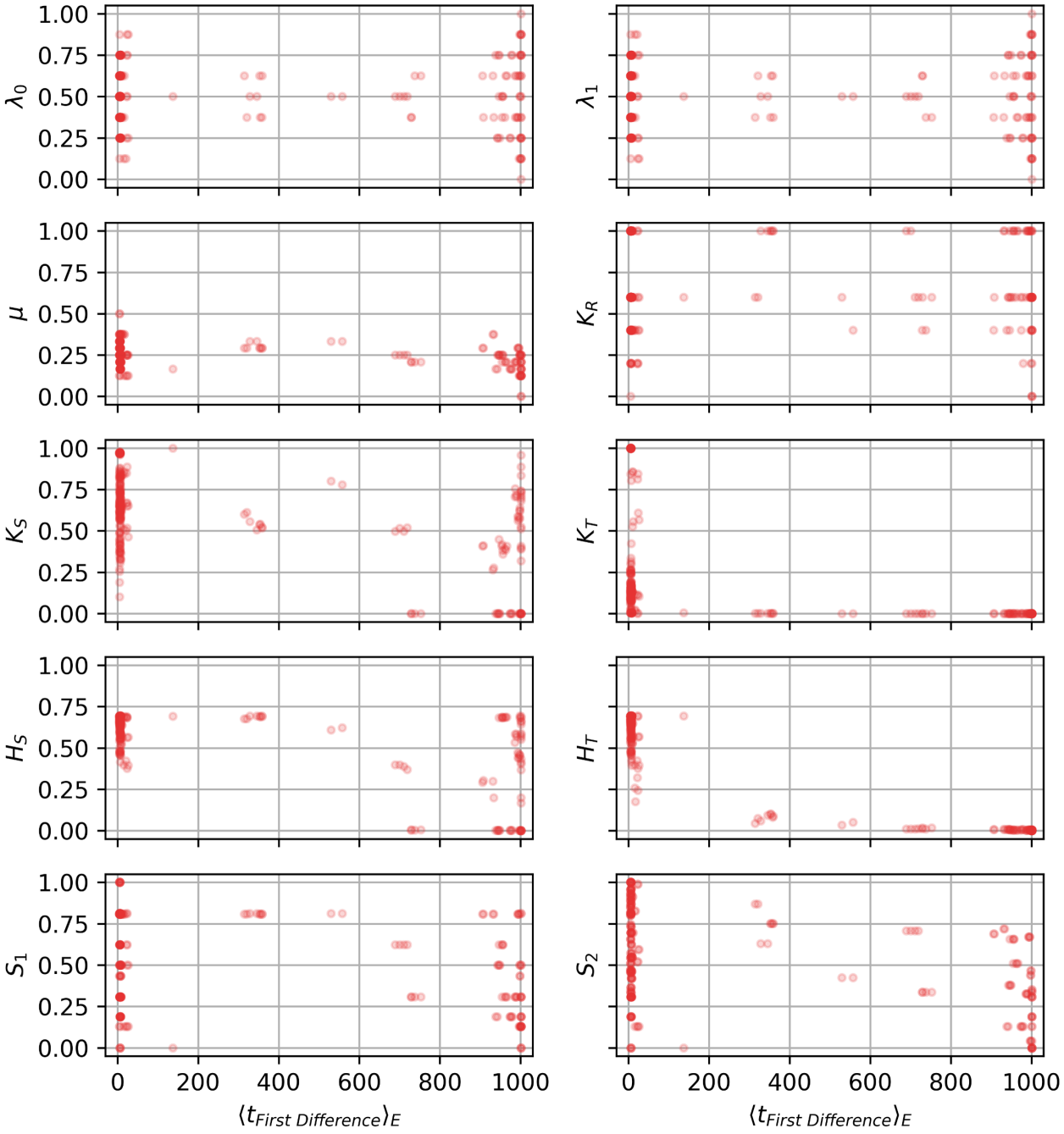


Figure 6.3 Synchronization with Delay - Classification Quantities. Classification quantities from section 3.3.3 and section 3.3.4 with numerical values in appendix A.2.2 as a function of the average time to the first difference $\langle t_{First\ Difference} \rangle$.

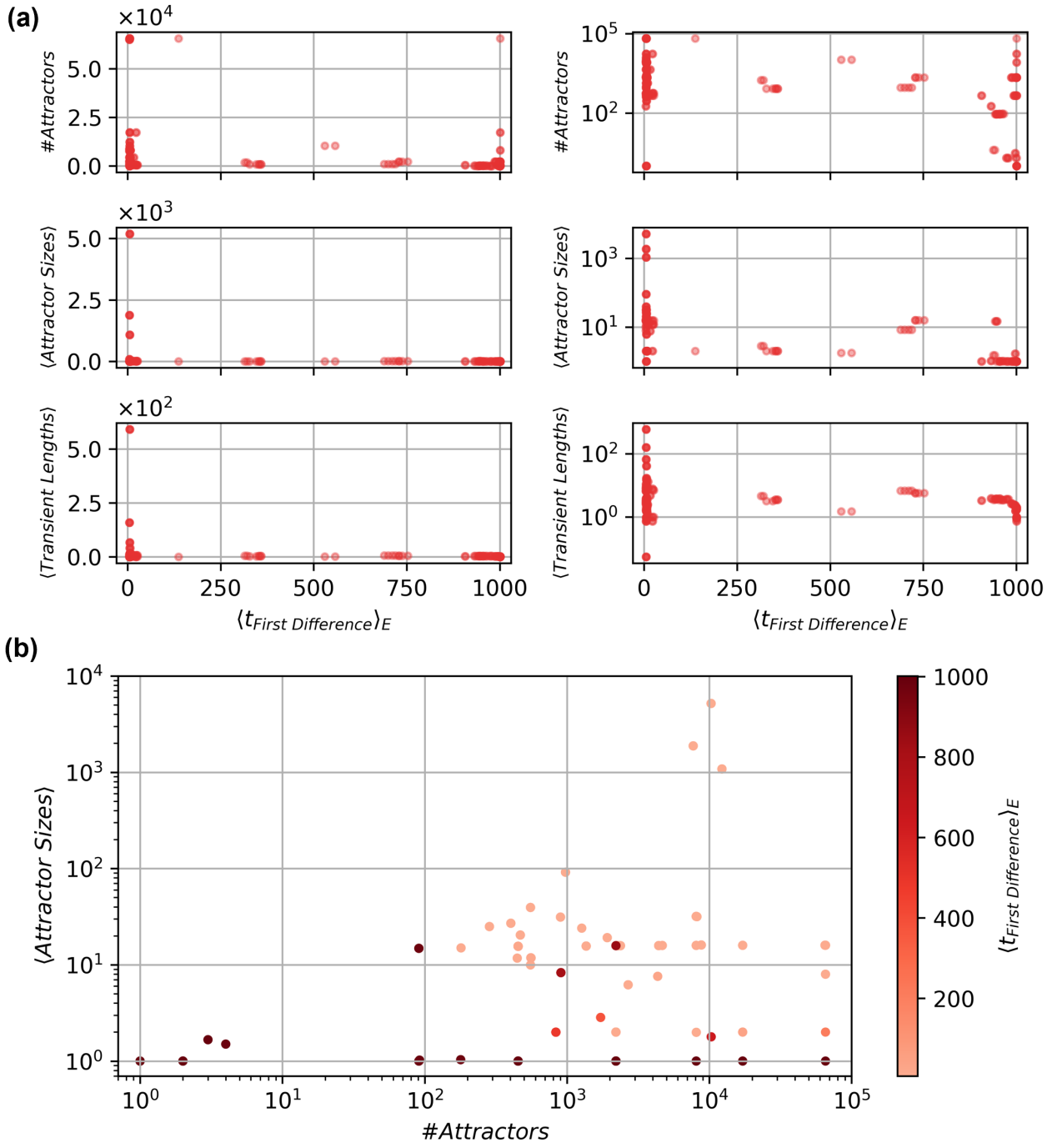


Figure 6.4 Synchronization with Delay - Attractor Properties. (a) Number of attractors, average attractor sizes and average transient lengths for the CA dynamics as a function of the average time to the first difference $\langle t_{First\ Difference} \rangle$ for the elementary cellular automata rules with linear (left column) and logarithmic (right column) scaling of the "y-axis". (b) Scatter plot of the number of attractors and average attractor sizes, color coded by $\langle t_{First\ Difference} \rangle$.

6.3 Conservation of Synchrony with Temporal Correction Schemes

6.3.1 Introducing a Correction Scheme

In the previous section, a model for the conservation of synchronization in cellular automata with delay was introduced and its interplay with patterning was discussed. In this section, an additional possibility for damping the temporal deviation between neighboring cells by using correction schemes are introduced and characterized. For simplicity, the patterning through the rules is ignored during this section and reintroduced in section 6.4. The section is organized as follows. First, in this subsection a general overview of the components of the model is given together with the standard choices of the components. In the following subsections, variations of these choices are explored systematically.

To introduce a correction for the diffusive drifting of the individual update times of a cell, the temporal difference between the update of neighboring cell and the cell itself Δt is used. It is assumed that cells correct for the delay when calculating Δt. At each point in time when the cell sends an information about its temporal update to its neighbors, again assumed to be t_{delay} after its own update, a cell checks if it has received information about ticks of the neighboring cells (which reach it also with a delay of t_{delay}). The ticks are processed if they reach the cell within a detection window of size t_{detect}, i.e., if

$$\Delta t < t_{detect}. \tag{6.3}$$

If so it updates the time of its own next update or the ones of the neighbors or both by

$$t_{s+1,i}^{next,\text{after update}} = t_{s+1,i}^{next} + g\left(\Delta t\right) \tag{6.4}$$

(fig. 6.5 (b)).[2] The detection window t_{detect} is introduced to reduce confusion about if the ticking of a neighbor belongs to the previous generation or already to the next generation. Thus, information about neighboring "ticks" is only processed if it is received within a time-frame $[-t_{detect}, 0]$ around the "tick" of the cell itself (fig. 6.5 (c)). In a biological setting, this is similar to an "AND" gate, only activated if both the ticks of a cell and its neighbor are sufficiently close in time.

The core idea of the model is now specified. Apart from the numerical values of the parameters, left to define are the correction function $g\left(\Delta t\right)$, the probability distribution $P\left(\tau\right)$, and the topology and neighborhoods. The standard choice for the correction function $g\left(\Delta t\right)$ is directly proportional to the deviation $g\left(\Delta t\right) = m\Delta t$ with m being the proportionality constant specifying the correction strength. As grid, a static, 1D version of size L with periodic boundary conditions and cells only connected to their nearest neighbors is the standard choice with other choices explored in section 6.3.4. As initial condition, all cells will start in perfect synchrony. Throughout the rest of the section, the following parameters will be chosen (all in arbitrary units which are consistent with each other, e.g., for parameters carrying a temporal unit it will be the same choice of units for all):

m	t_{detect}	μ	σ	t_{delay}	L	S_{max}	N_E
0.3	0.15	1	0.016	0.1	20	1000	2000

Table 6.1 Standard Parameter Choices Used in the Simulation m is the proportionality constant, t_{detect} the detection window size, μ and σ are mean and standard deviation of the Gaussian temporal update distribution, t_{delay} the delay, L the system size, S_{max} the maximum number of simulated generations, and N_E the ensemble size.

As the initial condition for all cells is to start in synchrony, the main question discussed will be if synchronization is conserved. In growing systems, this can be a natural assumption if the daughters of a dividing cell inherit the state of the mother cell and the number of initial cells is low - this will be explored in section 6.3.4. Additionally, a different approach without this restriction will be discussed in section 6.6.1.

[2] This is closely related to "pulse coupling" in other synchronization models where oscillators influences each other only in specific phases of the oscillation cycle [93].

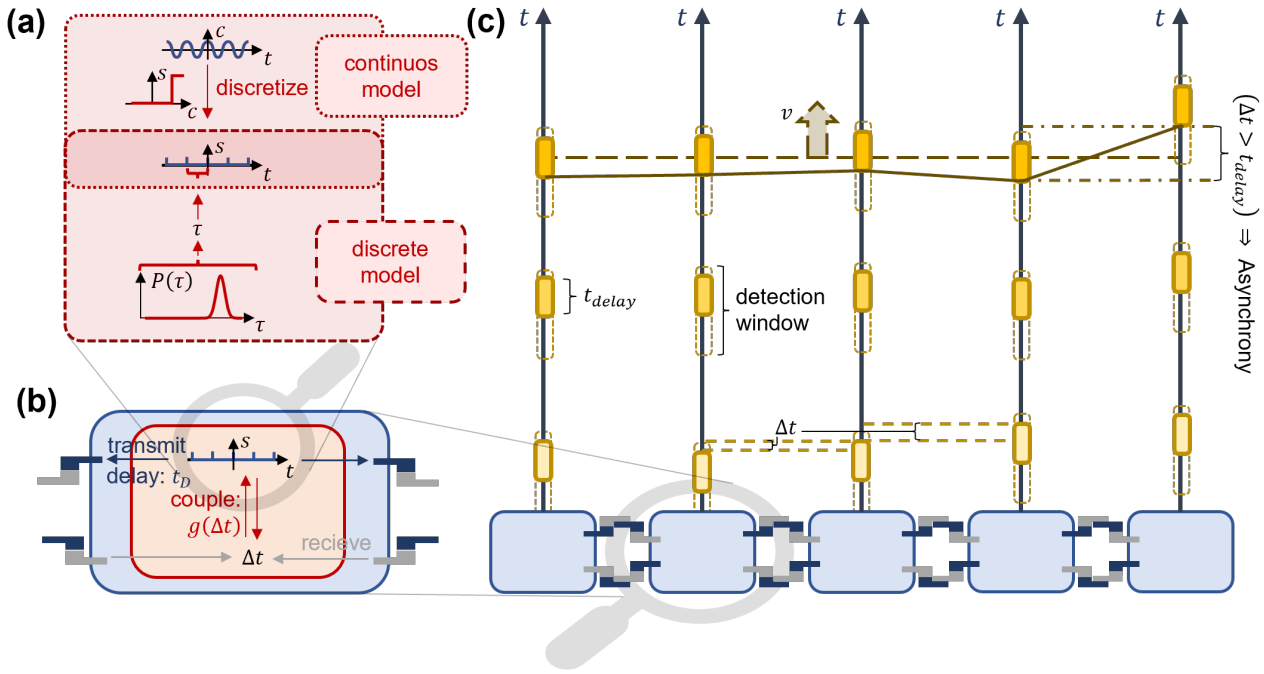


Figure 6.5 A Model for Synchronization with Correction in Pattern Formation. Same figure as fig. 6.1, but without the CA and with included correction. (a) Unchanged from fig. 6.1. (b) An individual cell transmits the time point of its own update with a delay t_{delay} to its neighbors, and receives their updates with the same delay. From the difference between the time of its own update and the update time of its neighbor Δt the cells calculates a correction $g\left(\Delta t\right)$ to its next update time if the update received is within the detection window t_{detect}. (c) Additionally to fig. 6.1 (c), also the detection window is shown as yellow rectangle with curved edges and a dashed line. As further observables to the time to first asynchrony (brown dash dotted line), examples for the temporal horizon of a generation (brown solid line) and average advancement of a generation (brown dashed line).

6.3.2 Correction Algorithm

To correct the diffusive drift of the temporal updates of the individual cells, different correction schemes $g(\Delta t)$ are applied:

- *Proportional* correction: $g_p(\Delta t) = m\Delta t$. This assumes that the cell is able to use the magnitude of the difference to the neighbor and adjust the update timings accordingly, scaled by a proportionality constant m. As stated above, it is also the standard choice for the correction scheme in the following chapters.
- *Constant* correction: $g_c(\Delta t) = m\,\mathrm{sign}\,(\Delta t)$. This assumes that the cell is only able to detect if it is ahead or behind in time compared to its neighboring cells and then adjusts the update timing by a constant m.
- *Kuramoto* correction: $g_K(\Delta t) = \frac{m}{2\pi} sin\,(2\pi\Delta t)$. This assumes a Kuramoto type correction [196] commonly used when discussion synchronization type problems. For small Δt, the Kuramoto correction scheme can be expected to behave similarly to the proportional correction scheme since

$$g_K(\Delta t) \approx m\Delta t - \frac{2m\pi^2}{3}\Delta t^3 + \ldots \approx g_p(\Delta t). \tag{6.5}$$

Additionally, the cells have the option to correct their own update timing or the update timing of their neighbors, equivalent to correcting their own update timing with information from the interval $[-t_{detect}, 0]$ or $[0, t_{detect}]$, respectively, or using both options. The options will be termed *backward*, *forward* and *symmetric* correction. Thus, in total 9 different update schemes and 2 basic parameters m and t_{detect} need to be discussed. The general strategy in the following sections will be to only change a few of these parameters and keep the others at their default values, which are chosen to be: A proportional, symmetric correction scheme with $m = 0.3$ and $t_{detect} = 0.15$.

Correction Strength m

As first variation the influence of different correction strengths m of the correction function is investigated. In fig. 6.6 (a) the observable $\langle t_{First\ Asynchrony}\rangle_E$ is chosen and only the proportionality constant m is varied. The resulting curve is unimodal with its maximum around $m = 0.3$. As first key result it should be noted that $\langle t_{First\ Asynchrony}\rangle_E$ at this maximum is drastically larger than $\langle t_{First\ Asynchrony}\rangle_E$ without an applied correction scheme, corresponding to the value at $m = 0$ in this case, meaning that the correction scheme works successfully. Second, the position of the maximum closer to $m = 0.5$ than to $m = 1$ can be explained by imagining a system of just 2 cells - then values close to $m = 1$ would correspond to the cells more or less exchanging update times, while lower values would bring the cells closer to each other.

In real biological systems, it can be expected that not all cells behave exactly identical but there may be systematic, temporarily constant but spatially varying deviations how cells react to inputs from other cells. In order to investigate how such deviations have an effect on the ability of the correction scheme to keep the system synchronous, the corrections strength is modified by adding a Gaussian random variable to $m_{i,j} = m + \eta\,(0, \sigma_m)$ for each individual connection between two neighboring cells i and j. In fig. 6.6 (b), $\langle t_{First\ Asynchrony}\rangle_E$ is shown as a function of σ_m for otherwise default values. The correction scheme is able to introduce a remarkable stability given that (a) already a single asynchrony between two cells is enough to render the system asynchronous and (b) with approximately 60% chance a connection has a number added that is larger than $2\sigma_m$ for the default values with $L = 20$.

The main purpose of the correction scheme is to correct the Gaussian update noise parameterized by the standard deviation σ. A priori, it is not clear if a proportionality constant for which $\langle t_{First\ Asynchrony}\rangle_E$ is large for a given noise level also produces large $\langle t_{First\ Asynchrony}\rangle_E$ for other noise levels. To check, in fig. 6.6 (c) $\langle t_{First\ Asynchrony}\rangle_E$ is plotted as a function of σ for different choices of the proportionality constant m, showing that for the investigated parameters the curves around the optimum choice do not overlap in general, indicating that the choice of m is not sensitive to the noise.

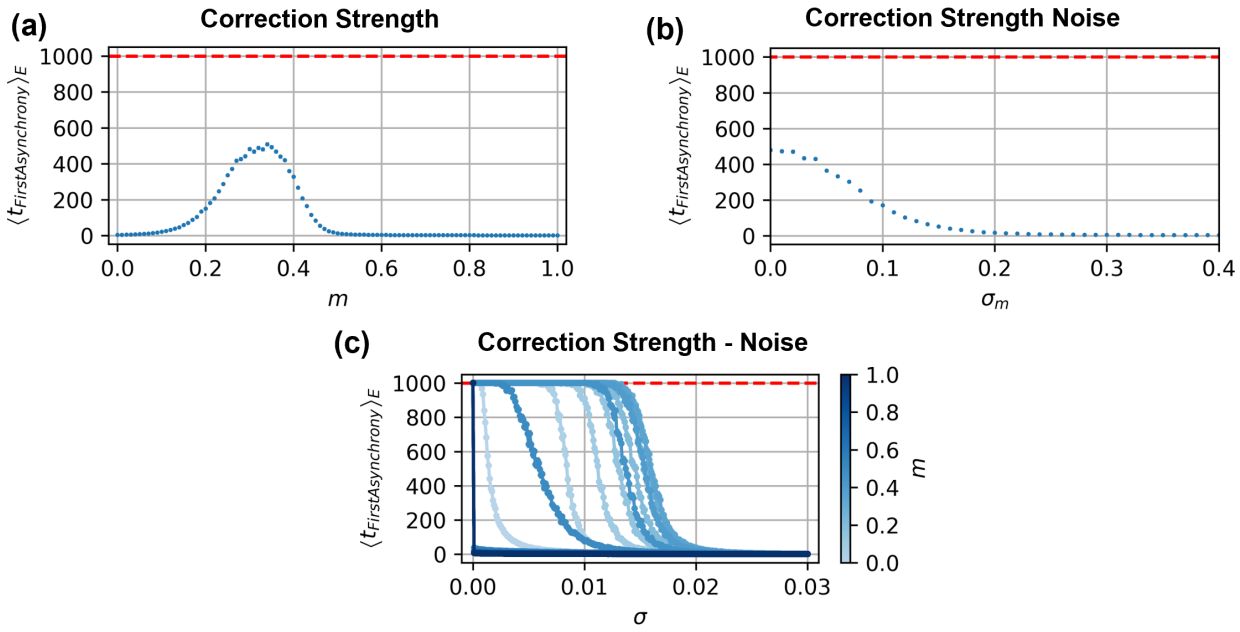


Figure 6.6 Correction Scheme - Proportionality Constant. (a) $\langle t_{First\ Asynchrony} \rangle_E$ as a function of the proportionality constant m for otherwise default parameters (blue curve) and maximum number of generations 1000 indicated by the dashed line. $\langle t_{First\ Asynchrony} \rangle_E$ shows a unimodal curve with the maximum around $m = 0.3$, the standard choice for m. $m = 0$ correspond to a system without correction. In the case of $m = 1$, a symmetric proportional connection leads to an exchange of the relative asynchrony, i.e., if cell i was ahead of cell $i+1$ by Δt, it will after the correction be behind by Δt rendering correction effectively useless. More effective values can be found between these two extreme values for m. (b) $\langle t_{First\ Asynchrony} \rangle_E$ as a function of the standard deviation of σ_m of a temporarily constant, spatially varying Gaussian random variable added to the correction proportionality constant $m = 0.3$. As a general trend, the larger the noise added, the lower is $\langle t_{First\ Asynchrony} \rangle_E$. Yet, compared to original value of $m = 0.3$, the tolerated noise where there is still a significant correction is large. (c) $\langle t_{First\ Asynchrony} \rangle_E$ as a function of the standard deviation of the update noise σ for different correction proportionality constants m. Again, intermediate correction proportionality constants provide the most effective corrections, however noises with $\sigma > 0.2$ are typically not effectively correctable. Around the optimal correction proportionality constant the exact choice of the proportionality constant does not strongly affect $\langle t_{First\ Asynchrony} \rangle_E$.

Update Schemes

In section 6.3.1, 9 possible correction schemes were described: proportional, constant and Kuramoto correction, each in the variant forward, backward and symmetric correction. In fig. 6.7, for these 9 possibilities and three different choices of σ, the expected time of first asynchrony $\langle t_{First\ Asynchrony} \rangle_E$ is plotted as a function of the proportionality constant m.

The proportional and Kuramoto correction schemes show nearly identical curves, indicating that the dynamics happens in the regime where the linear approximation eq. (6.5) of the Kurmaoto scheme is valid. For the constant correction scheme, m is not unit-less but in units of time and is the magnitude of the applied correction. Thus, it is not surprising that the optimal m values are different compared to the other correction schemes. Constant correction seems in general to perform worse than the other schemes, which is not surprising as for constant correction only the sign of the temporal differences between cells is taken into account compared to the full value for proportional and Kuramoto correction, providing less information for the correction scheme to use.

Forward and backward correction schemes produce similar behaviour for all cases, performing usually worse with regards to the maximal values and being effective for larger m values than symmetric correction schemes. The worse performance can be explained by the fact that when the correction is applied only one of the cells shifts its update in time. But this shift is usually larger than in the symmetric case, which may introduce stronger differences for the other neighboring cells. Similarly, because only one cell moves its update time, to achieve bringing two neighboring cells close together in time, the optimal m has to be double as large compared to the symmetric.

Comparing the curves for different noises (different rows of fig. 6.7), a similar behavior as in fig. 6.6 (c) can be found for all correction schemes: the optimal correction proportionality constant is independent of the noise.

When synchronizing distributed systems, an important consideration is how a correction scheme influences the global temporal propagation, i.e., is the average advancement of the temporal surface accelerated, decelerated or not influenced by the correction scheme (see e.g. [197]). To check this, first for each generation the spatial and ensemble average of the temporal surface is calculated (fig. 6.8 (a)). To measure the advancement per generation, the difference of the average time between subsequent generations is taken. In order to determine if the advancement is accelerated or decelerated compared to the system without correction, the mean of the update noise μ is subtracted from this difference and plotted as a function of the generation s (fig. 6.8 (b-d)). The resulting curves are usually independent of the generation (fig. 6.8 (c-d)) and can thus be fitted by a linear function $as + b$, where b indicates if a constant acceleration or deceleration is present and a detects all linear deviations as function of the generation. The examples in fig. 6.8 (c-d) show the fits for default values with (d) and without correction (c). A more systematic picture can be obtained from fig. 6.9, where similarly as in fig. 6.6 the fit parameter b and a (insets) are plotted as a function of the proportionality constant m for the different correction schemes and different choices of the noise. The fit parameter a is small in all cases, strengthening above statement of a constant update speed. The symmetric correction scheme introduces only small deviations of b from 0. However, forward and backward correction schemes lead to an deceleration or acceleration, respectively. While proportional and Kuramoto correction show again a similar dependency of b on m which simply behaves monotonically as function of m, in the case of constant correction, the curve is unimodal. The latter can be attributed to the constant size of the correction irrespective of the temporal deviations - if the correction is large even for small perturbations, it can bring the system close to a state where the temporal deviation is larger than the detection window - leading to no correction being applied at all. The dependency on the noise is limited to small quantitative changes - higher noise leads to a small increase of b for proportional and Kuramoto correction, as well as a shift of the position of the maximum for constant correction to smaller values supporting the above explanation of the unimodal distribution, but qualitatively no changes can be seen.

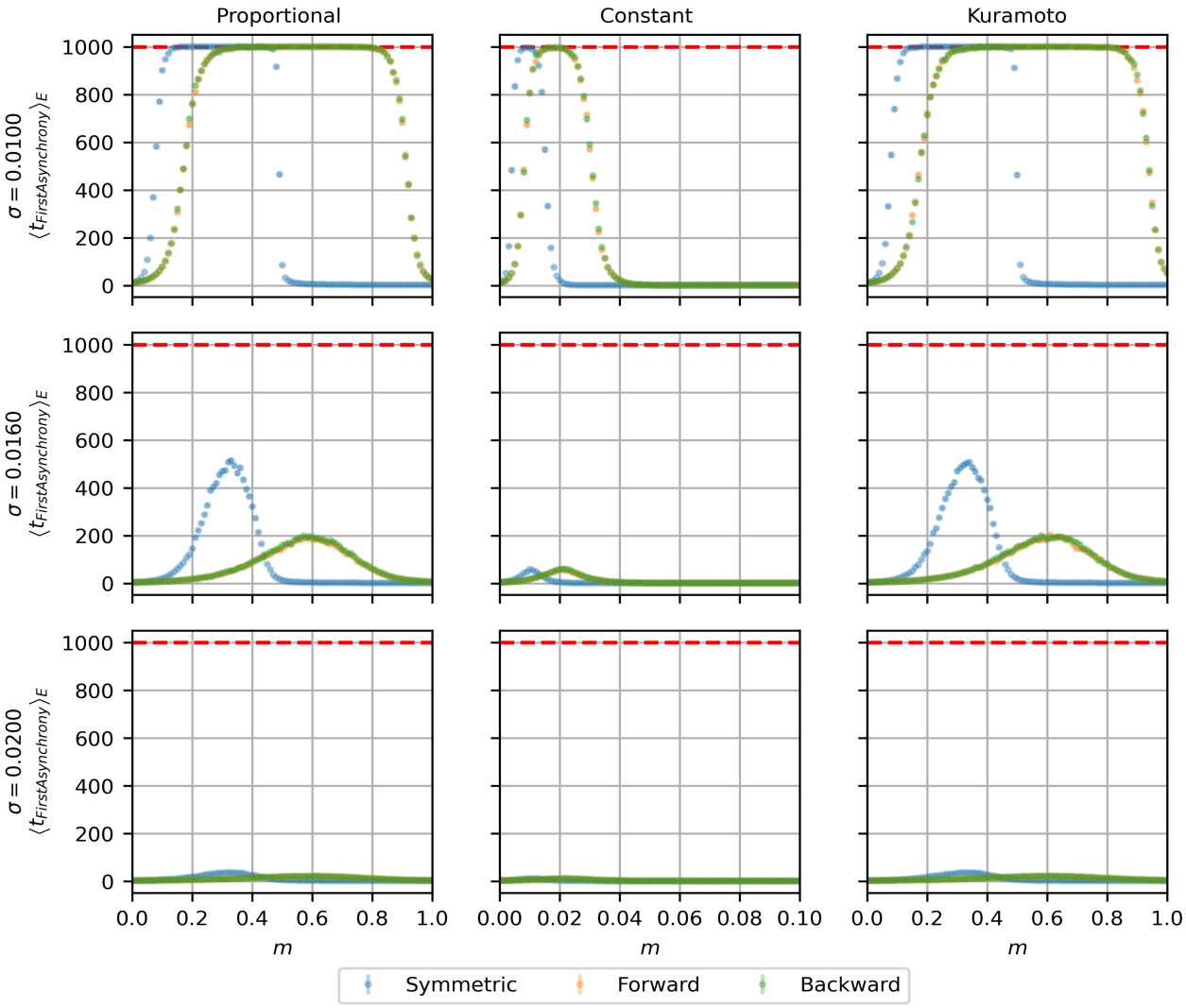


Figure 6.7 Correction Scheme - Type and Direction. $\langle t_{First\ Asynchrony} \rangle_E$ as a function of the proportionality constant m for proportional correction (first column), constant correction (second column) and Kuramoto correction (third column) as well as symmetric, forward and backward correction (different colors) and standard deviations of the update noise (different rows). Please note that while for proportional and Kuramoto correction m has no units, for the constant correction case m is in time units and corresponds directly to the magnitude of the applied correction. Forward and backward correction result in similar curves, offset by a factor of roughly 2 in m space to the symmetric correction and with lower maximal $\langle t_{First\ Asynchrony} \rangle_E$. Proportional and Kuramoto correction show nearly identical curves, indicating that in the case of Kuramoto correction Δt is in the regime where the linear approximation is valid.

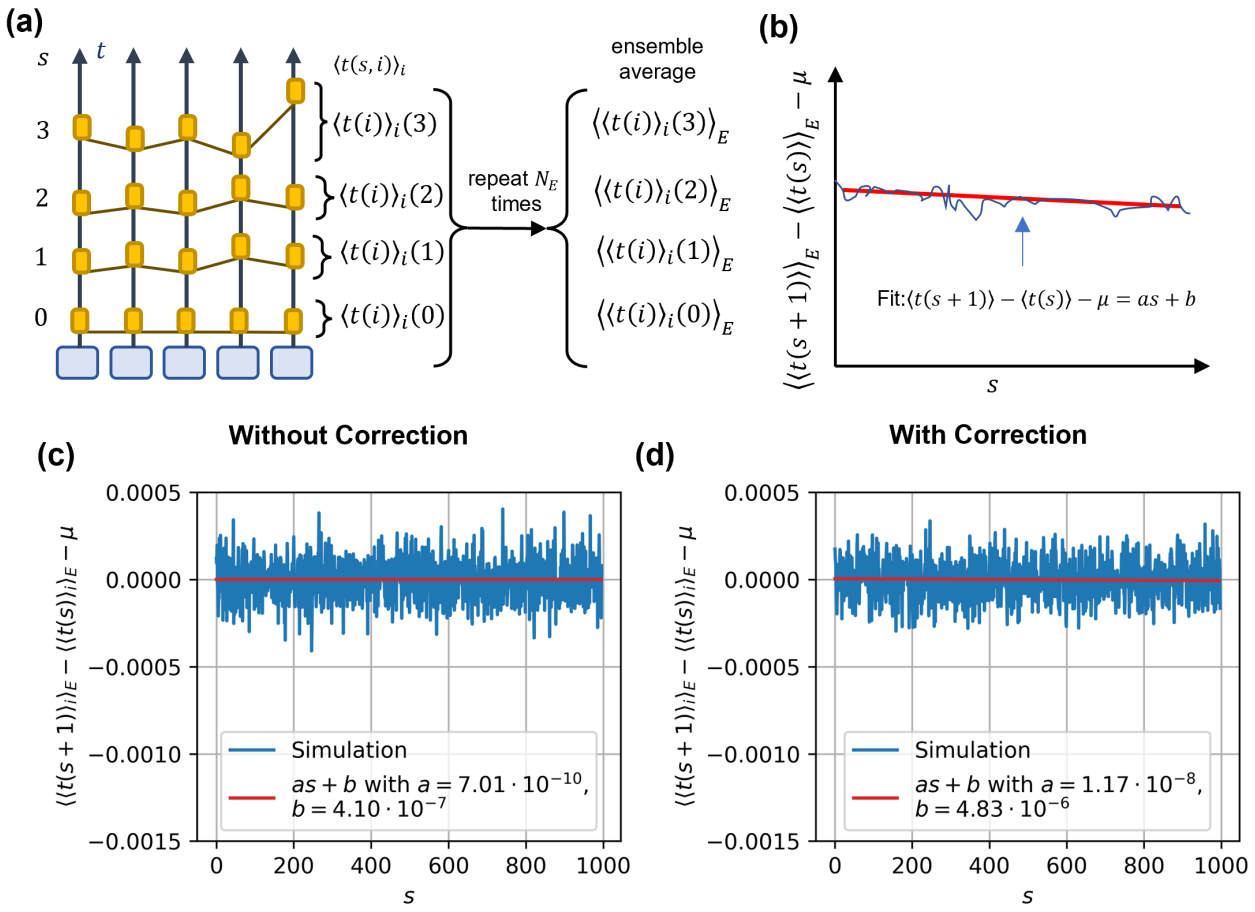


Figure 6.8 Correction Scheme - Velocity Fit. (a) Illustration of the calculation of the ensemble average of the average time of a generation. For an individual simulation, the update times of each cell of each generation s are recorded. The average time of a generation is calculated by spatially averaging $\langle t\,(s,i)\rangle_i$. After performing the simulation N_E times, the corresponding ensemble averages $\langle\langle t\,(s,i)\rangle_i\rangle_E$ are calculated. (b) Illustration of the fit of effective changes of the movement velocity. To calculate the effective change of the movement velocity of the temporal surface, the difference between the average time of subsequent generations $\langle\langle t\,(s+1,i)\rangle_i\rangle_E - \langle\langle t\,(s,i)\rangle_i\rangle_E$ are calculated. Without correction, a velocity of $\mu = 1$, the average temporal update per generation, would be expected, and is thus subtracted from the difference to illustrate the deviation from the expectation. The resulting curve as a function of the generation is plotted and fitted with a linear function $ax + b$ to detect constant shifts as well as linear temporal changes. (c-d) Exemplary data for the cases $m = 0$ (without correction, c) and $m = 0.3$ (with correction, d) and otherwise default parameter choices with corresponding fit and fit parameters in the legend. The value of a close to 0 shows for both effectively no net deviation from a shift of the temporal surface of μ per generation.

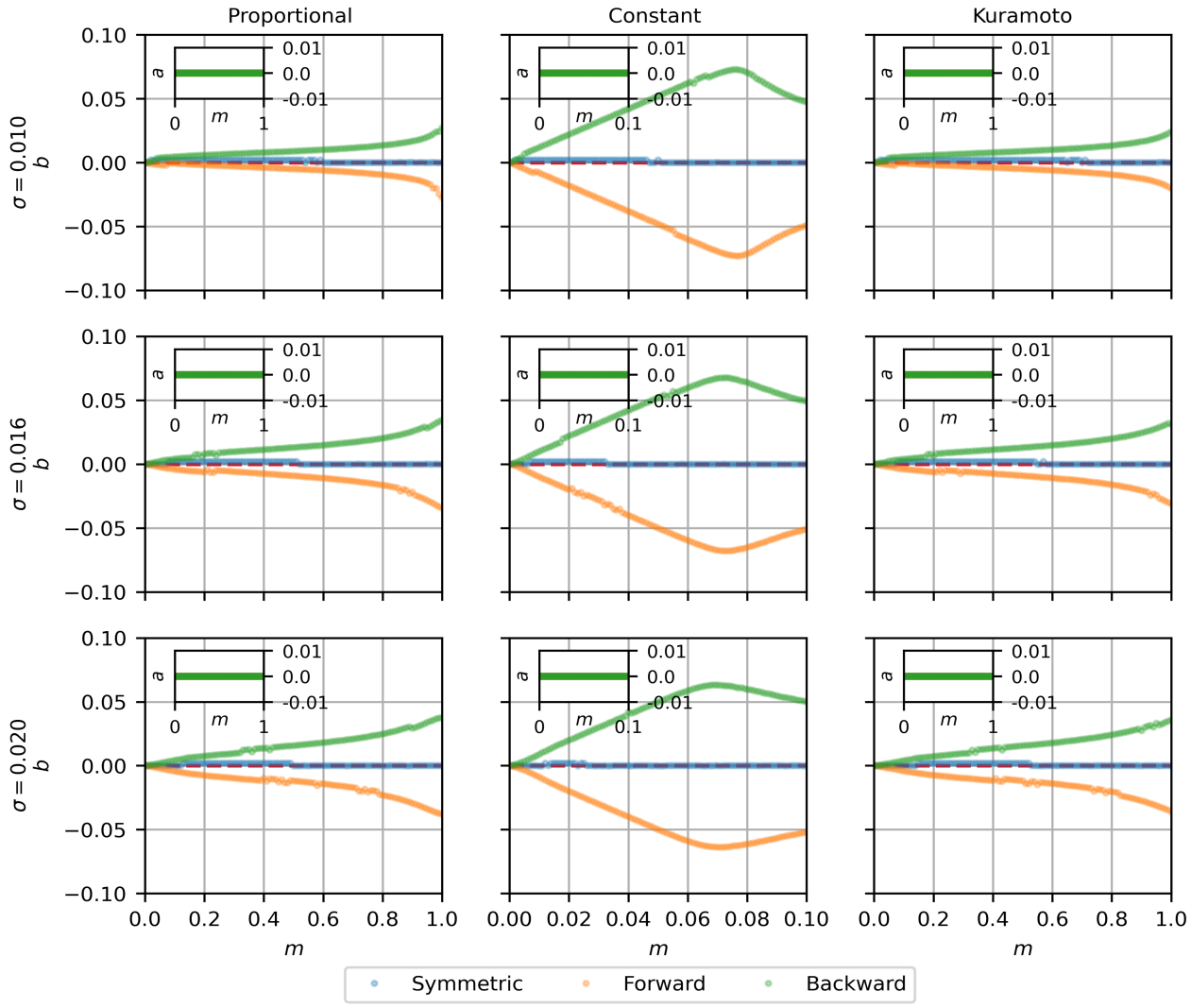


Figure 6.9 Correction Scheme - Velocity Results. Fit parameter b and a (insets) as illustrated in fig. 6.8 as a function of the proportionality constant m for proportional correction (first column), constant correction (second column) and Kuramoto correction (third column) as well as symmetric, forward and backward correction (blue, orange and green colors, respectively) and standard deviations of the update noise (different rows). Since a has values practically 0 in all cases (the curves overlap in each inset) and thus even over the course of a thousand generations, $a \cdot s$ is small compared to $\mu = 1$, there is no linear temporal increase or decrease of the velocity. While for symmetric correction, there is no constant shift of the temporal update velocity, for forward (backward) correction the velocity is decreased (increased). For proportional and Kuramoto correction the increase gets larger with increasing proportionality constant m, which can be explained with the fact that the introduced bias in one direction scales with the proportionality constant. For the constant correction case the unimodal curve can be explained by the fact that for large corrections m, the applied correction always has the same value but might actually introduce destabilizing effects, since it may “over-correct”, and once the temporal difference is larger than the detection window, no correction is applied at all.

Detection window

As last parameter of the correction schemes the detection window size will be explored in this section. In fig. 6.10 (a), the expected time to first asynchrony $\langle t_{First\ Asynchrony} \rangle_E$ is plotted as a function of the detection window size t_{detect} together with an indication of the size of the delay time t_{delay} for otherwise default values. For increasing t_{detect}, $\langle t_{First\ Asynchrony} \rangle_E$ is small until the detection window t_{detect} is only a little bit smaller than the delay t_{delay}, then increases sharply until it reaches an plateau value at t_{delay}. $\langle t_{First\ Asynchrony} \rangle_E$ drops again when $\mu - t_{delay} = 1 - t_{delay}$ is reached, and stays small until $t_{detect} = 1$, which marks the average onset time of the next generation.

For the first region of this curve, a sharp increase in $\langle t_{First\ Asynchrony} \rangle_E$, the hypothesis can be formed that for a too small detection window the noise is too strong and pushes the system out of detection with relatively large probability in a single step even if the temporal difference Δt between the two cells in question was 0 before.[3] This hypothesis is confirmed in fig. 6.10 (b-d), where the same plot is shown for different delay times t_{delay} and standard deviations of the update noise sigma σ, and the steep increases of $\langle t_{First\ Asynchrony} \rangle_E$ happen at lower values of t_{detect} for lower noise levels.

The second region of the curve is the plateau value which is abruptly reached at $t_{detect} = t_{delay}$, and can be easily explained by the fact that if $t_{detect} \geq \Delta t > t_{delay}$, despite the correction scheme still functioning the system is already asynchronous by definition. Larger delay times allow similarly as in systems without correction scheme the tolerance of larger noises (fig. 6.10 (b-d)).

Finally, the sharp decrease once the system reaches $t_{detect} = 1 - t_{delay}$ can be attributed to the possible interference of other cells which are in different generations. The correction of update times of these cells may not lead to asynchrony when compared to cells within the same generation. But because their updates are detected by neighboring cells updating at a different generation, the proportional correction scheme induces a large correction of a magnitude $|m\Delta t| > 0.3 \cdot 0.9 = 0.27 > 0.1 = t_{detect}$. The decrease is again also seen in a similar fashion for other delay times and noise levels (fig. 6.10 (b-d)).

6.3.3 Cellular Level

In the previous section, the properties of the correction algorithm have been varied and discussed. In this section, the focus is shifted to parameters on the cellular level independent of correction algorithm - the noise of the temporal update time and the delay of information transmission.

Noise

A first exploration of the dependency of $\langle t_{First\ Asynchrony} \rangle_E$ on the standard deviation of the update noise σ has already been performed in the context of discussing its interplay with the correction strength m (fig. 6.6 (c)) which established the general sigmoidal like form of the resulting curve, and that the optimal correction strength is independent of the noise.

Another question with regards to the noise in the system is to ask how well the correction scheme performs compared to not having a correction scheme, and compared to having a perfect correction scheme. While the former comparison can easily be obtained by simulating the system with $m = 0$, the latter refers to a system where a miraculous correction schemes is able to perfectly flatten the temporal landscape, i.e., achieve $\Delta t_{i,j} = 0$ for all neighboring cells i, j. The only way the asynchrony is now induced

[3] As a rough estimate: for a single update, the standard deviation of the Gaussian random variable is $\sigma = 0.016$ for standard parameters. Since the essential variable is the temporal difference between two neighboring cells, this difference has a standard deviation of $\sigma_{\Delta t} = \sqrt{2}\sigma = 0.0226$. If the detection window size is set at $3\sigma_{\Delta t} = 0.0678$, after 20 steps there is already a chance of approximately 64% that the system has lost synchrony in a single step.

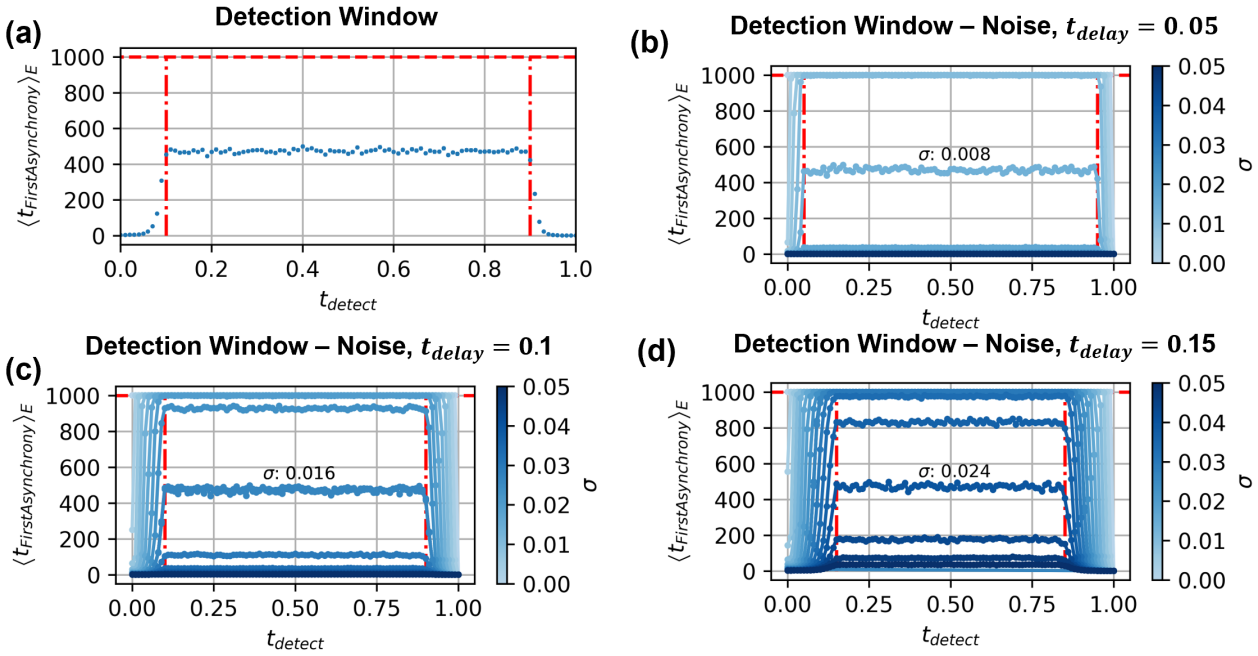


Figure 6.10 Correction Scheme - Detection Window. (a) $\langle t_{First\ Asynchrony} \rangle_E$ as a function of the detection window t_{detect} with $t_{delay} = 0.1$ and $\mu - t_{delay} = 0.9$ shown as dash-dotted lines. Between these lines, $\langle t_{First\ Asynchrony} \rangle_E$ stays effectively at a constant value, increasing towards it for low values and decreasing for large values, indicating that the lower of the two parameters, t_{delay} and t_{detect}, is dictating $\langle t_{First\ Asynchrony} \rangle_E$. (b-d) $\langle t_{First\ Asynchrony} \rangle_E$ as a function of the detection window t_{detect} for different values of the standard deviation σ of the update noise and for different delay times $t_{delay} = 0.05$ (b), $t_{delay} = 0.1$ (c), $t_{delay} = 0.15$ (d) shown as dash - dotted line, confirming the hypothesis that $\min(t_{detect}, t_{delay})$ is the relevant parameter while $\max(t_{detect}, t_{delay})$ is not relevant within the shown parameter limits. Additionally, the larger $\min(t_{detect}, t_{delay})$, the higher the tolerated noise.

is if a single step in the noise process directly leads to asynchrony before correction can be applied. $\langle t_{First\ Asynchrony} \rangle_E$ can then be estimated to be

$$\langle t_{First\ Asynchrony} \rangle_E \approx \sum_{s=1}^{S_{max}} s \left(\left(\mathrm{erf}\left(\frac{t_{delay}}{2\sigma} \right) \right)^L \right)^{s-1} \left(1 - \left(\mathrm{erf}\left(\frac{t_{delay}}{2\sigma} \right) \right)^L \right) + S_{max} \left[1 - \sum_{s=1}^{S_{max}} \left(\left(\mathrm{erf}\left(\frac{t_{delay}}{2\sigma} \right) \right)^L \right)^{s-1} \left(1 - \left(\mathrm{erf}\left(\frac{t_{delay}}{2\sigma} \right) \right)^L \right) \right] \quad (6.6)$$

with the first term being an approximation for the expectation value of $\langle t_{First\ Asynchrony} \rangle_E$ if the system starts to be asynchronous before the maximum number of generations $S_{max} = 1000$ is reached and the second term being an estimate how the latter is represented in a finite time simulation. The three resulting curves can be seen in fig. 6.11 (a), showing that the correction scheme achieves a performance close the perfect correction estimation.

Above estimation for perfect correction (eq. (6.6)) revealed a direct dependency on the size of the system through the number of total connections that may get asynchronous. In fig. 6.11 (b), the effect of the interplay of different system sizes L and the standard deviation of the update noise σ on $\langle t_{First\ Asynchrony} \rangle_E$ is plotted. Even for small system sizes, a large enough noise will eventually induce an asynchronous state, but the sensitivity on the length reduces with increasing system size.

Up until now, only one possible type of update noise was considered: annealed noise, modeled by a Gaussian random variable with mean μ and standard deviation σ newly drawn for each temporal update and cell. The goal of this type of noise is to mimic statistical fluctuations which are independent from each other in space and time. However, different cells may have systematic differences which are constant in time - to model this kind of systematic noise, a form of quenched noise is introduced by replacing the constant mean update time μ_i of each cell by a Gaussian random variable with mean μ and standard deviation σ_q which is constant in time but varies in space. In fig. 6.11 (c, d) $\langle t_{First\ Asynchrony} \rangle_E$ is shown as a function of the standard deviation of the annealed noise σ (standard deviation of the quenched noise σ_q) for different σ_q (σ). While the general shape of $\langle t_{First\ Asynchrony} \rangle_E$ as a function of the noises is in all relevant cases basically a monotonically decreasing function, the different type of noises induce slight differences. On the one hand, for the annealed noise, the noise is temporally independent and a new random variable for this contribution is computed at each step. On the other hand, the quenched noise represents systematic differences between the cells so the “dice” for the quenched noise are only “rolled” once at the beginning, and then stays temporarily constant. Thus, assuming perfect correction and only quenched noise, if the random variables are not large enough to force the system immediately out of synchrony, asynchrony will not occur during the time of the simulation. However, for the annealed noise the redrawing of the random variable at each time step increases the chances of pushing the system out of synchrony before the maximum simulation time is reached. This explains why for very low values of the other noise type, for quenched noise larger noise values are tolerated before $\langle t_{First\ Asynchrony} \rangle_E \approx 0$ is reached compared to similar values of the annealed noise.

Delay

To get a first impression of the effect of the delay t_{delay} on the conservation of synchrony measured by $\langle t_{First\ Asynchrony} \rangle_E$, in fig. 6.12 (a) it is shown as a function of t_{delay} for otherwise default parameters. Starting at very low values, $\langle t_{First\ Asynchrony} \rangle_E$ increases sharply around $t_{delay} \approx 5\sigma$, reaching a plateau at the maximum simulation time, and then falling again at large t_{delay}. At $t_{delay} = t_{detect} = 0.15$ no noticeable change in the curve is observed.

The initial upraise at the empirically determined $t_{delay} \approx 5\sigma$ can be explained by the fact that, assuming perfect correction, 5σ is approximately equivalent to producing a temporal difference larger than t_{delay}

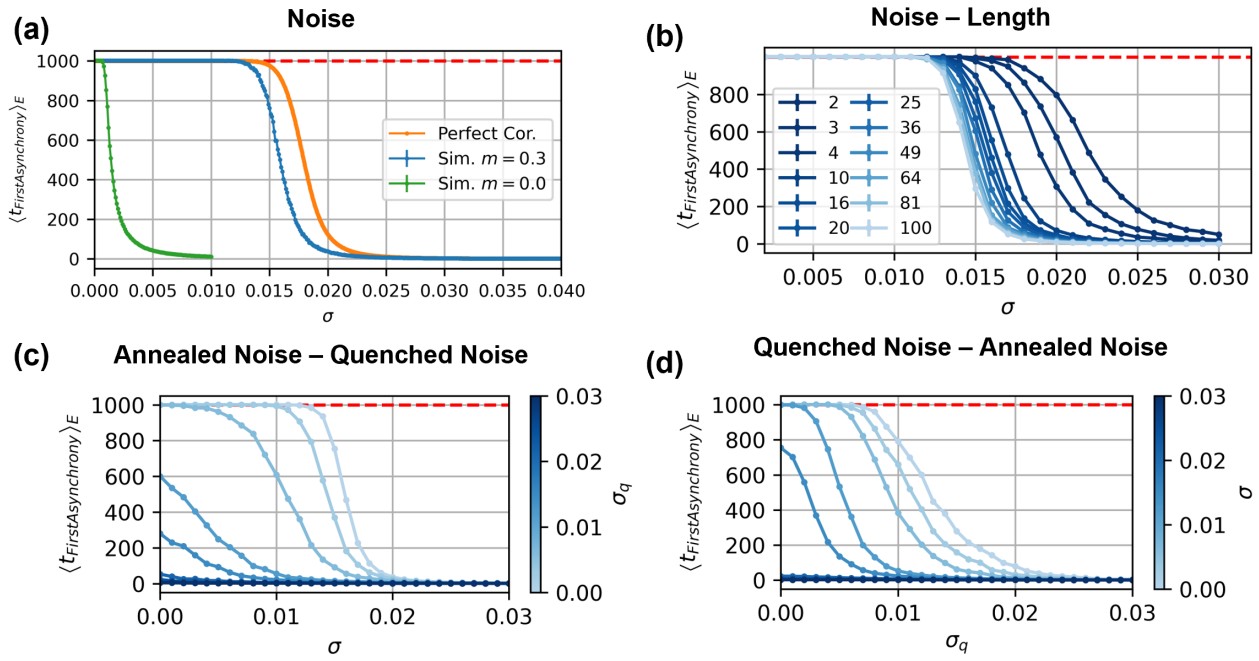


Figure 6.11 Correction Scheme - Noise. (a) $\langle t_{First\ Asynchrony} \rangle_E$ as a function of the standard deviation of the update noise σ for default parameters ($m = 0.3$), no correction ($m = 0$), and perfect correction, the last assuming a flat temporal landscape after each step, showing that for default parameter choices the proportional correction scheme is close to achieving a perfect correction. (b) $\langle t_{First\ Asynchrony} \rangle_E$ as a function of the standard deviation of the update noise σ for different system sizes, showing less dependency with increasing system size for all σ. (c) $\langle t_{First\ Asynchrony} \rangle_E$ as a function of the standard deviation of the (annealed) update noise σ for different choices of the standard deviation of the quenched noise σ_q changing the mean update time in a spatially varying, temporally constant way. (d) $\langle t_{First\ Asynchrony} \rangle_E$ as a function of the standard deviation of the quenched noise σ_q for different choices of the standard deviation of the update noise σ. For large fixed quenched noises the curves as functions of the annealed noise are lower than for exchanged roles of the noises.

with a chance of $\frac{1}{2500}$,[4] so roughly the region where intermediate values of $\langle t_{First\ Asynchrony} \rangle_E$ would be expected since in total $L \cdot s_{max} = 20 \cdot 1000 = 20000$ possibilities for such an event exist. This tendency of $\langle t_{First\ Asynchrony} \rangle_E$ to rise at $t_{delay} \approx 5\sigma$ is confirmed in fig. 6.12 (b-d), where the same plot is shown for different standard deviations of the noise σ and different detection windows t_{detect}.

The delay t_{delay} was already briefly discussed when looking at the detection window t_{detect}, with the result that t_{delay} acts as an upper limit for the effect the detection window has on synchronization conservation. For the default parameter values, t_{detect} is so large that $\langle t_{First\ Asynchrony} \rangle_E$ has already reached the maximum simulation time, thus no interplay between t_{detect} and t_{delay} can be observed. For larger standard deviations of the noise σ and smaller detection windows t_{detect}, it is apparent that lower values of t_{detect} lead to lower values of $\langle t_{First\ Asynchrony} \rangle_E$, even for larger values of t_{delay}. This can be explained by the fact that for small t_{detect} in comparison to σ, the temporal differences can by chance quickly grow larger than t_{detect}, from which on the synchronization schemes can no longer influence the temporal differences which in turn follow effectively a diffusive behavior until t_{delay} is reached and thus the system is asynchronous.

The decline of $\langle t_{First\ Asynchrony} \rangle_E$ for values of t_{delay} close to 1 is due to the fact that for large t_{delay} the system may formally still be in synchrony, but cells may for large enough detection windows detect cells from the "previous" generation. This induces large, unwanted corrections Δt and was partially the reason for introducing corrections windows in the first place.

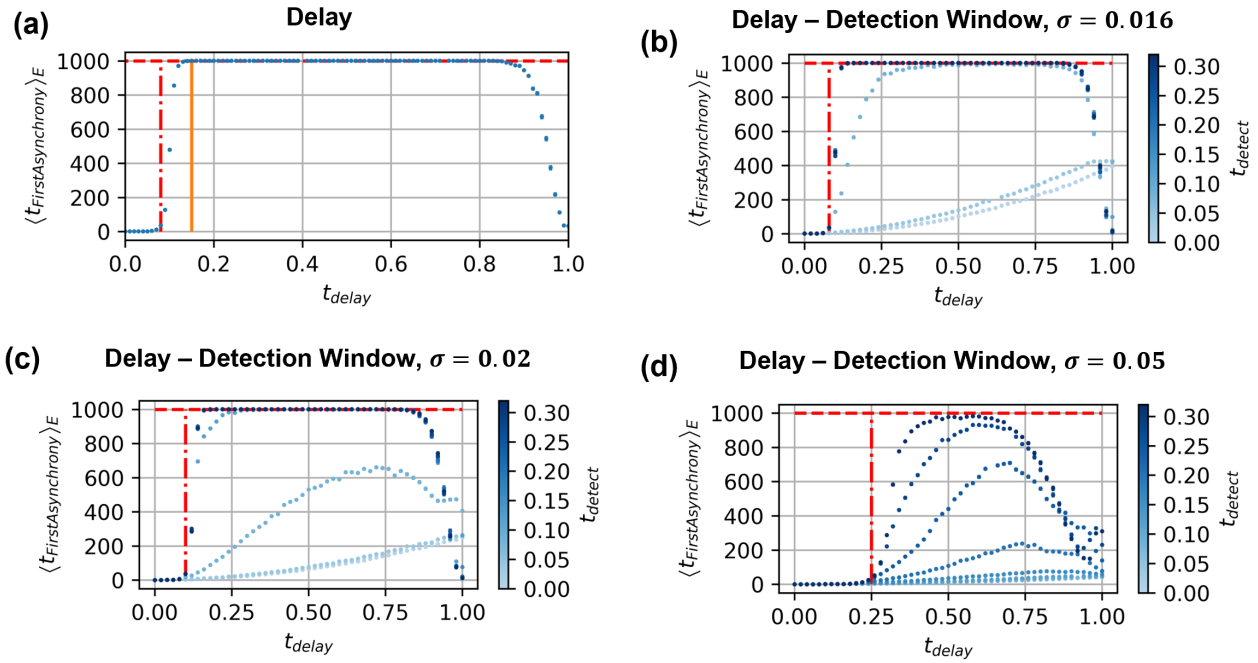


Figure 6.12 Correction Scheme - Delay. (a) $\langle t_{First\ Asynchrony} \rangle_E$ as a function of the delay t_{Delay} with the detection window size $t_{detect} = 0.15$ as solid orange line and 5σ marked by a dash-dotted line. (b-d) $\langle t_{First\ Asynchrony} \rangle_E$ as a function of the delay t_{Delay} for different standard deviations of the update noise $\sigma = 0.016$ (b), $\sigma = 0.02$ (c), $\sigma = 0.05$ (d), with 5σ marked by a dash-dotted line indicating that if the delay and detection window size are large enough compared to the noise the correction algorithm is able to smooth out the noise enough to allow for a large $\langle t_{First\ Asynchrony} \rangle_E$.

[4] This is evaluated by calculating the chance that a Gaussian random variable with mean 0 and standard deviation σ^* has absolute values $|\Delta t| \geq \frac{5}{\sqrt{2}}$,

$$P\left(|\Delta t| \geq \frac{5}{\sqrt{2}}\right) = 1 - \int_{-\frac{5}{\sqrt{2}}}^{\frac{5}{\sqrt{2}}} \frac{1}{\sqrt{2\pi\sigma^{*2}}} \exp\left[\frac{\Delta t^2}{2\sigma^{*2}}\right] d\Delta t = 1 - erf\left(\frac{5}{2}\right) \approx 0.0004 = \frac{1}{2500} \tag{6.7}$$

6.3.4 Tissue Level

All properties so far have been discussed in the context of a static 1D grid of (mostly) size $L = 20$ with periodic boundary conditions, and the focus was on the influence of the correction algorithm and the characteristics of individual cells. In this section, the focus is shifted to the properties of the tissue - its size and shape in the form of boundary conditions, if it is static or grows, in how many dimensions the tissue is situated and how cells are connected to each other.

Size, Boundary Conditions and Growth

A first look on the dependency of the expected time to first asynchrony $\langle t_{First\ Asynchrony} \rangle_E$ on the system size L was taken in fig. 6.11 (b), which indicated that for larger system sizes $\langle t_{First\ Asynchrony} \rangle_E$ decreased. In fig. 6.13 (a), this dependency is shown explicitly, with $\langle t_{First\ Asynchrony} \rangle_E$ at first rapidly decreasing with system size. Ignoring finite size effects most likely introduced by the fixed number of simulated generations, the data can be well approximated by $\langle t_{First\ Asynchrony} \rangle_E \sim \frac{1}{L}$.

The default setting with regards to boundary conditions is having a "periodic" setting, where the left and right end of the 1D line of cells are connected to each other, opposed to an "open" boundary condition setting where the cells are disconnected. *A priori* it is not clear if the additional connection between the two boundary cells is detrimental to synchronization because an additional connection exists at which the asynchrony could appear, or favorable for synchronization since two distant cells are coupled reducing the overall average distance between the cells leading to a potentially more temporally coherent system. To investigate the influence of the choice of boundary conditions, in fig. 6.13 (b, c), $\langle t_{First\ Asynchrony} \rangle_E$ is plotted as a function of the proportionality constant m for different boundary conditions and system sizes. For small system sizes, the periodic systems have a slightly larger optimal $\langle t_{First\ Asynchrony} \rangle_E$. The proportionality constant m_{opt} at which this optimum appears is in general lower for periodic boundary conditions than for open boundary conditions. Thus, it appears that the effect of the additional interaction favors synchronization and requires less correction input. Please note that in fig. 6.13 (c), for system size 2, $m_{opt} \approx 0.5$ for fixed and $m_{opt} \approx 0.25$ for periodic boundary conditions. These are the values one would naively assume for perfect correction since $m_{opt} \approx 0.5$ shrinks the difference for symmetric correction and only two cells with only one connection between them to 0, as well as $m_{opt} \approx 0.25$ does if there are two connection between the same two cells, which is the case for periodic boundary conditions. For large system sizes, the effects of the boundary condition diminish and the curves converge towards each other.

In the examples studied so far, the number of cells that try to conserve synchrony was kept constant. However, a standard feature of many pattern forming or synchronizing systems is a simultaneous growth, e.g., during somite formation in vertebrae [13]. To investigate if the conservation of synchronization can be realized in a growing system, a 1D system of initial size $L = 2$ with an open topology and otherwise default parameters is grown by adding an additional cell at the right end of the line of cells and initially inherits the same update time as its direct neighbor, but afterwards evolves freely like all other cells. This procedure is performed at every step in time until a predefined maximum amount of cells N_{added}^{max} have been added, from this point in time the number of cells is kept constant until the end of the simulation. In fig. 6.13 (d) $\langle t_{First\ Asynchrony} \rangle_E$ is shown as a function of m for different values of N_{added}^{max}, displaying a similar behavior then the curves in fig. 6.13 (b) without growth. This indicates that the synchronization conservation algorithm is able to tolerate growth. Furthermore, the problem of the artificial assumption that the system is initially in perfect synchrony is solved by this result as it is nearly equivalent to starting with a single cell which, through cell division, grows the tissue to the desired final size.

2 and 3 Dimensions

While different boundary conditions and system sizes have been investigated in the previous section, all the systems explored so far have been 1D. In this section regular 2D and 3D as well as irregular 2D system will be highlighted. The number of cells in each dimension will be denoted by L_1, L_2 and L_3, respectively. To be able to compare such different geometries, in all cases 36 cells are used - in the 1D

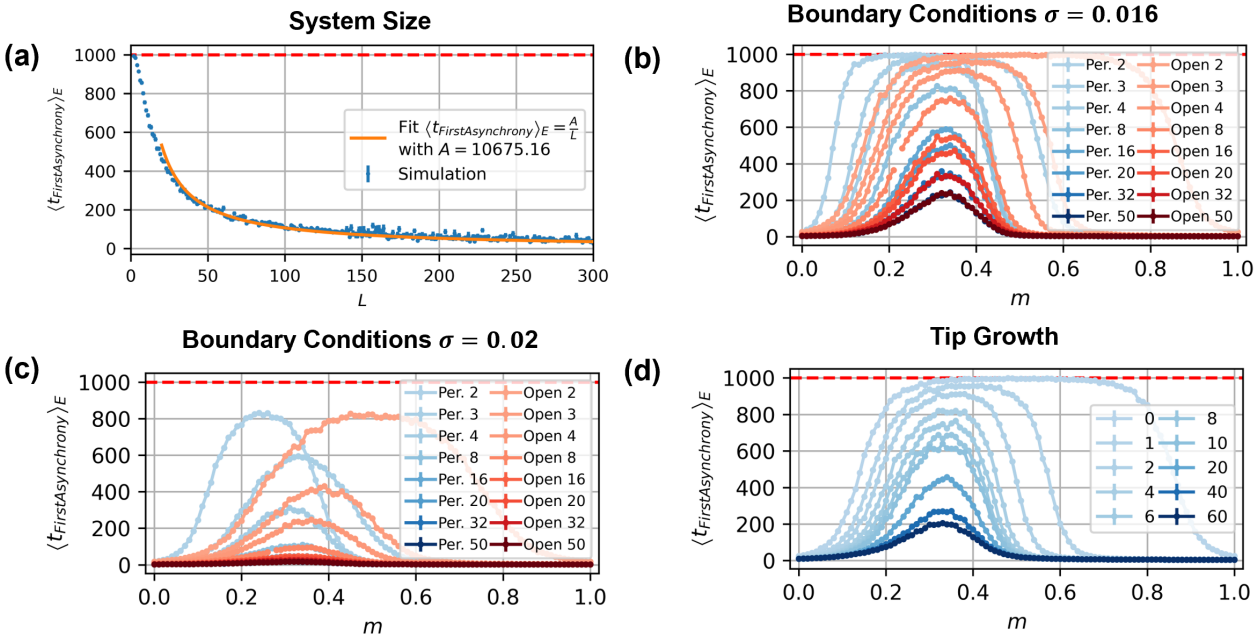


Figure 6.13 Correction Scheme - Size, Boundary and Growth. (a) $\langle t_{First\ Asynchrony}\rangle_E$ as a function of system size L shows a slowing, monotonic decrease (ensemble sizes $N_E = 10000$ for $2 \leq L \leq 4$, $N_E = 1000$ for $5 \leq L \leq 29$, $N_E = 500$ for $30 \leq L \leq 59$, $N_E = 200$ for $60 \leq L \leq 139$, $N_E = 20$ for $140 \leq L \leq 299$). (b-c) $\langle t_{First\ Asynchrony}\rangle_E$ as a function of the proportionality constant m for periodic (blues) and open (reds) boundary condition for different system sizes L (color gradients) and different standard deviations $\sigma = 0.016$ (b) and $\sigma = 0.02$ (c) of the update noise show that for small system sizes the periodic boundary conditions allow for a slightly better correction due to the one more available connection and requires a slightly lower proportionality constant for optimal correction, while for larger system sizes the difference diminishes. (d) $\langle t_{First\ Asynchrony}\rangle_E$ as a function of the proportionality constant m for a system with initial size $L = 2$ and open boundary conditions growing at one of its ends with a rate of one cell per time step up to a maximum number of added cells indicated by the labels in the legend.

case $L_1 = 36$, in the 2D cases $L_1 = 6$, $L_2 = 6$, and in the 3D cases $L_1 = 4$, $L_1 = 3$, $L_3 = 3$. For the regular 2D and 3D cases both square grids with von Neumann and Moore neighborhood are investigated, for 2D case additionally a hexagonal grid is taken into account. $\langle t_{First\ Asynchrony} \rangle_E$ as a function of the proportionality constant m is shown in fig. 6.14 (a). In principle, the same motives are at play as for the different boundary conditions - more connections may increase the number of possibilites cells can correct their relative position, but in turn also increase the number of connections where an asynchrony might appear. While for periodic boundary conditions the first effect seemed dominant, for additional dimensions there is no clear tendency with 2D system performing slightly better in most cases for optimal choice of m, while the 3D case with Moore neighborhood performs considerably worse. The curves seem to shift to small m values for an increased number of connections, with the 2D hexagonal and the 3D von Neumann neighborhood case with both 6 neighboring cells showing a nearly perfect overlapp. Rescaling m with the number of cells in the corresponding neighborhood, $m_{rescaled} = N_{Neighbors} m$, e.g., $m_{rescaled} = 3m$ for the 1D case, fig. 6.14 (b) is obtained which shows good overlapp of the curves with an optimum at $m_{rescaled} = 1$ indicating a general dependency on the relative amount of inputs the cell receives.

To check whether conservation of synchronization also works for irregular grids, these grids are produced by shifting midpoints of cells of an originally hexagonal grid randomly according to a triangular distribution with standard deviation r followed by a Voronoi tessellation as presented in appendix A.7. For the same irregularity r, 10 different grids are produced and each grid is sampled $N_E = 500$ times. In fig. 6.14 (c-f), $\langle t_{First\ Asynchrony} \rangle_E$ as a function of the irregularity of the grid r for different proportionality constant can be seen. Three trends can be identified: First, if m is similar to the optimal value for 2D hexagonal grids (fig. 6.14 (a, e)), larger $\langle t_{First\ Asynchrony} \rangle_E$ values can be observed. Second, in general the lower the irregularity, the larger $\langle t_{First\ Asynchrony} \rangle_E$, although the effect seems to be small. Third, there may be large differences between individual irregular grids as indicated by the small markers representing $\langle t_{First\ Asynchrony} \rangle_E$ for the 10 different grids for each r and m.

External Driven Cell

What if a cell does not follow the usual, noisy temporal update procedure but is instead driven externally to follow a precisely determined frequency and still couples to the other cells in the sense that it may influence its neighbors through its correction scheme, itself however does not react to temporal differences to the neighbor? In section 6.3.4, $\langle t_{First\ Asynchrony} \rangle_E$ as a function of the update step of such a single, externally controlled cell for otherwise default parameters is shown. While for $\mu_{Ext} = \mu = 1$, $\langle t_{First\ Asynchrony} \rangle_E$ seems to be slightly larger compared to a system without externally driven cell, a rapid decrease of $\langle t_{First\ Asynchrony} \rangle_E$ is obtained when μ_{Ext} deviates from 1. A possible explanation for this rapid decrease can be that the correction algorithm is optimized to correct the update time of both cells - but the externally driven cell does not change, putting the algorithm in a spot were it constantly under-corrects against a systematic increases of the temporal differences.

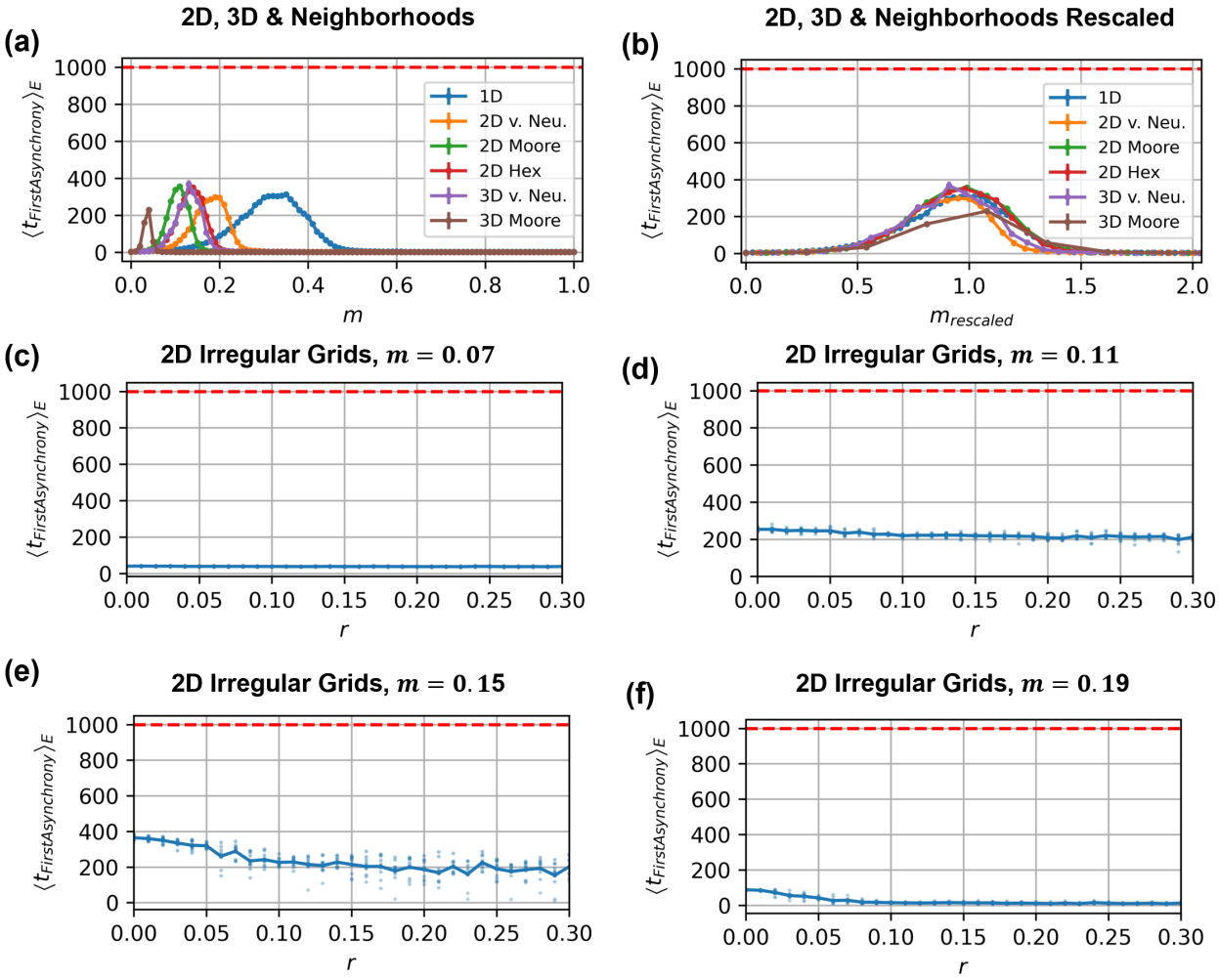


Figure 6.14 Correction Scheme - 2D and 3D. (a) $\langle t_{First\ Asynchrony} \rangle_E$ as a function of the proportionality constant m for different system dimensions, topologies and neighborhoods: 1D with periodic boundary conditions, 2D square grids with v. Neumann and Moore neighborhoods as well as hexagonal grids, 3D square grids with v. Neumann and Moore neighborhoods. All systems have 36 cells but a potentially different number of neighbors per cell. While the optimal proportionality constant shrinks for more neighbors per cell, the maximal expected time to first asynchrony $\langle t_{First\ Asynchrony} \rangle_E$ follows a nontrivial behaviour. The latter can be explained by the fact that more neighbors mean more opportunities for correction, but at the same time also more opportunities for asynchronies. (b) Same as in (a), but with $m_{rescaled} = N_{Neighbors} m$, shows that rescaling the proportionality constant by the number of neighbors leads to a good overlap of the curves. (c-f) $\langle t_{First\ Asynchrony} \rangle_E$ as a function of the irregularity of a 2D grid with 36 cells as illustrated in appendix A.7 for different values of the proportionality constant $m = 0.07$ (c), $m = 0.11$ (d), $m = 0.15$ (e), $m = 0.19$ (f). Solid lines indicates average over 10 different topologies with $N_E = 500$ for each topology, with individual points indicate values for the single topologies. For values of the proportionality constant around the optimum for hexagonal grid determined from panel (a) of this figure of $m = 0.15$, $\langle t_{First\ Asynchrony} \rangle_E$ is larger for low irregularities than for large irregularities, while away from this optimum this trend is weaker, yet even for larger irregularity of the grid an effective correction can be obtained.

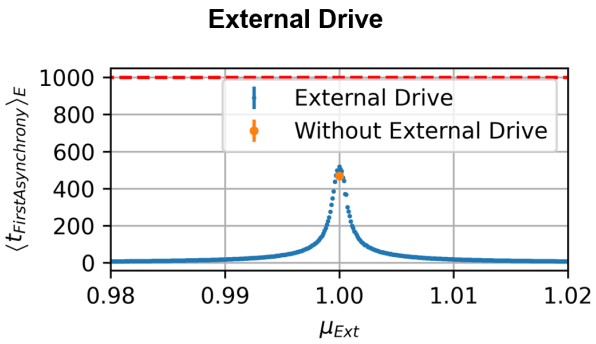


Figure 6.15 Correction Scheme - External Drive. $\langle t_{First\ Asynchrony} \rangle_E$ as a function of the temporal update step of a single, "externally driven" cell, i.e., a cell which updates its time with each generation by μ_{Ext} without the influence of noise. Only in a narrow region around $\mu = \mu_{Ext}$ correction schemes works appropriately.

6.4 Pattern Formation and Conservation of Synchrony in Cellular Automata

6.4.1 Coupling of Pattern Formation to Synchronization

The possibly simplest connection of the synchronization schemes discussed in section 6.3 to the CA with delay and noisy temporal update described in section 6.2 is a direct combination of the two models - a pattern forming CA is running as described in section 6.2 while a correction scheme as described in section 6.3 tries to conserve synchrony without any influence of the CA dynamics on the synchronization scheme but an asynchronous dynamics being (possibly) reflect in the CA pattern formation.

To compare the performance with and without synchronization scheme, in fig. 6.16 (a-h) $\langle t_{First\ Difference} \rangle_E$ for default parameter values is shown for the different Wolfram classes for the case with (left column, same as in fig. 6.2) and without correction (right column). As a first general observation, $\langle t_{First\ Difference} \rangle_E$ increases drastically for all rules with low values of $\langle t_{First\ Difference} \rangle_E$ in the case without correction schemes. For all class 1 rules, $\langle t_{First\ Difference} \rangle_E$ is close to the maximum simulation time, and for all other rules $\langle t_{First\ Difference} \rangle_E \gtrapprox 600$. This is especially interesting since in fig. 6.16 (i), the increase in $\langle t_{First\ Asynchrony} \rangle_E$ is shown, which is the same for all rules up to sampling noise, but in general significantly less than 600 - this indicates that even if the pattern formation process is already asynchronous, for all rules it takes on average at least a bit longer until this change reflects in the pattern formation. A plausible explanation for this behaviour could be that it is probable that the asynchrony first appears localized to two cells only since asynchrony is an unlikely event - if the state of the relevant cells is the same with or without synchronous update for those two cells due to the specific local pattern around those 2 cells at this time, the asynchrony will not reflect in $\langle t_{First\ Difference} \rangle_E$, but it will in $\langle t_{First\ Asynchrony} \rangle_E$. Additionally it is possible that the system will resynchronize randomly since it is likely that Δt is just a little larger than t_{delay} and the following temporal differences between the asynchronous cells are still smaller than t_{detect} leading with large probability to resynchronization. If there is no correction scheme, asynchrony is happening after approximately 5 steps on average and thus much more likely - which means it is also likely that a lot of cells are in asynchrony with each other, increasing the likelihood of an influence on pattern formation. The drastic effects of the synchronization scheme are again emphasized in fig. 6.16 (j) where the differences in $\langle t_{First\ Difference} \rangle_E$ for the case with and without correction scheme are depicted - illustrating that for many rules the difference is more in the realm of 800 generations, for some even close to the maximum simulation time.

To supplement the descriptive analysis of the previous paragraph it is interesting to look at the relationship of different rule classification quantities with $\langle t_{First\ Difference} \rangle_E$ and $\langle t_{First\ Asynchrony} \rangle_E$ (fig. 6.17).

Neither Langton's λ_0 nor the Kolmogorov complexity K_R show any tendency that would allow the identification of synchronization conserving rules. At least in the first case, this is not surprising since the general understanding is that Langton's λ is only a good discriminator between dynamical regimes for a large number of states and a large neighborhood [107]. Rules with zero μ are guaranteed to have maximum $\langle t_{First\ Difference} \rangle_E$, which is similar to the case without correction scheme and not surprising because if

the output of a rule does not change if any input is changed, the expected behaviour for rule 0 and 255, asynchrony is not an issue for pattern formation as discussed in section 6.2. The "changiness" C, defined as the relative number of times in the rule table in which the state of the cell is changed, shows that a low number of changes, consistent with the CA reaching invariant steady states quickly, are supportive of an patterning process unperturbed by possible asynchrony even in the presence of a correction scheme.

From the local pattern based classifications, both Kolmogorov complexities as well as single site entropies are able to deliver either sufficient or necessary criteria for synchronization conserving rules. Zero spatial Kolmogorov complexity, equivalent to a very simple average pattern, is a sufficient criterion. Yet, low temporal Kolmogorov complexity is a necessary criterion, i.e., if each cell changes quite unpredictably over time from its local point of view the chances are high that the pattern formation process is influenced by the asynchrony. Both low spatial and temporal single site entropy are sufficient for large $\langle t_{First\ Difference} \rangle_E$ with a correction scheme, implying that if the cells do not change their state very much both in spatial or temporal direction the asynchrony has no effect on the pattern formation dynamics. Compared to the case without correction, low spatial and temporal single site entropy are now a guarantee for a conservation of synchronization close to the maximum simulation time and not only an indicator of larger values of $\langle t_{First\ Difference} \rangle_E$, hinting that the correction scheme is able to keep the on average simple patterns long enough synchronous that an invariant steady state is reached.

The synergies do not show any particular tendencies as in the case without a correction scheme.

Classifications based on the global spatio-temporal dynamics again support some necessary conditions – a small average attractor size and small transient length are both unavoidable properties of a dynamics that is not influenced by the asynchrony. This observation is again consistent with the observations made when studying the CA dynamics without correction scheme.

To sum up, the introduction of a correction scheme does not only improve the conservation of synchrony in the sense of an increased $\langle t_{First\ Asynchrony} \rangle_E$, but even more the conservation of the formed pattern in the sense of an increased $\langle t_{First\ Difference} \rangle_E$ compared to a system without correction scheme. Still, largely the same properties that helped a CA without correction scheme to obtain a quasi synchronous pattern even in the presence of asynchrony remain relevant.

6.4.2 Coupling of Pattern Formation to Synchronization and Synchronization to Pattern Formation

As a second variant of introducing correction schemes, not only CA dynamics is coupled to the correction scheme, but also the correction scheme to the CA dynamics. In practice, information about both correction scheme and CA state may be transmitted over the same channel, e.g., the state is communicated via a concentration of a specific molecule, and if the CA updates and changes its state this concentration changes. The change in concentration in the signal of the neighbors is then used as a marker for their temporal update. This means that temporal updates can only be detected and used in the correction if the state of the neighbor changes, introducing the mentioned coupling between CA and correction scheme. Such a correction scheme with coupling between CA and correction scheme will be termed "state dependent", while a correction scheme without this coupling will be termed "state independent".

For class 1 rules, the same rules which with correction scheme showed a larger value of $\langle t_{First\ Difference} \rangle_E$ than without a correction scheme (fig. 6.16 (a, e)), again show a higher tendency when comparing correction schemes with and without state dependence (fig. 6.19 (a, e)), yet lower than in the previous case. For class 2, the effect of the state dependence is in many cases detrimental, as well as for all case 3 and 4 rules ((fig. 6.19 (b-d, f-h)). Nevertheless, a direct comparison with the case without a correction scheme (fig. 6.16 (c-d)) reveals slight improvements in some cases. For most rules, the time to first asynchrony is rather low when employing correction schemes with state dependence (fig. 6.19 (i)), yet for some class 2 rules it has nearly the same value as in the case without state dependence.

The behavior of class 1 rules could be expected since the lower boundary of having no correction at all was already very high (fig. 6.2) and class 1 rules reach their homogeneous steady states quickly and are thus unperturbed by asynchronous dynamics. However, the diverse behavior of class 2 rules needs

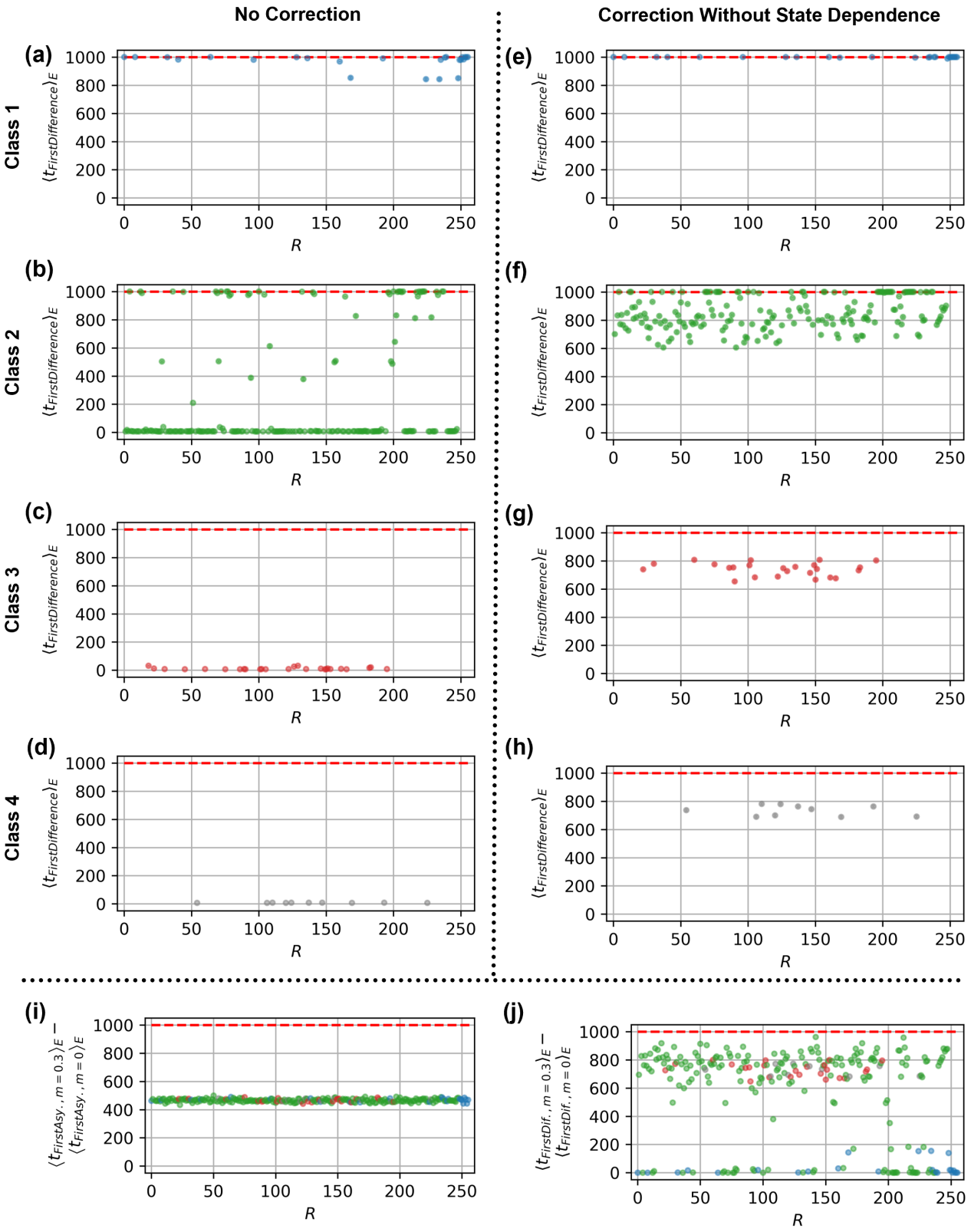


Figure 6.16 Synchronization without Correction or Correction without State Dependence (a-h) $\langle t_{First\ Difference} \rangle_E$ as a function of the rule for the 4 Wolfram classes for no correction algorithm (left column) and with a correction algorithm (right column). (i, j) Difference of $\langle t_{First\ Asynchrony} \rangle_E$ (i) and $\langle t_{First\ Difference} \rangle_E$ (j) with and without a correction algorithm as a function of the rule for all rules, as dots with colors as in (a-h).

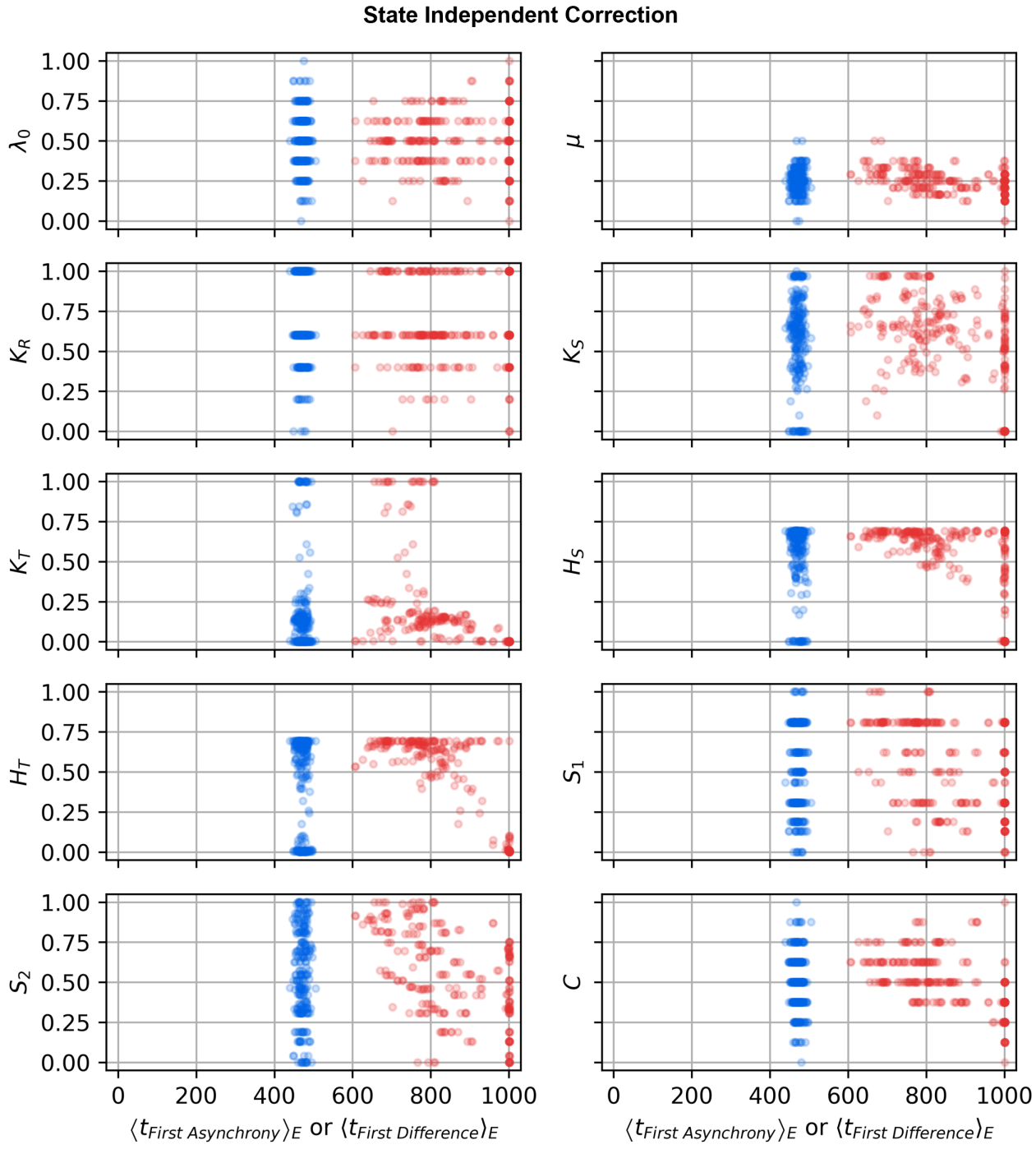


Figure 6.17 Synchronization with State Independent Correction - Classification Quantities. Classification quantities from appendix A.2.2 as a function of $\langle t_{First\ Difference} \rangle_E$ (red) and $\langle t_{First\ Asynchrony} \rangle_E$ (blue) for state independent correction.

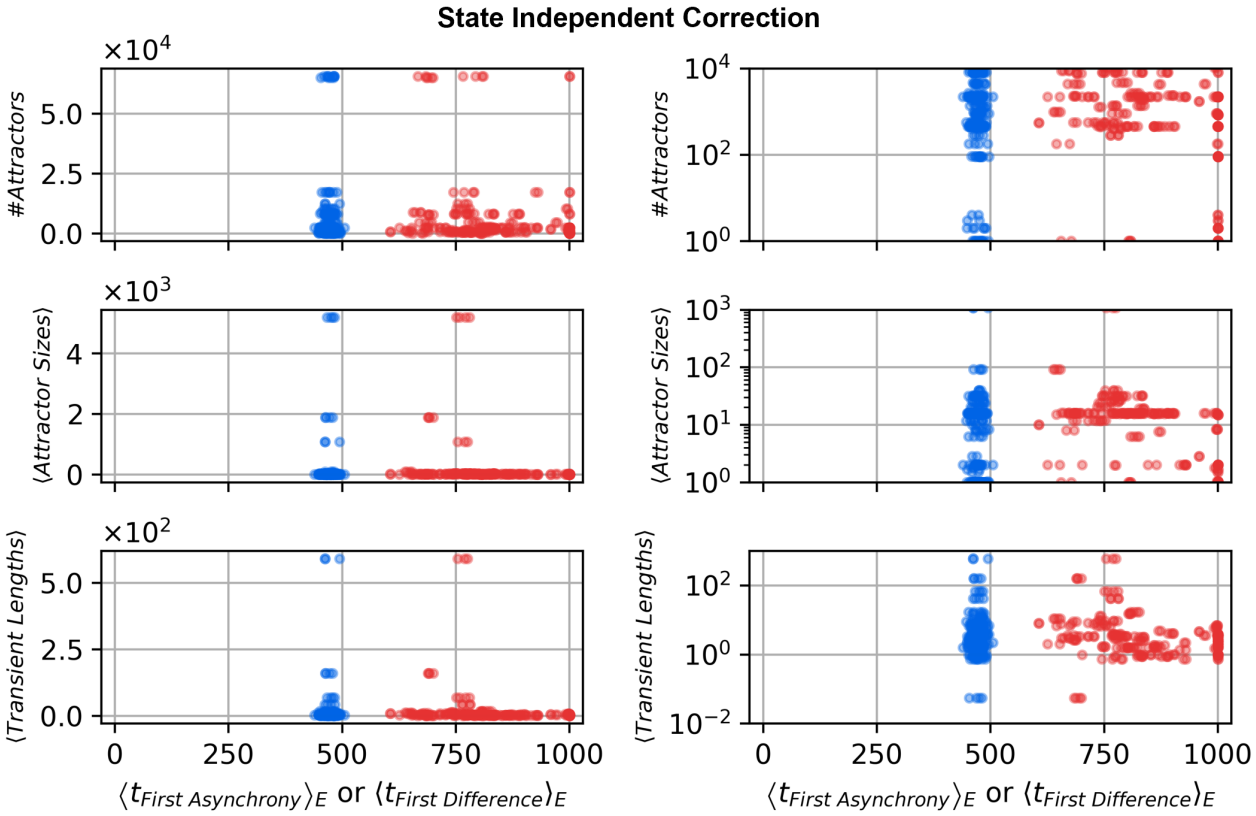


Figure 6.18 Synchronization with Correction - Attractor Properties. Number of attractors, average attractor sizes and average transient lengths as a function of $\langle t_{First\ Difference} \rangle_E$ (red) and $\langle t_{First\ Asynchrony} \rangle_E$ (blue) for state independent correction for the elementary cellular automata rules with linear (left column) and logarithmic (right column) scaling of the “y-axis” and state independent correction.

a bit more explanation. Class 2 consists of rules exhibiting either stable structures or periodic structures. Both periodic structures and stable structures can be favorable and detrimental, depending on the exact specificities. On the one hand, a highly repetitive pattern of quickly changing states of a periodic structure leads to a detection of updates of neighbors and thus to a larger $\langle t_{First\ Asynchrony} \rangle_E$, which is again a lower bound for $\langle t_{First\ Difference} \rangle_E$. On the other hand, quick succession of cell changes may lead to an immediately differing pattern once asynchrony is reached. Similarly, for cells in a localized stable structure on the one hand the correction scheme does not work anymore, leading to a quick asynchrony, which could manifest itself in the pattern if the stable structure is not yet permanent. On the other hand, if it is permanent and a global steady state is reached quickly it is invariant under asynchronous updates even if the steady state is not homogeneous. The chaotic and strange behavior of classes 3 and 4, respectively, lead with no large surprise to low values for $\langle t_{First\ Difference} \rangle_E$ since as few as on average approximately 5 steps without correction can be easily sufficient to reach asynchrony as shown in fig. 6.2, and this asynchrony can manifest itself quickly in the pattern if the cells in the asynchronous connection start changing their state again.

The classification quantities as a function of $\langle t_{First\ Difference} \rangle_E$ show qualitatively a similar behavior regarding larger trends as described for the cases without correction scheme or with state independent correction scheme (fig. 6.20). Through the coupling from CA to synchronization procedure, $\langle t_{First\ Asynchrony} \rangle_E$ for the first time depends on the rule in question and thus on its classification quantities. For $\langle t_{First\ Asynchrony} \rangle_E$ to be large, it is necessary that the rule has $\lambda_0 = 0.5$, $\mu \approx 0.25$, $K_R > 0.5$, large changiness C, small temporal Kolmogorov Complexity K_T, and large single point entropies H_S and H_T. All these findings are consistent with a rapidly changing pattern, which in turn would allow the correction scheme to correct for temporal differences and lead to large asynchrony times. Since $\langle t_{First\ Asynchrony} \rangle_E$ is a lower bound for $\langle t_{First\ Difference} \rangle_E$ by definition, for rapidly changing patterns $\langle t_{First\ Difference} \rangle_E$ can in general also be large.

Summed up, if the pattern formation of the CA and the synchronisation scheme are coupled in both directions, rules from class 3 and 4 and many rules from class 2 behave significantly worse. Two very different rule properties that can safeguard against changes in the pattern in the presence of update noise could be identified: Either the rule reaches quickly a dynamical fixed point making its dynamics invariant under asynchronous updates, or the rule produces a rapidly changing pattern allowing the correction scheme to work effectively and conserve the synchronous update.

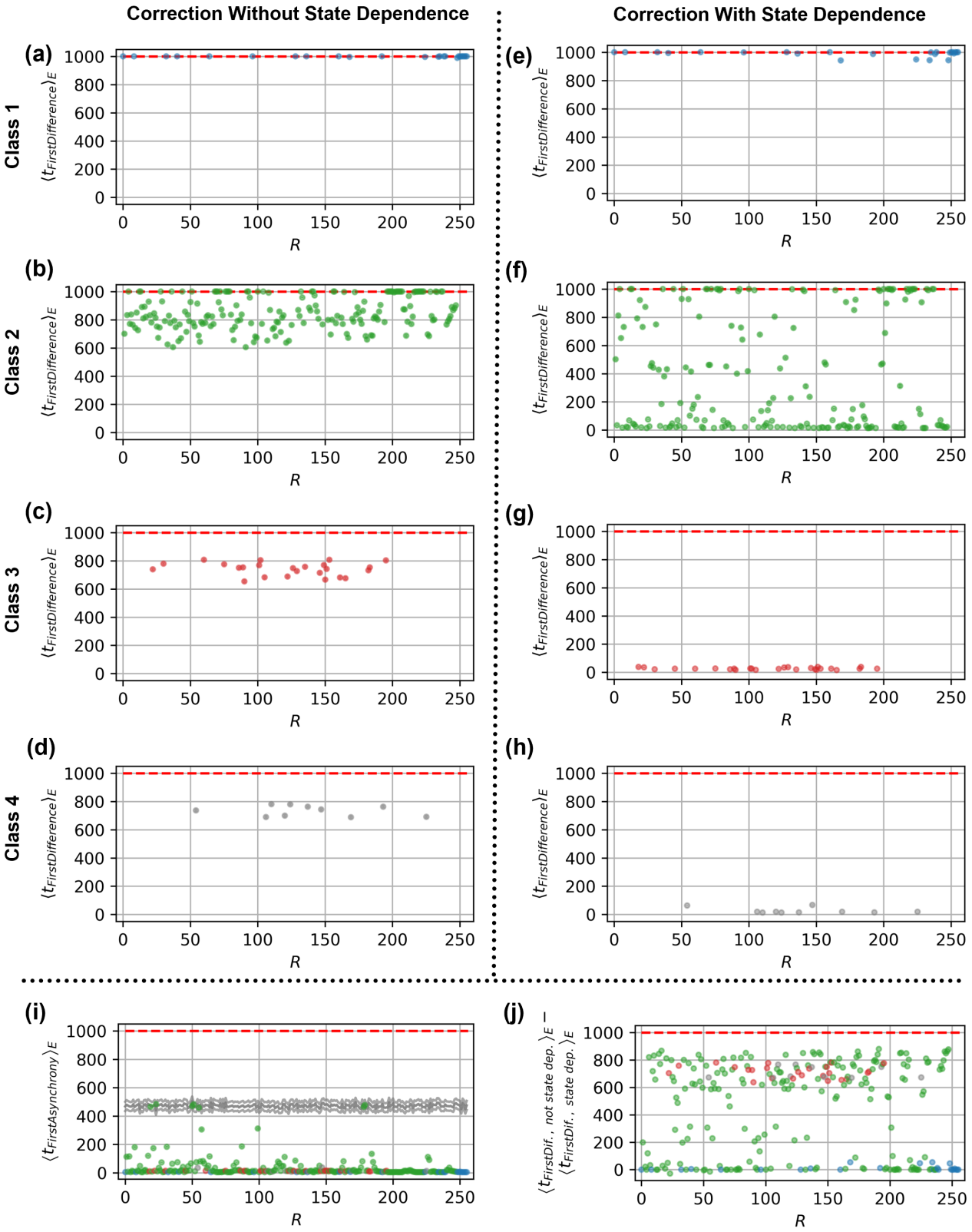


Figure 6.19 Synchronization with State Independent and Dependent Correction (a-h) $\langle t_{First\ Difference} \rangle_E$ as a function of the rule for the 4 Wolfram classes for a correction algorithm which does not depend on state changes (left column, same as fig. 6.16) or does depend on state changes (right column). (i) $\langle t_{First\ Asynchrony} \rangle_E$ as a function of the rule for all Rules, as dots with colors as in (a-h) with a correction algorithm dependency on state changes. As grey lines the values of $\langle t_{First\ Asynchrony} \rangle_E$ without dependency of the rule on state changes - the average in the middle, average plus/minus three standard as top/bottom grey line. (j) Difference between $\langle t_{First\ Difference} \rangle_E$ of a state independent and state dependent correction algorithm as a function of the rule.

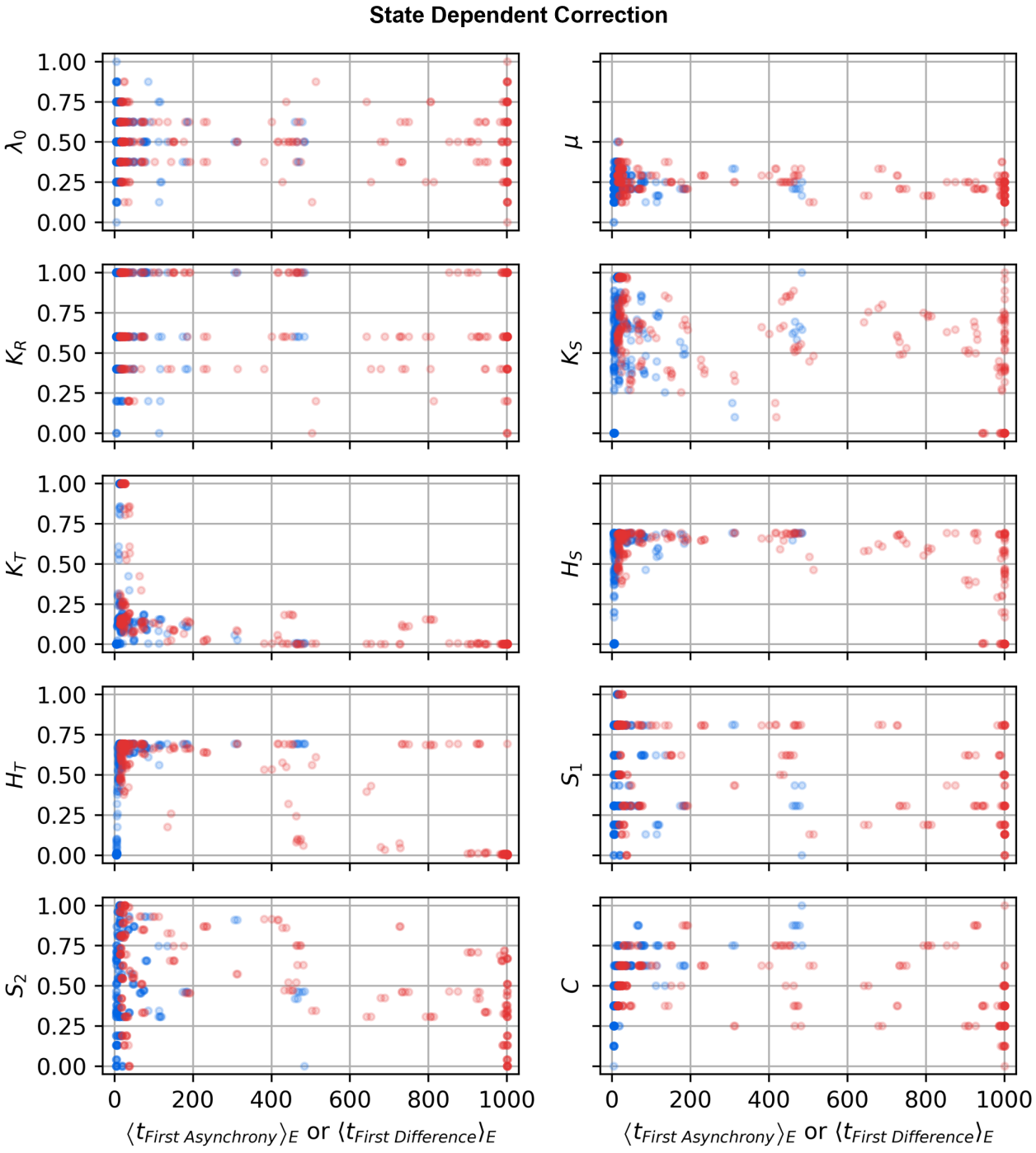


Figure 6.20 Synchronization With State Dependent Correction - Classification Quantities. Classification quantities from appendix A.2.2 as a function of $\langle t_{First\ Difference} \rangle_E$ (red) and $\langle t_{First\ Asynchrony} \rangle_E$ (blue) for state dependent correction.

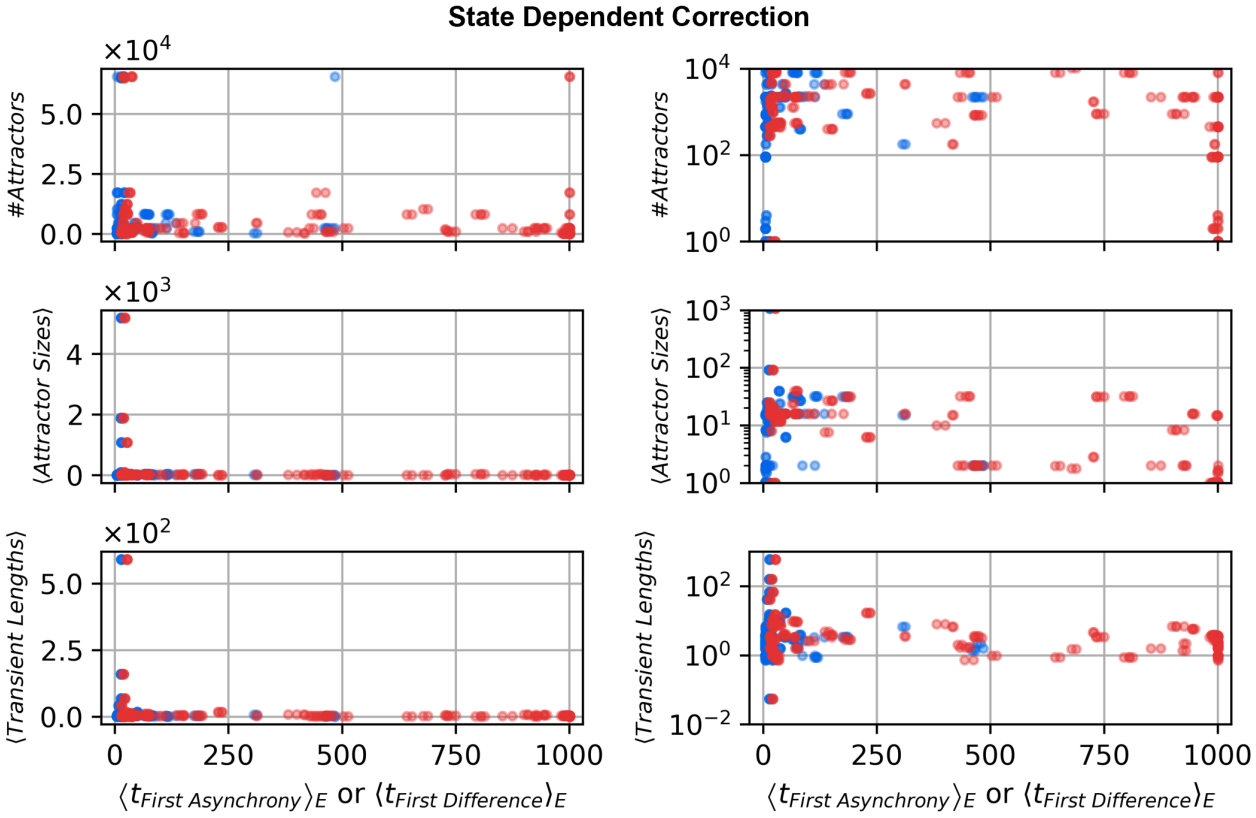


Figure 6.21 Synchronization with Correction - Attractor Properties. Number of attractors, average attractor sizes and average transient lengths as a function of $\langle t_{First\ Difference} \rangle_E$ (red) and $\langle t_{First\ Asynchrony} \rangle_E$ (blue) for state independent correction for the elementary cellular automata rules with linear (left column) and logarithmic (right column) scaling of the “y-axis” and state dependent correction.

6.5 Connection to the Physics of Surface Growth

6.5.1 Observables and Models of Surface Growth

To connect to the physics of surface growth [198], it is possible to interpret the time $t_s(\vec{x})$ of the sth update of each cell as a "height" above time $t = 0$. Interpolating between the individual cells, this is reminiscent of a surface. With each update step, this surface grows more and more, comparable to growth models in surface physics with the substitution of update step s and time $t_s(\vec{x})$ in the synchrony model with the time $\mathcal{T}$ and height $h(\vec{x}, \mathcal{T})$ in the surface growth model. The standard observable of interest in surface growth models is the standard deviation of the height function for a given s, called interface width, translated to the synchrony model in 1D with length L here

$$w(L, s) = \sqrt{\frac{1}{L}\sum_{i=1}^{L}\left[t_s(i) - \langle t_s(i)\rangle_i\right]^2}. \tag{6.8}$$

If the growth of one point of the surface is independent of the growth at another point of the surface, i.e., there is no correction in the framework of the synchrony model,

$$w(L, s) \sim s^{\frac{1}{2}} L^{\frac{1}{2}} \tag{6.9}$$

by the central limit theorem in the appropriate limits.

For interacting systems, i.e., systems with correction, experimentally and in toy models a similar scaling of the interface width of the form

$$w(L, s) \sim s^{\beta} F_w\left(L s^{\frac{1}{z}}\right) \sim \begin{cases} s^0 L^{\alpha} & \text{for } A t^{\frac{1}{z}} \gg L \\ s^{\beta} L^0 & \text{for } A t^{\frac{1}{z}} \ll L \end{cases} \tag{6.10}$$

with $z = \frac{\alpha}{\beta}$, scaling function $F_w(\cdot)$, and a constant A has been found [198]. The above scaling relations also imply invariance under the scaling

$$x \mapsto bx \qquad s \mapsto b^z t \qquad \delta t \mapsto b^{\alpha} \delta t \tag{6.11}$$

with $\delta t(x, s) = t(x, s) - \langle t(x, s)\rangle_x$ being the deviation of the time to the average [198]. For the scaling exponents α, β, and z, two characteristic sets of values exist describing two different universality classes: the Edwards-Wilkinson (EW) class and the Kardar-Parisi-Zhang (KPZ) class.

For the EW class, the scaling exponents take the values

$$(\alpha, \beta, z) = \left(\frac{1}{2}, \frac{1}{4}, 2\right). \tag{6.12}$$

A few toy models fall in this universality class, among them in discrete space random deposition with diffusion, and in continuous space the EW equation (in surface physics notation)

$$\frac{\partial}{\partial \mathcal{T}} h(x, \mathcal{T}) = v_0 + \nu \nabla^2 h + \eta(x, \mathcal{T}) \tag{6.13}$$

with $\eta(x, \mathcal{T})$ being Gaussian white noise and v_0 and ν being constants.

For the KPZ class, the scaling exponents take the values

$$(\alpha, \beta, z) = \left(\frac{1}{2}, \frac{1}{3}, \frac{3}{2}\right). \tag{6.14}$$

Amongst the toy models of this class are in discrete space ballistic deposition, and in continuous space the KPZ equation [199]

$$\frac{\partial}{\partial \mathcal{T}} h(x, \mathcal{T}) = v_0 + \nu \nabla^2 h + \frac{\lambda}{2} (\nabla h)^2 \eta(x, \mathcal{T}), \tag{6.15}$$

again with constants v_0, ν, λ.

6.5.2 Interface Width of the Temporal Surface under the Influence of Correction Schemes

To determine the interface width of the temporal surface, for each generation the standard deviation of the temporal surface is calculated and the ensemble average taken for each generation separately (fig. 6.22 (a)). The initial and final part of the curve are fit with a function $\langle w \rangle_E = A_1 s^{c_1}$ and $\langle w \rangle_E = A_2 s^{c_2}$, respectively (fig. 6.22 (b)). Without correction, both fits result in $c_i \approx 0.5$, consistent with the independent growth in the surface model (eq. (6.9), fig. 6.22 (c)). For the default parameter choice, the results of the fits are $c_1 = 0.176$ and $c_2 \approx 0$ (fig. 6.22 (d)). The result for c_2 is consistent with the results found in the KPZ and EW case for large times, while the result for c_1 does not fit either (eq. (6.12), eq. (6.14)). This is not necessarily surprising since according to the strong universality conjecture models usually need appropriate rescaling by "zooming" out in x and s dimension and scaling t appropriately to converge to the fixed points with EW or KPZ characteristics, respectively [200].

For the purpose of analyzing the exponents more systematically, in fig. 6.23 and fig. 6.24 the fit parameters are plotted as a function of m for different correction schemes and standard deviations of the noise σ. While there are quantitative differences, the general shape of the curve for c_2 is the same in all cases: For $m = 0$ and large m, the correction scheme is basically not in effect and $c_2 \approx 0.5$. For intermediate values around the optimum correction constant, $c_2 \approx 0$ and the correction scheme does work effectively preventing any further increase in the interface width. The data for c_1 (fig. 6.24) reveals a slightly more complicated picture. Proportional and Kuramoto correction schemes show a similar dependency: Starting from $c_1 = 0.5$ at $m = 0$, falling to slightly below 0.25 (forward and backward scheme) or around 0.15 (symmetric scheme) and subsequently rising again to larger values once m is significantly larger than the optimal proportionality constant. In its effective range, the correction scheme is thus not only able stabilize the interface width for large times but also to slow down the increase of the interface width in the initial phase. Again, the values expected for the theoretical limiting models are not met. For the constant correction scheme, the c_1 drops further down. Values below zero can be explained with the fit not including the first few values, giving room for a drop in interface width.[5]

[5] This specific procedure is used because for the simplistic applied fit procedures employing least square fitting with the python package scipy [201] it showed a better stability. With a more sophisticated approach currently developed by M. Bojer using various different functions to fit the curve for both small and large number of generations and deciding the most appropriate model via statistical testing procedure, better results might be possible.

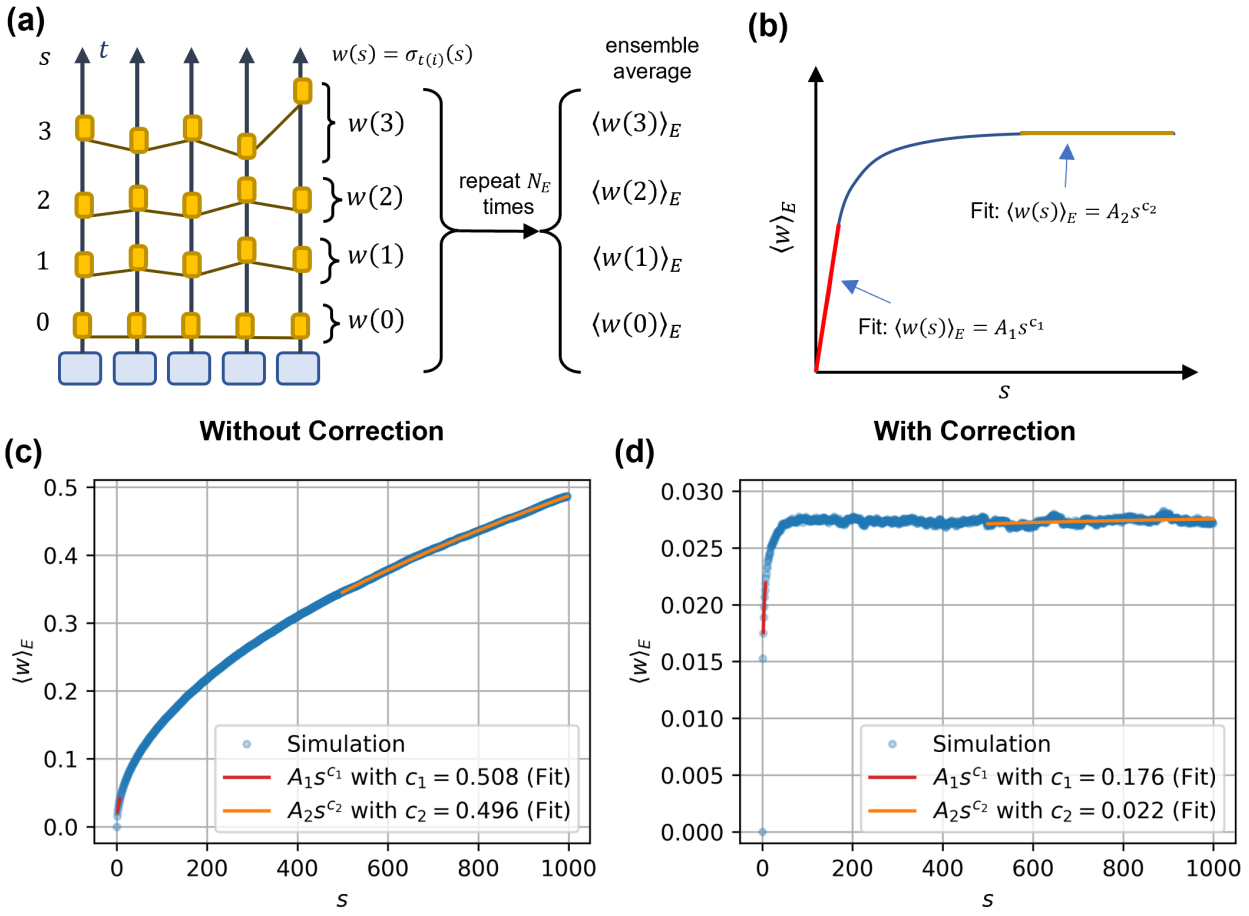


Figure 6.22 Temporal Surface Interface Width Fit (a) Illustration of the calculation of the ensemble average of the interface width of a generation. For an individual simulation, the update times of each cell of each generation s are recorded. The spatial standard deviation of the time of a generation is calculated $w(s) = \sigma_{t(i)}(s)$. After performing the simulation N_E times, the corresponding ensemble averages $\langle w(s)\rangle_E$ are calculated. (b) To compare to the standard surface growth classification quantities, $\langle w(s)\rangle_E$ is fitted by $A_1 s^{c_1}$ in the first few generations and with $A_2 s^{c_2}$ in the last half of all generations. (c-d) Examples of the plot in (b) with data from a simulation without correction ($m = 0$, c) and with correction ($m = 0.3$, d) for otherwise standard parameters with the corresponding fits and their parameter values in the legend.

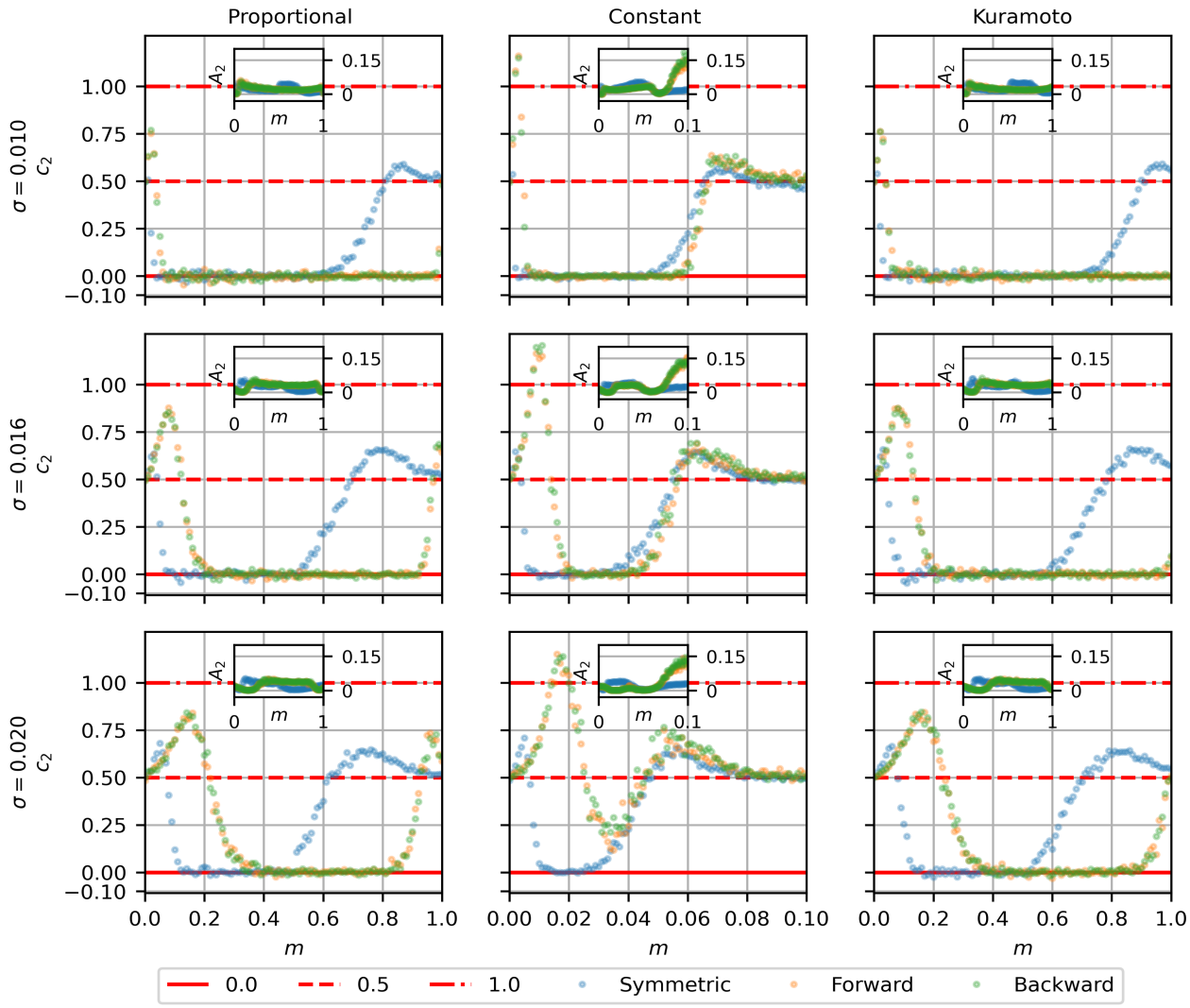


Figure 6.23 Temporal Surface Interface Width Results c_2. Fit parameter c_2 and A_2 (insets) of the long time limit as illustrated in fig. 6.22 as a function of the proportionality constant m for proportional correction (first column), constant correction (second column) and Kuramoto correction (third column) as well as symmetric, forward and backward correction (different colors) and standard deviations of the update noise (different rows). Values for c_2 for EW- and KPZ-class (0) as solid line, for random growth (0.5) as dashed line and for ballistic growth (1.0) as dash-dotted line.

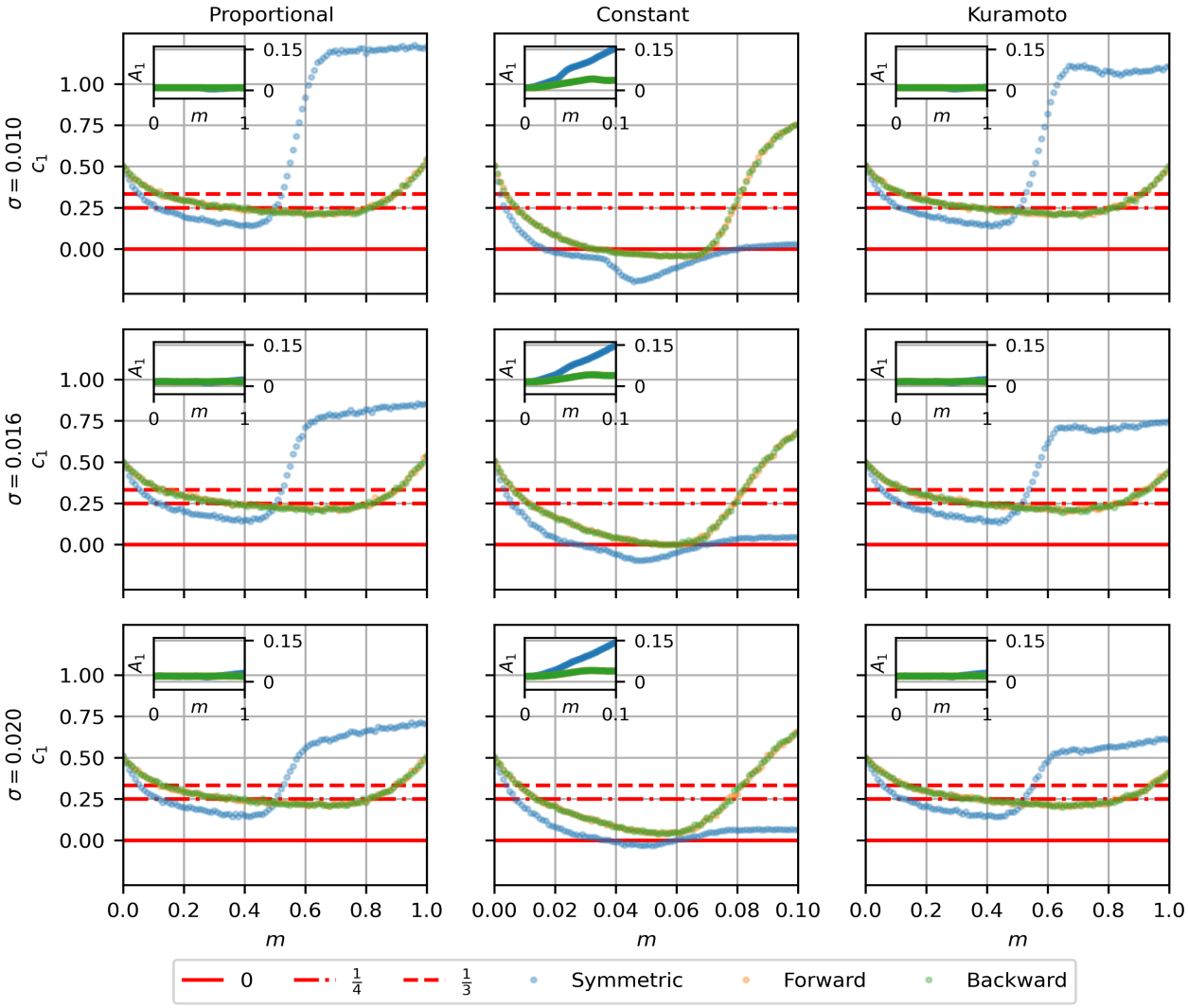


Figure 6.24 Temporal Surface Interface Width Results c_1. Fit parameter c_1 and A_1 (insets) of the small time limit as illustrated in fig. 6.22 as a function of the proportionality constant m for proportional correction (first column), constant correction (second column) and Kuramoto correction (third column) as well as symmetric, forward and backward correction (different colors) and standard deviations of the update noise (different rows). Values for c_1 for the Edward-Wilkinson Class ($\frac{1}{4}$) as dash-dotted line and for the KPZ class ($\frac{1}{3}$) as dashed line.

6.6 Discussion and Further Avenues

In this chapter, pattern formation with CA in the context of arising asynchrony was investigated. A minimalistic model of discrete updates with noisy timing was extended to include a communication delay and various correction schemes to alleviate the effects of noise in timing processes. It was found that in the context of pattern formation, communication and information processing delay can be a productive force acting as a buffer against asynchrony. If the patterning is coupled to the synchronization, but not the other way round, patterning dynamics which reach steady states quickly are in general robust against asynchrony. If the synchronization is additionally coupled to the patterning dynamics, also rules which have a highly variable pattern formation process show robustness against asynchrony. Conservation of synchronization can be achieved by using correction mechanisms which are either only based on the information if a neighboring cell is faster or slower, or use the exact time difference in a linear fashion or in a model similar to the classic Kuramoto model, with the latter two being equally more performant than the constant correction scheme. Yet, these correction schemes cannot reach the performance of a "perfect" correction. The schemes can deal with quenched as well as annealed noise, can handle systems growing by cell division at the "tip" or anywhere in the system, different boundary conditions and system geometries. The strength of the noise that can be handled depends apart from the efficiency of the correction on the amount of information that is received and the amount of delay acting as a buffer. The process can also be interpreted in the context of the physics of surface growth, showing similar limiting characteristics regarding the "roughness" of the surface of update times.

The connection of surface physics models and temporal surfaces has already been made in the context of parallel discrete-event simulations [202]. Using a nearest neighbor update model which additionally guarantees causality and is disturbed by a Poisson noise, they show a behavior close to the behavior of the Edward Wilkinson model. Their model is guaranteed to never loose causality, albeit at the cost of having an average update speed of approximately $\frac{1}{4}$ for $L \to \infty$, compared to approximately 1 for the small length studied in our model. Introducing additional random connections between cells successfully reduces the roughness of the temporal horizon [197].

Obvious extensions of the work done in this chapter are the inclusion of other motives investigated during this thesis, e.g., dynamic topologies and coupling of the synchronization to the mechanical properties of tissues (cp. chapter 8), inclusion of more complex pattern formation process and target measures (cp. chapter 4, chapter 5), and extending the range of the interactions (cp. chapter 7). Additionally, while here pattern formation in the context of mere conservation of an already present effective synchrony was investigated, pattern formation in the context of a system that is just reaching a synchrony may be interesting from a practical viewpoint. This could be realized for example via the Kuramoto correction scheme. A different strategy to model this problem is discussed in the following subsection section 6.6.1. One of the implicit assumptions of the models and observables was that the goal was to synchronize the cells not only in the same "phase", but also in the same generation. While this is important for the pattern formation aspect and there are examples of gradual maturation with each cell division [203], in the absence of a pattern formation mechanism the synchronization of the "phases" might be enough. Loosing this restriction could bring the model closer to biological systems like the segmentation clock [13] or genetic oscillations in *Anabaena* [149].

6.6.1 Outlook: Information Theoretical Inference of Phases During Synchronization of Heterogeneous Oscillators

This section describes collaborative work by Johannes Harth-Kitzerow, Torsten Enßlin, Ulrich Gerland and the author of this thesis, with the conceptual development performed by J.H.-K. and S.K. with the help of T.E. and U.G.. The detailed calculations were executed by J.H.-K. and are not shown here.

The goal of this section is to develop a Bayesian approach to infer the update times of the neighbors from the perceived time differences for potentially different update frequencies of the cells which are then adjusted by the correction algorithm. An overview over the algorithm can be found in fig. 6.25.

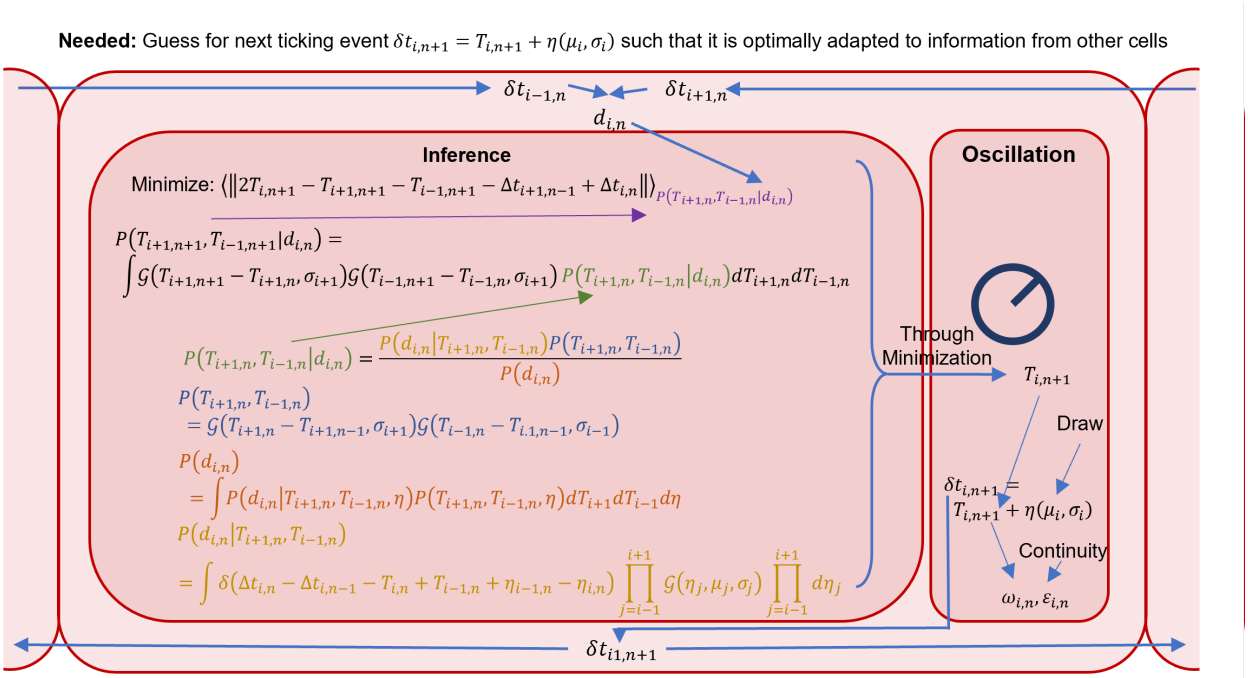


Figure 6.25 Inference Algorithm Overview. View of a single cell in the inference algorithm. The cell receives the data $d_{i,n}$ in the form of the temporal differences of the last ticks form the neighboring cells and tries to minimize a quadratic loss function to optimally choose its new update time $T_{i,n+1}$. The distribution with which the average is taken can be calculated assuming Gaussian priors and noises, as well as using Bayes theorem. From the new update time, the new ticking frequency and from a continuity assumption the new ticking phase can be determined.

Set-Up

Cell i ticks in generation n with a frequency $\omega_{i,n}$ which can differ from cell to cell and generation to generation. The ticking can be modeled by a ticking function

$$h_i(t) = \delta(1 - \cos(\omega_{i,n} t + \epsilon_{i,n})) \tag{6.16}$$

with i denoting again the cell and n denoting again the generation. The time difference between ticks without noise is then

$$T_{i,n} = \frac{2\pi}{\omega_{i,n}} + \varepsilon_{i,n} - \varepsilon_{i,n-1} = \frac{2\pi}{\omega_{i,n}} + \Delta\varepsilon_{i,n} \tag{6.17}$$

with $\Delta\varepsilon_{i,n} = \varepsilon_{i,n} - \varepsilon_{i,n-1}$. Assuming the ticking function changes smoothly at the nth update at time t_n,

$$cos(\omega_{i,n-1} t_n + \epsilon_{i,n-1}) = cos(\omega_{i,n} t_n + \epsilon_{i,n}) = 1 \tag{6.18}$$

is required, thus

$$\omega_{i,n} t_n + \epsilon_{i,n} \overset{!}{\in} 2\pi\mathbb{Z} \tag{6.19}$$

$$\omega_{i,n-1} t_n + \epsilon_{i,n-1} \overset{!}{\in} 2\pi\mathbb{Z}. \tag{6.20}$$

The updated phase $\epsilon_{i,n}$ can thus without loss of generality be chosen from

$$(\omega_{i,n} - \omega_{i,n-1}) t_n + \Delta\varepsilon_{i,n} = 0 \tag{6.21}$$

to be

$$\epsilon_{i,n} = \epsilon_{i,n-1} + (\omega_{i,n-1} - \omega_{i,n}) t_n. \tag{6.22}$$

Deviations from the optimal behavior will again be modelled as diffusive drift by a Gaussian, giving an effective ticking time of

$$\delta t_{i,n} = T_{i,n} + \eta(0, \sigma_i). \tag{6.23}$$

A cell i receives information about its neighbors if they have ticked before cell i, which is for two neighbors indexed by $i-1$ and $i+1$

$$d_{i,n} = \left(\min\left(0, -\Delta t_{i,n}\right), \min\left(0, -\Delta t_{i+1,n}\right)\right) \tag{6.24}$$

with

$$\Delta t_{i,n} := \sum_{k=0}^{n} \left(\delta t_{i,k} - \delta t_{i-1,k}\right). \tag{6.25}$$

To simplify the equations, in the following it is assumed that the data is fully available, i.e.,

$$d_{i,n} = \left(-\Delta t_{i,n}, -\Delta t_{i+1,n}\right). \tag{6.26}$$

Loss Function

The goal of the cell is to minimize the time difference of the next ticking events, i.e., the time differences

$$T_{i,n+1} - T_{i+1,n+1} - \Delta t_{i+1,n} \tag{6.27}$$

$$T_{i,n+1} - T_{i-1,n+1} - \Delta t_{i,n}. \tag{6.28}$$

The loss function now effectively determines the correction scheme and different variations can be investigated. An example is a quadratic loss function, in the noiseless case

$$l = \langle \| 2T_{i,n+1} - T_{i+1,n+1} - T_{i-1,n} - \Delta t_{i+1,n} + \Delta t_{i,n} \| \rangle_{P(T_{i+1,n+1}, T_{i-1,n+1} | d_{i,n}, \Delta t_{i+1,n-1}, \Delta t_{i+1,n-1})}. \tag{6.29}$$

With noise, the loss function becomes

$$l = \langle \| 2\delta t_{i,n+1} - \delta t_{i+1,n+1} - \delta t_{i-1,n} - \Delta t_{i+1,n} + \Delta t_{i,n} \| \rangle_{P(T_{i+1,n+1}, T_{i-1,n+1} | d_{i,n}, \Delta t_{i+1,n-1}, \Delta t_{i+1,n-1})}. \tag{6.30}$$

Bayesian Model

For developing a Baysian inference model for $P\left(T_{i+1,n+1}, T_{i-1,n+1} | d_{i,n}, \Delta t_{i+1,n-1}, \Delta t_{i+1,n-1}\right)$, the prior information and the noise are assumed to be Gaussian distributed, thus

$$\begin{aligned} P\left(T_{i+1,n+1}, T_{i-1,n+1} | d_{i,n}, \Delta t_{i+1,n-1}, \Delta t_{i+1,n-1}\right) = \\ \int \mathcal{G}\left(T_{i+1,n+1} - T_{i+1,n}, \sigma_{i+1}\right) \\ \mathcal{G}\left(T_{i-1,n+1} - T_{i-1,n}, \sigma_{i-1}\right) \\ P\left(T_{i+1,n}, T_{i-1,n} | d_{i,n}, \Delta t_{i,n-1}, \Delta t_{i+1,n-1}, T_{i,n}\right) dT_{i+1,n} dT_{i-1,n}. \end{aligned} \tag{6.31}$$

The posterior is now determined using Bayes Theorem:

$$\begin{aligned} P\left(T_{i+1,n} T_{i-1,n} | d_{i,n}, \Delta t_{i,n-1}, \Delta t_{i+1,n-1}, T_{i,n}\right) = \\ \frac{P\left(d_{i,n} | T_{i+1,n}, T_{i-1,n}, T_{i,n}, \Delta t_{i,n-1}, \Delta t_{i+1,n-1}\right) P\left(T_{i+1,n}, T_{i-1,n}\right)}{P\left(d_{i,n} | \Delta t_{i,n-1}, \Delta t_{i+1,n-1}, T_{i,n}\right)}. \end{aligned} \tag{6.32}$$

The three components of the right hand side can be determined by the assumptions stated above. For the prior,

$$P\left(T_{i+1,n}, T_{i-1,n}\right) = \mathcal{G}\left(T_{i+1,n} - T_{i+1,n-1}, \sigma_{i+1}\right) \mathcal{G}\left(T_{i-1,n} - T_{i-1,n-1}, \sigma_{i-1}\right). \tag{6.33}$$

The likelihood is, again under Gaussian assumptions[6]

$$\begin{aligned} P\left(d_{i,n} | T_{i+1,n}, T_{i-1,n}, T_{i,n}, \Delta t_{i,n-1}, \Delta t_{i+1,n-1}\right) = \\ \int \delta\left(\Delta t_{i,n} - \Delta t_{i,n-1} - T_{i,n} + T_{i-1,n} + \eta_{i-1,n} - \eta_{i,n}\right) \\ \delta\left(\Delta t_{i+1,n} - \Delta t_{i+1,n-1} - T_{i+1,n} + T_{i,n} + \eta_{i,n} - \eta_{i+1,n}\right) \\ \mathcal{G}\left(\eta_{i-1}, \sigma_{i-1}\right) \mathcal{G}\left(\eta_i, \sigma_i\right) \mathcal{G}\left(\eta_{i+1}, \sigma_{i+1}\right) d\eta_{i-1} d\eta_i d\eta_{i+1}. \end{aligned} \tag{6.34}$$

[6] This step omits 10 lines of calculation by J-H.K.

The evidence $P(d)$ can be obtained from normalization or marginalization

$$P(d_{i,n}) = \int P(d_{i,n}|T_{i+1,n}, T_{i-1,n}, \eta_i)\, P(T_{i+1,n}, T_{i-1,n}, \eta_i)\, dT_{i+1,n} dT_{i-1,n} d\eta_i. \tag{6.35}$$

This model could now be solved either analytically or numerically.

7 How to Use Local Interactions to Refine Patterns Created by Global Signaling? - Coupling Gradients and Local Signaling

This chapter comprises work from a bachelors thesis by Ludwig Burger [204] supervised by M. Bojer and the author of this thesis as well as work by Mareike Bojer [116] in which the author of this thesis was involved conceptually and which was published jointly [205]. All were supervised by Ulrich Gerland.

7.1 Morphogen Gradients and Local Interactions

Axial pattern formation was formulated by Wolpert as the already previously discussed French Flag problem [19]. The main conceptual solution to this problem was a gradient model [20], where gradients of diffusible morphogens (fig. 7.1 (a), (b)) are read out by the individual cells and cell types are determined via a thresholding mechanism. This solution model received experimental underpinning in the years since, e.g., in *Drosophila* [25] or vertebrate neural tube [23]. Typically, through the thresholding mechanism different domains of gene expression are formed along the gradient axis as showcased by a French Flag. A central problem for pattern formation with this mechanism is noise in its possible various sources [14], e.g., stochastic production of morphogens [139, 206], local variations due to the stochastic nature of diffusive processes, or variations in the readout of the morphogen concentrations by the cells [14]. This leads to domain boundaries which are not precise but rather "rough" [14], which might have negative effects for the downstream development of the embryo. To straighten the boundaries, several mechanisms are possible. For example, cells can have different mechanical properties for different cell types and thus sort themselves into the "correct" domain via mechanical forces [151], e.g., mediated by Eph/Ephrin Signaling [151]. Another conceptually possible mechanism, which is explored here, is that cells additional to the information about the concentration of the morphogen at their position ("global signal") also interact with their direct neighbors (fig. 7.1 (c), (d)) locally and exchange information about their states ("local signal"). In the following sections a set of assumptions are made: Cells are assumed to be on a static grid, receiving (possibly noisy) signals from the morphogen gradient and their immediate neighbors (fig. 7.1 (b), (d)). The signal from the gradient, which is assumed to be an exponential profile [207], will be processed logarithmically, leading effectively to a linear gradient signal [208]. While on the left and right fixed boundary conditions with cells in the "correct" (according to a noiseless global signal) state are used, along the perpendicular axis periodic boundary conditions are chosen. Two models with different dynamics, signal processing and noise parametrization are employed. In the next section, first a equilibrium model similar to Hillenbrand et al. [208] is used to analyze a minimally constrained version of the problem of getting smooth boundaries via local interaction. In the third section a more specific, dynamical model is analyzed with the focus on which signal processing logic performs best when combining local and global signals in order to smooth boundaries.

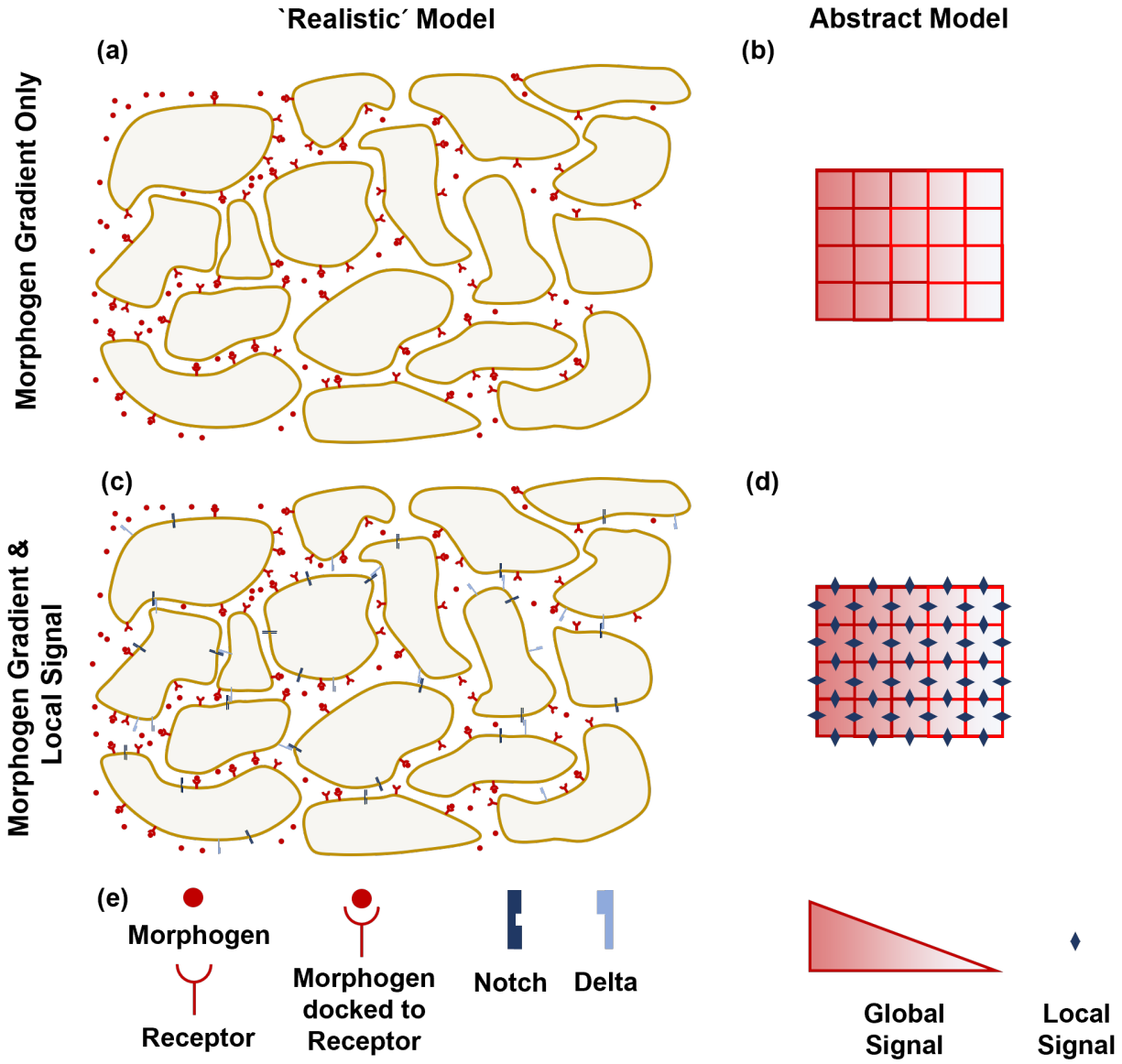


Figure 7.1 Models for Axial Pattern Formation. (a) In the classical solution to the French Flag problem of axial pattern formation diffusible morphogens set a gradient in extracellular space, which is read out by the cells and trough a thresholding mechanism converted into a state. (b) The realistic model is conceptually simplified in an abstract model where the cells are placed on a grid with the diffusible morphogens simplified in an field spanning the whole grid. Different cell types are marked by different colors of the cell boundary. (c, d) Same as in (a, b), but with additional local signaling. (e) Legend. (a) & (c) adapted from Bollenbach [209].

7.2 Equilibrium Model

7.2.1 Model Definition and Observables

In this section, it is assumed that cells are placed on a grid of size N x M and that they can have two states $s = -1$ and $s = +1$ (fig. 7.2). They try to align their states with an external field $h(x)$ which is chosen to be linear with a zero crossing in the middle of the grid

$$h\left(x\right) = J_e\left(x - \frac{M}{2}\right) \tag{7.1}$$

which is modeled by an "energy" function

$$\mathcal{H} = -\sum_i h(x) s_i(x). \tag{7.2}$$

Cells interact with their direct neighbors such that they favor to have the same state, modeled again by a corresponding "energy" function

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} s_i s_j. \tag{7.3}$$

Taking both local and global interactions together,

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} s_i s_j - \sum_s h(x) s_i(x) \tag{7.4}$$

describes the "energy" of the system, which is constructed now such that cells try to have a state that aligns with the external field and with their neighbors, i.e., a domain system with a smooth boundary minimizes the energy for appropriately chosen parameters. Each configuration $\{s\}$ has now an "energy" associated with it. To model the noise in the pattern formation process it is assumed that each configuration is appearing with a certain probability depending on the "energy" of the configuration. The probability distribution is chosen to be of a Boltzmann type

$$p = \frac{1}{Z} \exp\left[\frac{-J \sum_{\langle i,j \rangle} s_i s_j - \sum_s J_e\left(x - \frac{M}{2}\right) s_i}{\eta}\right] \tag{7.5}$$

with η quantifying the noise and Z being a normalization constant

$$Z = \sum_{s_1 = \pm 1} \cdots \sum_{s_{N \cdot M} = \pm 1} \exp\left[\frac{-J \sum_{\langle i,j \rangle} s_i s_j - \sum_s J_e\left(x - \frac{M}{2}\right) s_i}{\eta}\right]. \tag{7.6}$$

This choice is motivated by the following reasons: (1) configurations with a straighter boundary at the right position have a higher statistical weight, (2) it is the maximally unbiased distribution for a fixed average "energy" [115], and (3) it is equivalent to the Ising model, an established toy model for locally interacting spin systems which has already been sucessfully employed to model developmental systems [210].

The individual cells are placed on a lattice - either a square lattice with von Neumann neighborhood, a hexagonal lattice, or an irregular lattice created by randomly shifting an initially regular hexagonal pattern of points randomly (with the random shifts drawn from a triangular distribution with variance r) and then defining cells by Voronoi tessellation (cp. appendix A.7). Boundary conditions are chosen to be fixed on the left and right end of the simulation, and periodic on top and bottom.

As main observable, the roughness is chosen which is defined as the ensemble average of the standard deviation of the interface position:

$$roughness = \left\langle \sqrt{\sum_y \left[x_{boundary}\left(y\right) - \left\langle x_{boundary}\left(y\right)\right\rangle_{configuration}\right]^2} \right\rangle_{ensemble} \tag{7.7}$$

It is obtained by Monte Carlo Sampling [211] of the configuration space, for implementation details see [204].

7.2.2 Results

The results of boundary formation for the different types of grids can be seen in fig. 7.2 (b) - (i). For a square grid, the roughness as a function of the intrinsic noise is plotted in (b). The curve can be subdivided into 3 regimes, a self stabilization regime (1), and first (2) and second (3) linear regime. In the self stabilization regime, the boundary is perfectly straight despite a nonzero noise. In the first linear regime, spin flips in the row directly adjacent to the boundary are possible. There is a comparably large energy barrier for the flip of a first, isolated spin at the boundary of $4J + J_e$, while every additional flip only costs J_e. In the second, also spin flips away from the boundary are possible. These have a very low probability of occurrence each due to their large energy penalty of $8J + J_e \left| x - \frac{M}{2} \right|$, but there exist a lot of configurations with these, and thus they can outweigh the changes directly adjacent to the boundary by their multiplicity. Additionally, through the large distance to the boundary position they also contribute with a greater amount towards the variance.

The larger the local interactions are, the later is the onset of the first linear regime (fig. 7.2 (c)). To model the dependency, the self stabilization regime and the first linear regime are approximated by linear fits and their intersection defined as the crossover point between the regimes or characteristic intrinsic noise η_C (fig. 7.2 (d)). Plotting the characteristic intrinsic noise as a function of the local coupling constant yields indeed a linear relationship (fig. 7.2 (e)).

For the hexagonal grid, the roughness plotted as a function of the intrinsic noise follows a qualitatively different behaviour (fig. 7.2 (f)) as there is no self stabilization regime (1). This is due to the fact that in a hexagonal grid cells position with their midpoint directly at the coordinate of the zero crossing of the external field have, assuming all other cells have a "correct" state, no energy preference for either state. Furthermore, again assuming all other cells have a "correct" state, cells half a cell diameter distant from the zero crossing of the external field have 3 black and 3 yellow neighbors, and thus only an energy penalty proportional to J_e but independent of J when switching states.

For irregular grids, a similar behaviour as for the hexagonal grid occurs (fig. 7.2 (g)), although the initial roughness at extremely low noise is larger due to the grid irregularities and the subsequent increase slower since the number of state changes whose energies does not depend on the local interaction energies is smaller due to the irregularities. The second regime this time is nearly completely stable for two reasons: For these noises, the energy required to change states whose changes are penalized by local interactions is a nearly unreachable barrier. Changes which only depend on the energy of the gradient however are already realized in the roughness calculations also for lower intrinsic noise levels. Calculating again a characteristic noise analogously to the procedure for square grids, only this time between the second and third regime, yields again a nearly linear dependency of the characteristic noise as a function of the local interaction constant J (fig. 7.2 (h)). As intuitively expected, the larger the noise of the grid generation parameterized by r and thus the more irregular the grid (fig. 7.2 (i)), the slower the increase of the characteristic noise (fig. 7.2 (h)).

Summarized, local interactions smooth boundaries in regular as well as irregular grids - the stronger the interaction the lower the roughness of the boundary (fig. 7.2 (e), (h)). While for square grids the local interactions facilitate a regime with a completely straight boundary, for hexagonal and irregular grids this is not possible due to state changes with no or extremely low energy costs. Yet, as soon as the intrinsic noise is large enough to realize these transitions, again a self stabilization regime appears with relatively constant roughness as a function of the intrinsic noise until the noise is large enough to overcome also barriers introduced by the local interactions. In general, disordered grids perform worse than ordered grids. Interestingly, this is different than the behaviour of the Ising model without external field. Using the magnetization as order parameter and the critical temperature as observable, Eltinge showed that the transition from ordered to disordered phases occurs at higher temperature for disordered lattices than for ordered lattices - it is conjectured that this is due to the on average larger number of nearest neighbors since a similar increase can be seen by going from square to hexagonal lattices [212].

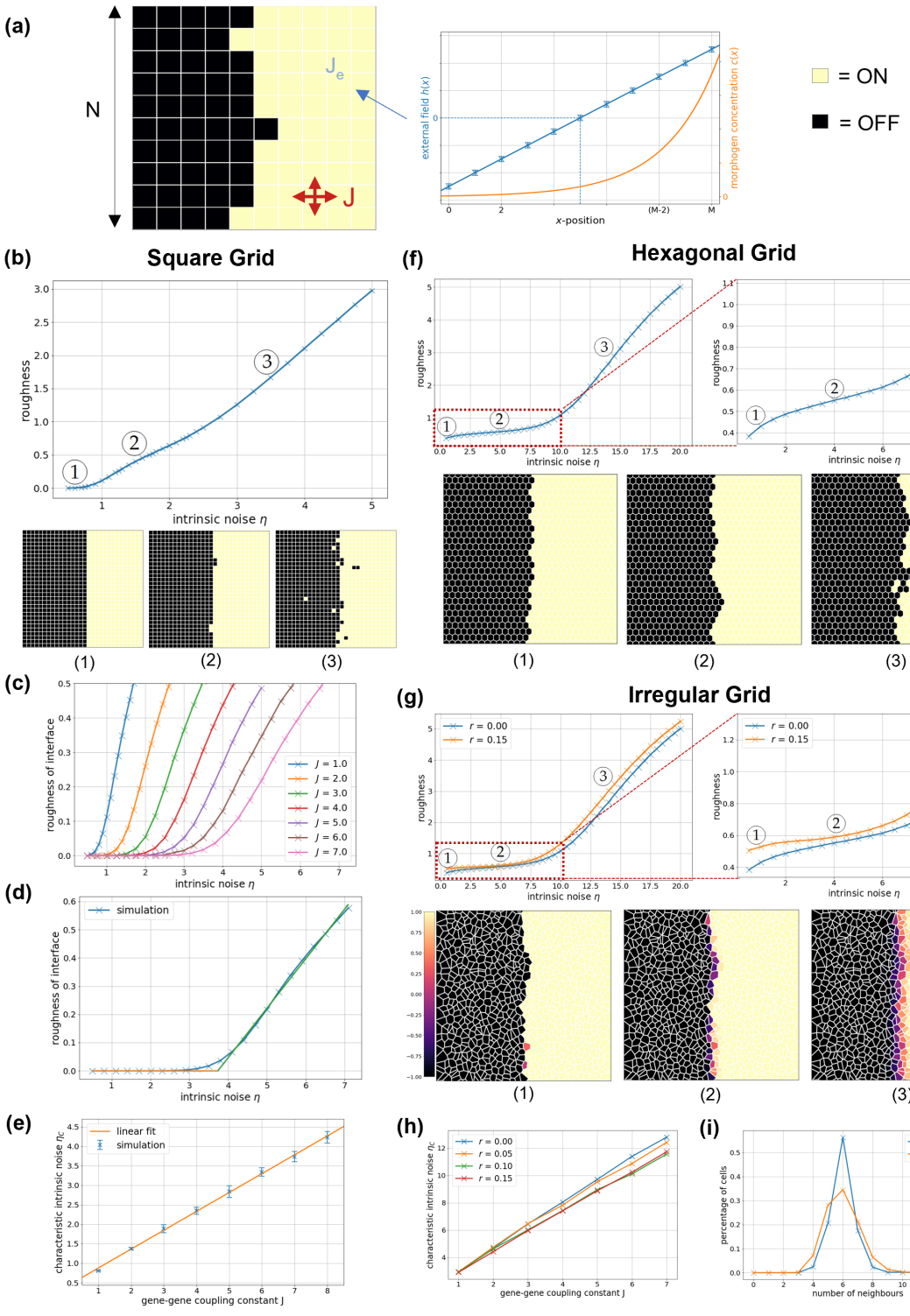

(a)
N
J_e
J
external field $h(x)$
morphogen concentration $c(x)$
x-position
= ON
= OFF
(b)
Square Grid
roughness
intrinsic noise η
(1)
(2)
(3)
(f)
Hexagonal Grid
(c)
roughness of interface
J = 1.0
J = 2.0
J = 3.0
J = 4.0
J = 5.0
J = 6.0
J = 7.0
(d)
simulation
(e)
linear fit
simulation
characteristic intrinsic noise η_c
gene-gene coupling constant J
(g)
Irregular Grid
r = 0.00
r = 0.15
(h)
r = 0.00
r = 0.05
r = 0.10
r = 0.15
(i)
r = 0.05
r = 0.10
percentage of cells
number of neighbours

Figure 7.2 Equilibrium Model. Adapted from Burger (2019) [204] with permission. (a) Basic model for a square grid (MxN) with a exponential morphogen concentration creating a linearly varying signal with proportionality constant J_e, as well as local signaling with proportionality constant J. Boundary conditions are fixed (horizontal direction) and periodic (vertical direction). **Square grid:** (b) Roughness as function of the intrinsic noise η. Three regimes are marked corresponding to the exemplary grids below. In regime (1), self stabilization, the boundary is perfect despite nonzero noise. In regime (2) spins flip in the row directly adjacent to the boundary. In regime (3) spin flips far away are possible. (c) Roughness as a function of the intrinsic noise for different local gene-gene coupling constant J. The larger the coupling constant, the lower the roughness. (d) Exemplary roughness as a function of the intrinsic noise with fits of the first two regimes to determine the characteristic intrinsic noise η_C at their intersection. (e) Characteristic intrinsic noise η_C as a function of the gene-gene coupling constant, together with a linear fit function. **Hexagonal Grid:** (f) Roughness as a function of the intrinsic noise with different regimes, as well as exemplary configurations for the regimes below. In contrast to the square grids, there is no self stabilization regime. **Irregular Grids:** (g) Roughness as a function of the intrinsic noise for different irregularities of the grid, again with different regimes marked and exemplary configurations below. (h) Characteristic intrinsic noise as a function of the gene-gene coupling constant J for different irregularities of the grid calculated similar to the square grids but for the second and third marked regimes. The larger the irregularities, the smaller the characteristic intrinsic noise for a given coupling constant. (i) Relative amount of cells that have a certain number of neighbors for different irregularities of the grid.

7.3 Nonequilibrium Model

7.3.1 Model Definition and Observables

After having explored that local interactions can indeed smoothen boundaries established by gradients in the presence of noise in equilibrium models, now the focus will be shifted towards a nonequilibrium model. Again, a set of cells will be placed on a square grid, with each cell having a state $c \in \left\{-\frac{1}{2}, \frac{1}{2}\right\}$ being subjected to a signal from a morphogen gradient $s^G_{i,j}$ and their nearest neighbors $s^{\mathcal{L}}_{i,j}$ (fig. 7.3 (a), (b)). The state of a cell $c_{i,j}(t)$ at position i, j and time t is evolved in discrete steps and is a direct function of these two signals

$$c_{i,j}(t+1) = \mathcal{L}\left[s^G_{i,j}(t), s^L_{i,j}(t)\right] \tag{7.8}$$

The gradient is again modeled as a linear function of slope m with noise introduced by an explicit noise term ξ^G_{ij} which is chosen to be Gaussian white noise with zero mean and standard deviation σ^G

$$s^G_{i,j}(t) = mi + \xi^G_{ij}(t) \tag{7.9}$$

The local signal is calculated as the sum of the states of the neighbors plus again an Gaussian white noise term with zero mean and standard deviation σ^G accounting for any inaccuracies due to errors in signal production, transmission and readout

$$s^L_{i,j}(t) = \sum_{(k,l)\in \text{neighbors}(i,j)} c_{k,l}(t) + \xi^L_{ij}(t) \tag{7.10}$$

Left for definition is now to specify how the local and global signal give rise to the state in the next time step (fig. 7.3 (c)). In a situation with only a gradient the update function $\mathcal{L}$ is defined to be consistent with the gradient solution of the French Flag problem by introducing a threshold a below which the state in the next time step is $-\frac{1}{2}$, and above which it is $\frac{1}{2}$:

$$\mathcal{L}^{GRAD}\left[s^G_{i,j}(t)\right] = \Theta\left[s^G_{i,j}(t) - a\right] - \frac{1}{2} \tag{7.11}$$

If the state of the cell is defined by the concentration of a single protein and the signal acts as activator for the expression of said protein, this is consistent with having an Hill input function for the expression of the protein with infinite Hill coefficient [89]. Taking the local signal into account as well, both gradient and local signal can give rise to an expression in different ways. Assuming again the state of the cell to be defined

by the concentration of a protein, its expression could require both local and global signal to be present, i.e., both activator signals binding to a promotor for the expression of the associated gene, leading to an AND logic [88, 89, 213]

$$\mathcal{L}^{AND}\left[s_{i,j}^{G}\left(t\right),s_{i,j}^{L}\left(t\right)\right]=\Theta\left[s_{i,j}^{G}\left(t\right)-a\right]\Theta\left[s_{i,j}^{L}\left(t\right)-a^{AND}\right]-\frac{1}{2}. \tag{7.12}$$

Alternatively, either could be sufficient, giving rise to an OR gate [88, 213, 89]

$$\mathcal{L}^{OR}\left[s_{i,j}^{G}\left(t\right),s_{i,j}^{L}\left(t\right)\right]=\Theta\left[s_{i,j}^{G}\left(t\right)-a\right]+\Theta\left[s_{i,j}^{L}\left(t\right)-a^{OR}\right]-\Theta\left[s_{i,j}^{G}\left(t\right)-a\right]\Theta\left[s_{i,j}^{L}\left(t\right)-a^{OR}\right]-\frac{1}{2}. \tag{7.13}$$

Finally, both could act on the same promoter and thus can be modeled by a SUM input function [89]

$$\mathcal{L}^{SUM}\left[s_{i,j}^{G}\left(t\right),s_{i,j}^{L}\left(t\right)\right]=\Theta\left[s_{i,j}^{G}\left(t\right)+s_{i,j}^{L}\left(t\right)-a\right]-\frac{1}{2}. \tag{7.14}$$

In the following, these will be called AND, OR, and SUM rule. While for the SUM rule the thresholding takes place relative to a single threshold, for the AND and the OR rule the thresholding needs to be done with respect to a specific threshold a^{AND} and a^{OR}, respectively, before invoking the eponymous logic operation. If the gradient formation is set to work from both homogeneous as well as random initial conditions and at the same time reduce the used observables (see below) optimally, it can be shown numerically that $a^{AND}=-1$ and $a^{OR}=+1$ are the only reasonable choices [205]. To reduce the parameter space further, the total noise is defined as

$$\xi:=\xi^{L}+\xi^{G} \tag{7.15}$$

with a total variance of

$$\sigma^{2}:=\left(\sigma^{L}\right)^{2}+\left(\sigma^{G}\right)^{2} \tag{7.16}$$

Then a parameter

$$\alpha:=\frac{\left(\sigma^{L}\right)^{2}}{\sigma^{2}} \tag{7.17}$$

can be defined to characterize the relative noise strength. This parameter is for most of the following paragraphs chosen to be $\alpha=\frac{2}{mL+2}$. Further, for the rest usually the paramter sets m and $\frac{a}{m}$ are used instead of a and m.

All updates are performed synchronously, yet the stationary results for the main observables (see below) qualitatively agree for asynchronous, random update [205].

As first observable, the boundary position $\mathcal{B}$, defined as the average number of cells with state $-\frac{1}{2}$,

$$\mathcal{B}:=\frac{1}{L}\sum_{i,j}\delta\left(c_{i,j},-\frac{1}{2}\right), \tag{7.18}$$

is chosen. To measure the fluctuations around this average, as second observable the Fuzziness $\mathcal{F}$ as the number of cells in the wrong state with respect to the boundary position (regardless of their distance to said boundary),

$$\mathcal{F}:=\frac{1}{L^{2}}\left(\sum_{i<\mathcal{B},j}\delta\left(c_{i,j},\frac{1}{2}\right)+\sum_{i>\mathcal{B},j}\delta\left(c_{i,j},-\frac{1}{2}\right)\right), \tag{7.19}$$

is used. The fuzziness as an observable is able to include outlier cells which are enclosed completely by other cells of different cell types as well as a rough boundary between homogeneous groups of cell types.

7.3.2 Results

Kinetics

In contrast to equilibrium model of the previous chapter, the non-equilibrium model allows to study if and how a steady state is reached from different initial conditions. Figure 7.3 (d) shows boundary position and fuzziness as a function of time for the AND and SUM rule for different initial conditions (random, all $\frac{1}{2}$, or all $-\frac{1}{2}$) and different noise levels. For the AND rule, starting from an all $-\frac{1}{2}$ (OFF) grid, the boundary moves roughly one cell on average towards the middle of the system (see also zoom ins in insert (i)) since the structure of the rule needs both local and global signal to switch to the on state with the local signal initially being available only at the boundary. For random or all ON initial condition, local signals are (for sure in the all ON case, usually in the random case) available throughout the system and thus a boundary in the middle of the system quickly forms. In all cases, after forming the boundary in the middle of the system the roughness quickly reaches its stationary value. It can be shown that the OR is qualitatively similar to the AND rule with inverted state values [205].

For the SUM rule and large noise, the boundary position and fuzziness quickly converge to their stationary values. For small noise, the boundary position converges to a value close to the value expected by the pure gradient of $\frac{L}{2}$, after which the signal from the gradient is not enough to outweigh the local signal until a random fluctuation due to the noise introduces a "seed" (inset (iii), leftmost grid). The seed gradually spreads through out the boundary row by flipping the neighbors (inset (iii), next 3 grids) increasing the fuzziness temporarily, and finally consolidates into a boundary position at the zero crossing of the gradient (inset (iii), rightmost grid).

Noise Dependency

To investigate the dependency of the pattern formation on the noise more thoroughly, in fig. 7.3 (e) the dependency of the temporal averages of boundary position and fuzziness as well as the time necessary to reach the stationary values are depicted for the AND, OR and SUM rule as well as for the pure gradient (dark colors), together with predictions by analytical theory (light colors), all for a specific choice of system size and parameters of the gradient. The initial conditions are "all off" except for the transition time where the maximum of "all on" and "all off" initial conditions was taken.

The boundary position for the SUM rule is for all noise levels at the same position as the boundary position of the gradient, while for AND and OR rule the boundary position deviates as a roughly linear function of the noise level. The analytical values closely approximating the numerical results are derived based on the fact that for a stable stationary boundary at position i the probability of a fluctuation to disturb this boundary to the left must be smaller than the probability to disturb the boundary at $i-1$ to the right. The same must hold true for a disturbance of the boundary at i to the right and the boundary to the left at $i+1$ [205].

The fuzziness of all rules is a monotonically increasing function of the noise level and in general lower than the pure gradient behavior. The SUM rule in general outperforms the AND and OR rule and has a noticeable region of small noise where the fuzziness effectively stays zero, i.e., even though there is finite noise present, the SUM rule is able to make any fuzziness of a pure gradient rule disappear.

To get a proper upper bound for the transition time to the stationary value, for the AND and OR rule, the maximum time for an all ON and all OFF grid, respectively, is taken. Thus, it is independent of the noise for these two cases because of the mechanism detailed in the previous section. For the SUM rule it is a monotonically decreasing function - the larger the noise the more random flips appear aiding in the formation of the stationary pattern as discussed in the previous section.

Having discussed now the dependencies for a specific ratio of local to global noise and specific parameters of the gradient, in fig. 7.3 (f), for a fixed total noise $\sqrt{3}\sigma = 2$, for SUM, AND and OR rule the fuzziness is plotted as a function of the relative noise ratio α for different levels of $\frac{a}{m}$. For all rules, the fuzziness with rule is lower than the fuzziness of the pure gradient well beyond the half, indicating that even if the

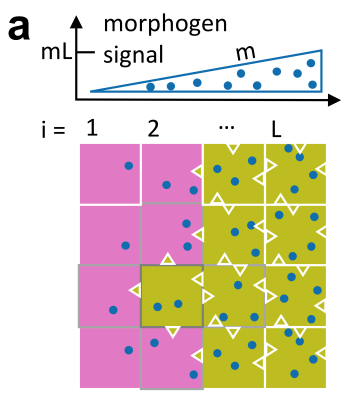
local interactions are more influenced by noise than the gradient signal itself the local rule is able to reduce boundary fuzziness for different parameters of the gradient.

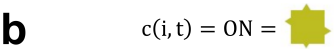
Scaling

A crucial property demanded for a solution of the French Flag problem [19] is that the solution is able to scale with system size. To investigate this property, the relative fuzziness and the boundary position relative to the system size for fixed noise and scaling gradient parameters are plotted in fig. 7.3 (g) as a function of the system size. Ignoring finite size effects for small lengths, the relative boundary position and the relative fuzziness are independent of the system size indicating that all solutions with rules scale similarly as the pure gradient solution with system size.

Gradient Dependency

The remaining parameters that ask for a systematic discussion are the gradient slope m and the threshold a. In fig. 7.3 (h) the average as well as the maximum and minimum values over the threshold a of the ratio of the fuzziness of a logic to the fuzziness of the pure gradient $\frac{\mathcal{F}^{Logic}}{\mathcal{F}^{Grad}}$ are plotted as a function of the gradient slope. For all investigated values of the gradient and threshold parameters, the AND and OR rules outperform the gradient logic and are themselves outperformed by the SUM rule.





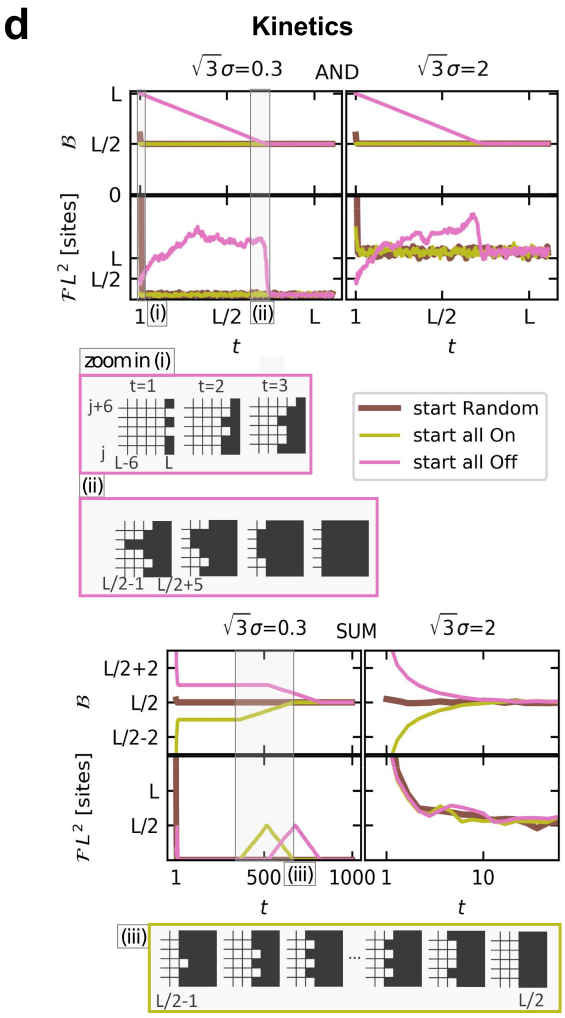
c

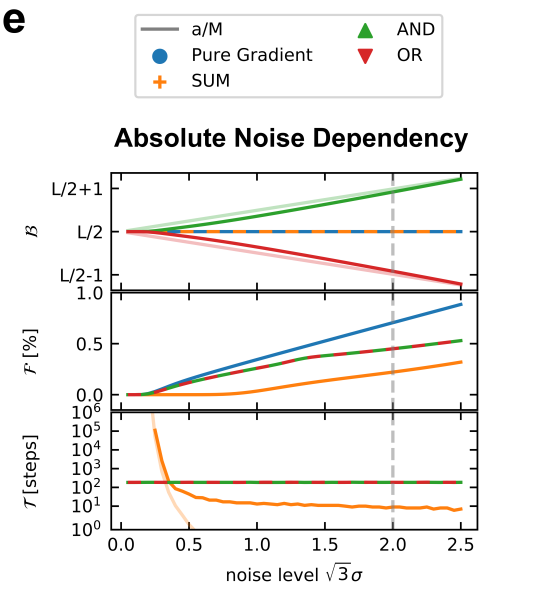
[Grad]: if [globalsignal $>$ a]: On; else: Off

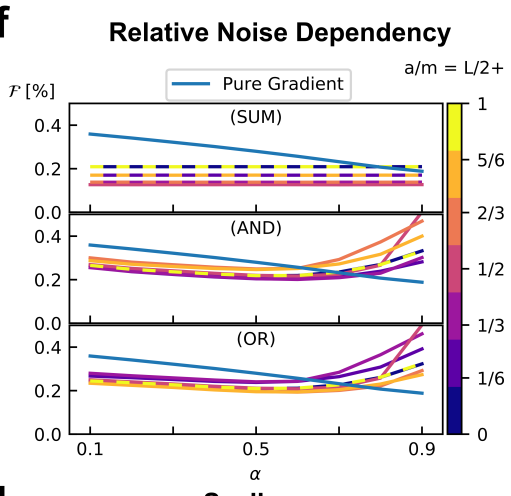
[SUM]: if [(globalsignal + localsignal) $>$ a]: On; else: Off

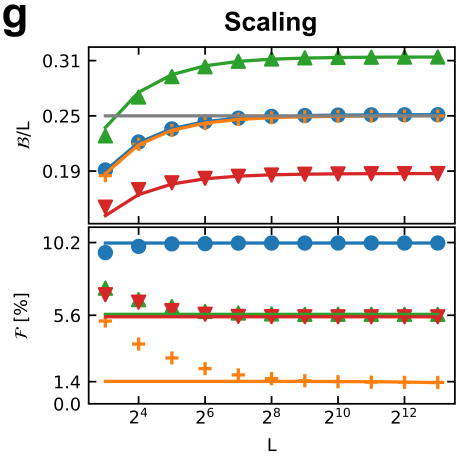
[AND]: if $\left[(\text{globalsignal} > a) \text{AND } (\text{localsignal} > a^{\text{And}})\right]$: On; else: Off

[OR]: if $\left[(\text{globalsignal} > a) \text{OR } (\text{localsignal} > a^{\text{OR}})\right]$: On; else: Off









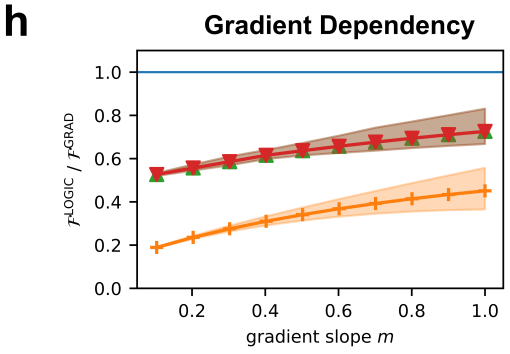


Figure 7.3 Nonequilibrium Model. Adapted from Bojer, Kremser, Gerland; 2022; *Robust boundary formation in a morphogen gradient via cell-cell signaling*; Phys. Rev. E 105, 064405 [205] ©2022 American Physical Society. **Model Definition:** (a) Cells in a square grids receive a morphogen signal which is linear as a function of the horizontal coordinate (i) as well as local signals from their neighbors. (b) Cells can in either one of two states ON and OFF (c) The state of a cell is determined by the local and global signal, as well as how are they coupled together. Either only the global signal is compared to a threshold a [Grad], both signals are summed up and compared to a threshold (a), or both are individually compared to thresholds a, a^{And} or a^{Or}, respectively, and then logically coupled via the AND or OR logic. **Results:** (d) **Kinetics:**. Boundary position and boundary fuzziness as a function of time for different standard deviations of the noise σ for the AND (upper panels) and SUM (lower panels) logic for different initial grid configurations, together with exemplary grid configurations at the marked times. AND logic: For "Random" and "all On" configurations, the boundary position $\frac{L}{2}$ is reached quickly, while for "all Off" the approach is gradually step by step as illustrated in (i). The fuzziness similarly reaches a plateau value for "all On" and "Random" initial conditions quickly, whereas it takes time until the boundary reaches $\frac{L}{2}$ to reach the plateau value for the "all Off" initial condition. SUM logic: The boundary position gets close to the plateau value $\frac{L}{2}$ quickly, for large noise reaching it nearly instantaneously, while it takes a long time for the low noise case. The fuzziness quickly reaches a low value for all initial conditions in the low noise case, only increasing and falling again when the boundary is reaching $\frac{L}{2}$ as illustrated in (iii). In the large noise case the fuzziness falls to a finite plateau value. (e) **Absolute noise dependency:** Boundary position, fuzziness and time to steady state as a function of the noise level for all presented logics. Simulation results in bold, analytical model in faint colors. The SUM rule has the same boundary position as the gradient, whereas for the AND (OR) rule it increases (decreases) for larger noise. For the SUM rule, the fuzziness stays 0 for finite noise levels, whereas for AND and OR logic it increases, although it is lower than with a pure Gradient. All logics reach the steady state quickly except for the SUM logic. (f) **Relative noise dependency:** Fuzziness as a function of the ratio of the local and total noise α for different logics and ratios $\frac{a}{m}$. All logics perform better than the gradient logic even when the noise in the local signal is larger than the noise of the gradient but not too large (g). **Scaling:** Boundary position per system size as well as relative fuzziness as a function of system size, all reaching constant values for larger system sizes. (h) **Gradient Dependency:** Fuzziness of the logics relative to the fuzziness of the gradient logic for different slopes of the gradient m, showing that in all cases the logics outperform the pure gradient.

7.4 Discussion and Further Avenues of Research

To sum up, local interactions can reduce the roughness/fuzziness of a boundary formed by a gradient mechanism in equilibrium and non-equilibrium models on various grids (fig. 7.2 (b), (f), (g)) and for various interaction rules of local and global signal (fig. 7.3), while keeping the scaling property of the underlying gradient model (fig. 7.3 (g)). In the non-equilibrium case it was shown that the SUM rule outperforms the other interaction rules regarding its reduction of fuzziness (fig. 7.3 (d) - (h)) and conservation of the boundary position imposed by a pure gradient rule (fig. 7.3 (e), (g)).

The simplicity of the models also has some serious drawbacks concerning the reach and strength of the statements that can be made. For the equilibrium model, while qualitatively capturing the features of a local interaction that aims to align cell types locally, a more thorough motivation of the chosen modeling assumption is not available. Additionally, being an equilibrium model, all temporal effects can inherently not be modeled like averaging signal measurements over time [14] and not only over space due to the local interactions. The latter is also partially true for the non-equilibrium model that only allows for temporal pooling of the previous values of the neighboring cells. Although several, including irregular, grids, were investigated for the equilibrium models, only square grids were used in the non-equilibrium model. Extending the models to different grids by either including experimentally sourced irregular grids, extending the models to 3 dimensions, including a mechanical model to also consider cell movements and topological grid changes (cp. chapter 8), or taking into account the dependency of the local signal strength on the shared interaction surface of the cells [62] are variations that should be readily implementable in the current code-base. It can be speculated that especially the latter can reduce clustering effects found in the equilibrium model for irregular grids [204].

Despite these drawbacks, the chosen models have advantages regarding their quick numeric solvability and thus multitude of parameter combinations and research questions can be probed. In both cases, it

could be established for a multitude of parameter combinations that local interactions can indeed reduce the roughness/fuzziness of a boundary formed by a gradient mechanism and allowed for the comparison of different mechanisms of combining the local and global signals in the non-equilibrium case. This provides theoretical support not only for developmental systems, but also for the challenging task of synthetic mimicry of boundary formation [214]. Exemplary platforms of synthetic pattern formation have demonstrated communication based on gradients [31, 215] and local interactions [30, 31]. Different communication logics can also be implemented and controlled in bacterial systems [216, 217]. The previous chapter supports the choice of the SUM logic versus AND and OR logic as a design principle. It can be speculated that the fact that for the SUM logic the signals are added up before thresholding and thus the variances of the noise are only added as sum of squares leads to the advantageous performance over the AND and OR logic [205]. However, it can be argued and experimentally supported in a bacterial system that for rare signals thresholding followed by signal combination is preferential to the reversed procedure [205, 218].

In a recent study by Lee et al. [87], for the formation of the primitive streak in the chick embryo two different models for interpreting morphogen gradients where investigated. The first compared the value of morphogen signal to a threshold, in our notation roughly equivalent to

$$\mathcal{L} = \Theta\left[s_i^G - a\right] \tag{7.20}$$

in the formulation of the model here. The second compared the signal to the average neighborhood signal,

$$\mathcal{L} = \Theta\left[\frac{s^G_{AverageNeighbors} - s_i^G}{s_i^G} - a\right] \tag{7.21}$$

with the size of the neighborhood may be larger then the nearest neighbors. Comparing these two models to data from different experimental settings lead Lee et al. to the conclusion that the second model combining local and global signaling is better suited to explain the data. It should be noted that, while the study shows the possibility for combining local and global signals in developmental settings, in their case, the combination of local and global signaling is primarily used to determine the boundary position, while in the case here the focus is on boundary fuzziness.

8 How to Include Cell Movement, Division and Death? - Coupling Vertex Models and Cellular Automata

This chapter consists of work done by Sabina Orazov in her master thesis [219] supervised by Mareike Bojer and the author of this thesis, all supervised by Ulrich Gerland.

8.1 Basic Vertex Model

The 2D vertex model (cp. fig. 8.1 a, b) in its usual form [220, 221, 222] is defined by specifying a set of vertices $\left\{\vec{R}_i\right\}$ which determine the cell boundaries on a 2D surface, and an energy function specifying the dynamics of the vertices of the form

$$E\left(\left\{\vec{R}_i\right\}\right) = \sum_{\alpha} \frac{K}{2}\left(A_\alpha - A_0\right)^2 + \sum_{\alpha_1,\alpha_2 \text{ adjacent}} \Lambda_{\alpha_1,\alpha_2} l_{\alpha_1,\alpha_2} + \sum_{\alpha} \frac{\Gamma}{2} L_\alpha^2 \tag{8.1}$$

The first term describes the resistance of cells to changes in area (volume), e.g., compression of the cell is met by a corresponding pressure with A_0 being the volume the cells try to realize if not other cell would be present and K a weighting factor for this contribution. The second term describes contributions due to e.g., cell-cell adhesion by cadherine molecules, as well as contracting forces, and depends on the interaction length (surface) between two cells α_1 and α_2, l_{α_1,α_2}. The third term is a proxy for the effects of actomyosin cortical tension and thus dependent on total circumference length of the cell α, L_α. Differentiating eq. (8.1) with respect to the vertex positions leads to an expression for the force on a single vertex $\vec{R}_i$.

$$\vec{F}_{net,\,i} = \frac{\partial}{\partial \vec{R}_i} E\left(\left\{\vec{R}_i\right\}\right) \tag{8.2}$$

The dynamics is then given in an overdamped setting by[1]

$$\frac{d}{dt}\vec{x}_i \sim \vec{F}_i \tag{8.3}$$

To prevent the dynamics from being stuck in local minima, the proportionality constant for the line tension term is not kept constant, but follows noisy dynamics as described in [221]:

$$\dot{\Lambda}_{\alpha_1,\,\alpha_2} = -\frac{1}{\tau}\left(\Lambda_{\alpha_1,\,\alpha_2} - \Lambda^0\right) + \Lambda^0_{\alpha_1,\,\alpha_2}\gamma\eta_{\alpha_1,\,\alpha_2}(t) \tag{8.4}$$

with the second term on the right hand side describing the noise with η_{α_1,α_2} being a Gaussian white noise with $\langle\eta_{\alpha_1,\alpha_2}\rangle = 0$ and $\langle\eta_{\alpha_1,\alpha_2}\eta_{\alpha_3,\alpha_4}\rangle = \delta_{\alpha_1,\alpha_2;\alpha_3,\alpha_4}\delta(t-t')$ as well as γ modeling the noise strength. The noise term is counterbalanced by a relaxation term acting on a typical timescale τ.

The system of stochastic differential equations is solved by the Euler-Maruyama method with periodic boundary conditions.

[1] Some terms presented in [221] are omitted here because they are only relevant with regards to implementing stresses in periodic boundary conditions, but not relevant for the following discussion

Apart from the geometrical changes by vertex movement, topological transitions (fig. 8.1 c-e) are possible as well. T_1-Transitions (fig. 8.1 c) happen when the length of an edge shrinks below a threshold $l_{T_1} \approx 0$, corresponding to a change in local neighborhood. Cell division and death are essentially triggered once a threshold value A_{Di} is crossed from below, or A_{De} from above, respectively. In the case of cell death, the dying cell is taken out of the simulation and the adjacent vertices are joined together (fig. 8.1 a, d). For cell division, 2 new vertices and a new edge are inserted at random positions along the circumference of the cell following the boundary condition of dividing the cell into to cells of equal size (fig. 8.1 a, e).

Growth in the vertex model in periodic boundary conditions can be implemented by increasing the local equilibrium value of A, A_0, to a large value, simultaneously scale the whole vertex model by the same area and equilibrate the system afterwards.

8.2 Coupling to Cellular Automata

To couple the dynamics of the vertex model to a patterning dynamics via local signaling, a cellular automaton can be run on the cells defined by the vertex model. Since the vertex model is defined in continuous time and a cellular automaton in discrete time, a timescale δt needs to be defined that specifies the amount of continous time between two cellular automaton updates. Taken together, the state s of cell α is changed at time $t + \delta t$ by the rule f, which depends on the states of the nearest neighbors.

$$s_\alpha\left(t+\delta t\right) = f\left(s_\beta,\ \beta \in \mathcal{N}_\alpha\right) \tag{8.5}$$

Since the cell state corresponds to the gene expression of the individual cells which can regulate their mechanical properties and growth, the state of the cell could also couple to the dynamics of the vertex model by changing the corresponding parameters. This can be modelled by using first order differential equations to smooth out the abrupt changes in cell state,

$$\dot{\Lambda}_{\alpha_1,\,\alpha_2} = -\frac{1}{\tau}\left(\Lambda_{\alpha_1,\,\alpha_2} - \Lambda^0_{\alpha_1,\,\alpha_2}\left(s_{\alpha_1}, s_{\alpha_2}\right)\right) + \Lambda^0_{\alpha_1,\,\alpha_2}\left(s_{\alpha_1}, s_{\alpha_2}\right)\gamma\eta_{\alpha_1,\,\alpha_2}\left(t\right) \tag{8.6}$$

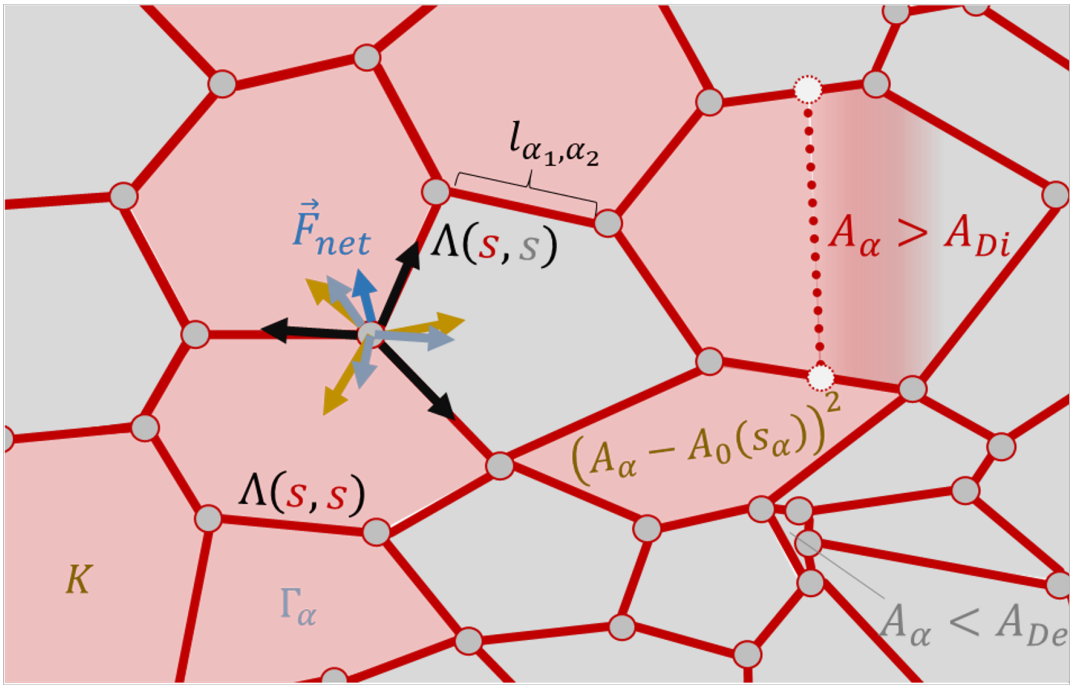
$$\Gamma_\alpha\left(s_\alpha\right) = h\left(s_\alpha\right) \tag{8.7}$$

$$\dot{A}_0\left(s_\alpha\right) = g\left(s_\alpha\right) \tag{8.8}$$

Consequently, the energy function then changes to include these additional time dependencies

$$E\left(\left\{\vec{R}_i\right\}\right) = \sum_\alpha \frac{K_\alpha}{2}\left(A_\alpha - A_0\left(t\right)\right)^2 + \sum_{\alpha_1,\alpha_2 \text{ adjacent}} \Lambda_{\alpha_1,\alpha_2}\left(t\right) l_{\alpha_1,\alpha_2} + \sum_\alpha \frac{\Gamma_\alpha\left(t\right)}{2} L_\alpha^2. \tag{8.9}$$

a



b **Dynamics of cell geometries:**

Vertex Positions

area elasticity

line tension

cortical tension

$$E(\{\vec{R}_i\}) = \sum_\alpha \frac{K_\alpha}{2}(A_\alpha - A_0(t))^2 + \sum_{\alpha_1,\alpha_2\ adjacent} \Lambda_{\alpha_1\alpha_2}(t) l_{\alpha_1,\alpha_2} + \sum_\alpha \frac{1}{2}\Gamma_\alpha(t) L_\alpha^2$$

$$\dot{A}_0(s_\alpha) = g(s_\alpha)$$

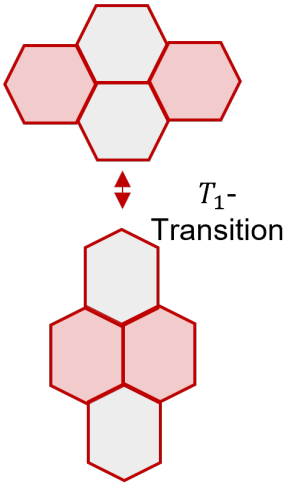
$$\dot{\Gamma}_\alpha(s_\alpha) = h(s_\alpha)$$

$$\dot{\Lambda}_{\alpha_1\alpha_2} = -\frac{1}{\tau}\left(\Lambda_{\alpha_1\alpha_2} - \Lambda^0_{\alpha_1\alpha_2}(s_{\alpha_1}, s_{\alpha_2})\right) + \Lambda^0_{\alpha_1\alpha_2}(s_{\alpha_1}, s_{\alpha_2})\gamma\eta_{\alpha_1\alpha_2}(t)$$

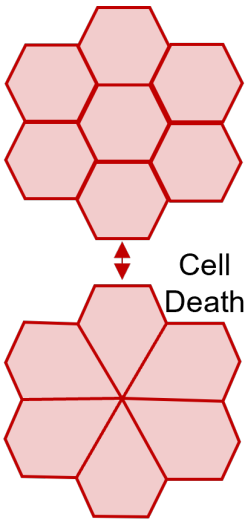
$$\overrightarrow{F_{net,i}} = \frac{\partial E(\{\vec{R}_j\})}{\partial \vec{R}_i}$$

Dynamics of cell states: $s_\alpha(t+\delta t) = f(s_\beta, \beta \in \mathcal{N}_\alpha)$

c



d



e

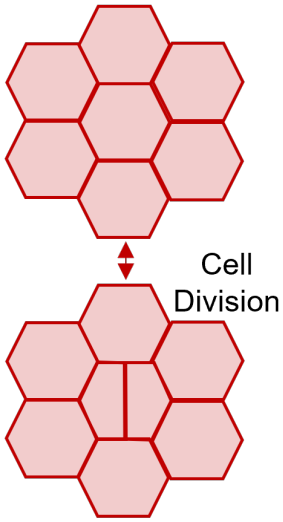


Figure 8.1 Modified Vertex Model. (Previous page) (a) & (b) Vertex model. Cells are represented as polygons, whose (shared) vertices follow an overdamped Newtonian dynamics. The force on each vertex consists of term from 3 qualitatively different origins: Area terms, depending on the area, representing the internal pressure of the cell; line tension terms, modeling the adherence between cells, depending on the length of edges between vertices; cortical tension terms, depending on the cellular circumferences, representing the forces due to the actomyosin ring on the cell surface. Adding up all the forces excerted on a vertex by the adjacent cells results in the net force determining the movement of a vertex. Technically, the forces are specified by defining an energy (b) and a force is obtained by differentiating this expression with respect to the vertex position. The parameters of the energy expression follow their own dynamics, only depending on the state of cells and, in case of the line tension proportionality parameter, a noise term. The dynamics of the cell states follows a typical discrete cellular automata equation. (c)-(e) The vertex model also allows for topological transitions, such as a change of local neighborhood positions (T_1-transition, (c)), cell death (d) and cell division (e). The processes are triggered once the length of an edge is smaller than a certain threshold (T_1 -transition) or the area of a cell is smaller(death)/larger(division) than a threshold, respectively.

8.3 Patterns in Growing Systems

As a first example, it was tried to construct a CA which qualitatively reproduces the “Salt and Pepper” like patterns of Delta-Notch inhibitory signaling. To that end, a CA with 2 states is chosen. Each cell receives as input the sum of the states of the other cells weighted by their contact length to account for the respective signaling strength, compare [62], and their own state. A cell takes state 1 if the total signal is low, and state 0 if the total signal received is high, with a hysteresis to avoid rapid cell state switches (fig. 8.2 (e) and [116]). The switching points are fixed to be apart by $\frac{1}{3}$ of the total signal required for the switch from 0 to 1 termed l_{low} (fig. 8.2 (e)). The latter is thus the only novel parameter left in the CA framework, together with the number of Euler steps in relation to the number of CA steps taken. Figure 8.2 (a) and (b) qualitatively show the dependence of the pattern on l_{low} for a system growing from 36 to 72 cells. For small values of l_{low}, all cells are in state 0, and for high values of l_{low} all cells are in state 1. For intermediate values around $l_{low} = 0.62$, a “Salt and Pepper” pattern appears. This is interesting especially in the context that in similar models for fixed, regular grids, only flickering of the whole grid appears and noise in the CA signal is needed to break the symmetry [116]. Keeping l_{low} constant at 0.62 and changing the relative amounts of CA update to Euler steps and topological transitions (fig. 8.2 (e), (d), (f)) qualitatively shows no difference in the pattern.

8.4 Future Avenues of Research

In the previous section it was already pointed out that the influence of a dynamical interaction topology and growth lead to effects that in static, regular systems could only be achieved through noise. Other potentially interesting phenomena could be investigated within this framework. In the example above a very specific CA was used to model a the specific phenomenon of “Salt and Pepper” patterns through mutual inhibition. Given a specific type of pattern, a CA that can produce such a pattern can be searched in principle in a similar way as presented in chapter 4 using evolutionary algorithms. The range of interaction can be increased to model protrusions [223] or to include diffusive long range signaling [116, 205]. Also, the coupling of chemical signaling (cell state) and mechanical properties can also be reversed as detailed above as the mechanical properties of a cell usually depend on the cell state [224]. Another interesting avenue might be the coupling of cell division on the chemical signals which is also a common theme in development [225].

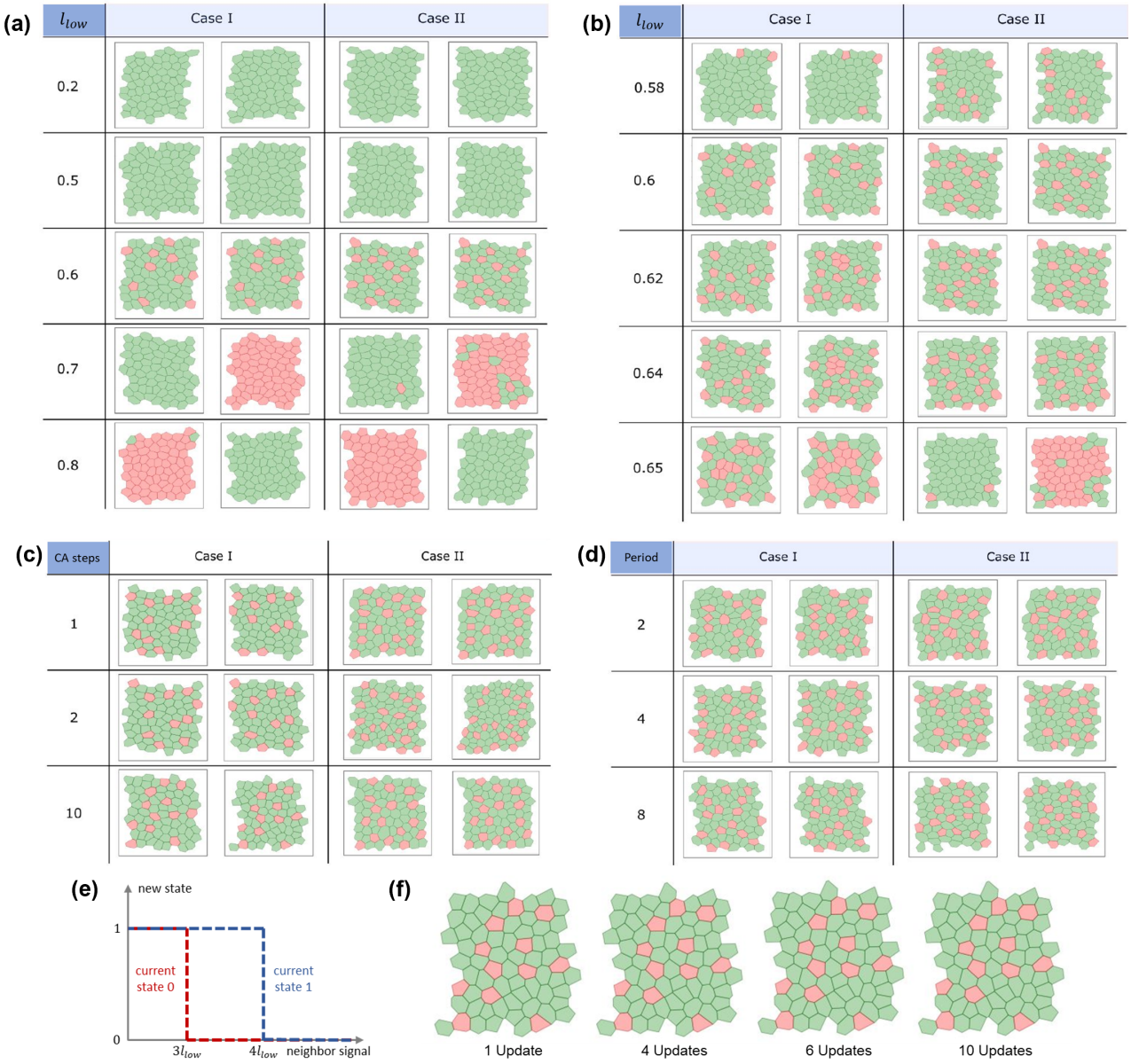


Figure 8.2 Mutual Inhibition Example. CA with rule as shown in (e) on a grid grown from 36 to 72 cells from an initially all 0 state for a vertex model with paramenters of case I ($\bar{\lambda} = 0.12$, $\bar{\gamma}$=0.04) and case II ($\bar{\lambda} = 0$, $\bar{\gamma}$=0.1) and a relative noise magnitude $z = 0.8$ in the nomenclature of [220, 219]. The system is shown before (left column in each case in (a)-(d)) and after (right) the last relaxation loop. (a) and (b): The CA parameter l_{low} was varied while there was one CA update per growth step (> 130 Euler steps, depending on the number of T1 transitions, and increase of $A_0 \to 1.4A_0$ of a single cell. (c) For $l_{low} = 0.62$, instead of before 1 now 1, 2, or 10 CA steps are carried out consecutively. An exemplary view after 1, 4, 6, and 10 updates for 10 CA steps is depicted in (f). (d) Amount of times a CA step is performed in between of topological rearrangements. After 50 initial Euler steps, 8 times a loop of cell division, T2 and T1 transitions with corresponding Euler relaxation steps in between is performed. On every 2nd, 4th or 8th loop (period), a CA Update is performed. In all of the cases from (c), (d), (f) there is no qualitative difference between the resulting patterns. Figure composition from [219].

9 How to Create Models of Locally Interacting Systems Directly from Data? - Infering Cellular Automata Rules from Biological Data - a Case Study for a 1D Model for the Small Intestine Tissue Renewal

The content of this chapter was developed in a master thesis by Max Sina Knicker [226], which was supervised by Mareike Bojer and the author of this PhD Thesis, responsible for concept development, all supervised by Ulrich Gerland.

9.1 Biologcial Data and Verbal Model for Small Intestine Ephithelial Dynamics

The small intestine epithelial tissue renewal process in mammals is the fastest and one of the most paradigmatic examples of self-renewing tissues [203]. The small intestine consists of villi protruding into the lumen of the gut as well as crypts dropping into the underlying connective tissue (fig. 9.1 (a)). Both villi and crypts are lined with a single layer of epithelial cells, and those cells can move on the underlying connective tissue. The main types[1] of cells found in this lining of the gut are stem cells, which can differentiate in all cell types found in the epithelium [228], absortive cells (enterocytes) and secretive cells (goblet cells, paneth cells and enteroendocrine cells), which may have different sub-types [203, 229]. The stem cells reside in the base of the crypt together with the paneth cells, while the enterocytes, goblet and enteroendocrine cells slide in a conveyor belt like fashion towards the tip of the villus where they finally undergo apoptosis [203]. The division of the stem cells and the following potential differentiation of the daughter cells in the crypt creates the cells that "feed" into the other end of the conveyor belt. Directly above the base of the crypt is the so called rapid amplifying zone with already fate committed precursor cells (transient amplifying cells) which divide to increase the number of cells of a certain type resulting from stem cell division [203, 229].

The signaling responsible for inducing proliferation and differentiation is based on a Wnt gradient in the crypt lumen as well as Notch and Eph/Ephrin signaling between adjacent epithelial cells [230, 231]. Notch signaling inhibits differentiation towards the secretory lineage, directing multipotent progenitors towards the absorptive lineage [229]. Secretory cells in turn express Notch ligands, leading to a lateral inhibition via direct contact interactions and thus surrounding of secretory cells with absorptive cells [229]. Further, Notch regulates the differentiation of stem cells keeping a stem cell pool [230]. Wnt proteins are screted by paneth and mesenchymal cells, forming a gradient in the crypt with largest values towards the base, and are essential for stemness of the proliferating stem cells in the crypt [231]. Eph/Ephrin signaling is present mediating cell migration and crosstalk between epithelial cell by this pathway control Wnt signaling [231].

[1] Also other cells types including tuft cells, m-cells and enterochromafin cells are found on the villus [227]

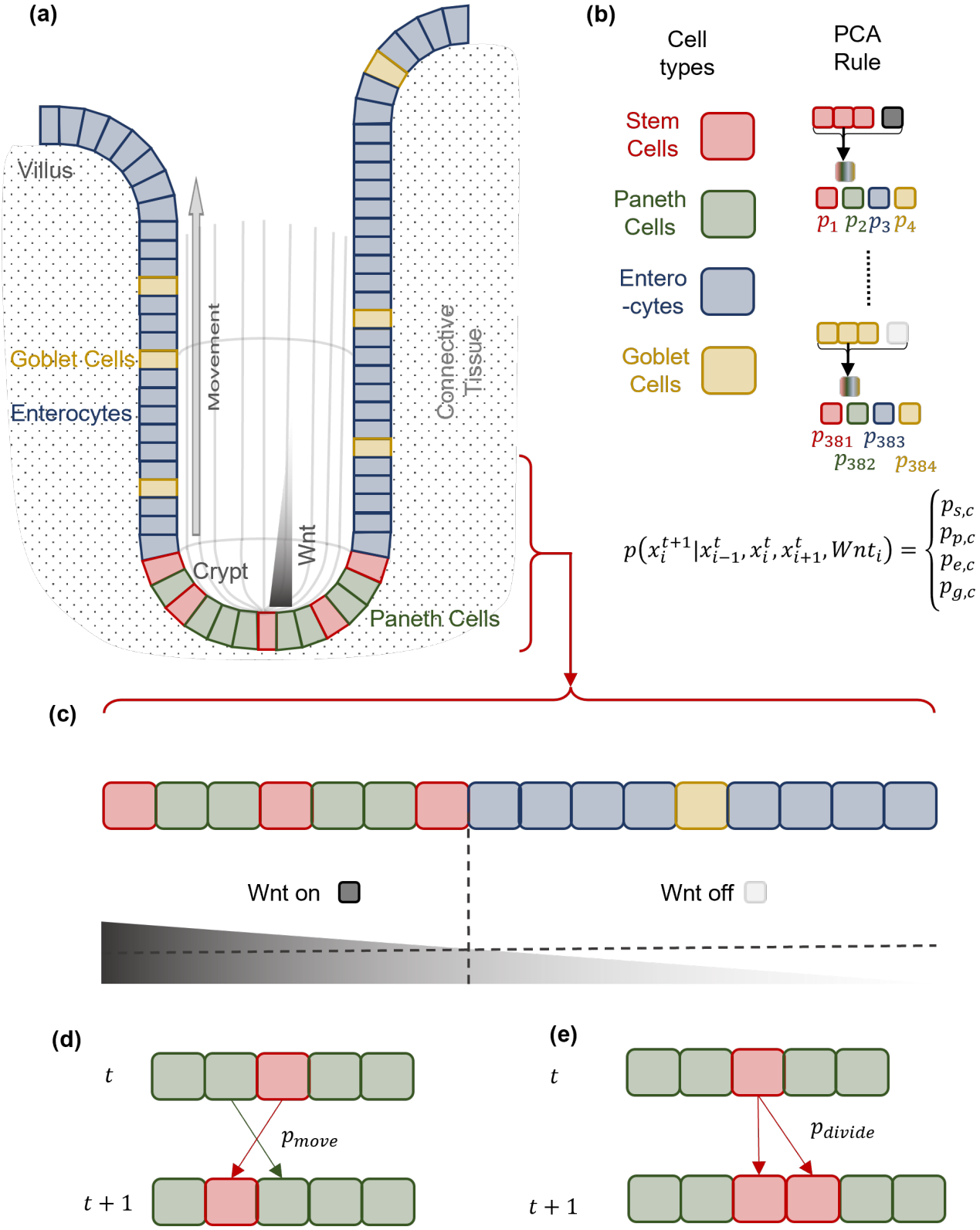

(a)
Villus
Goblet Cells
Enterocytes
Movement
Crypt
Wnt
Connective Tissue
Paneth Cells
(b)
Cell types
PCA Rule
Stem Cells
Paneth Cells
Entero -cytes
Goblet Cells
p_1 p_2 p_3 p_4
p_{381} p_{383}
p_{382} p_{384}
$p(x_i^{t+1}|x_{i-1}^t, x_i^t, x_{i+1}^t, Wnt_i) = \begin{cases} p_{s,c} \\ p_{p,c} \\ p_{e,c} \\ p_{g,c} \end{cases}$
(c)
Wnt on
Wnt off
(d)
t
p_{move}
$t+1$
(e)
t
p_{divide}
$t+1$

Figure 9.1 Small Intestine Model (previous page) (a) Schematic cross section for the small intestine crypt. Crypts are parts of the small intestine epithelium reaching into the underlying connective tissue. The epithelium consists manly of stem cells (red) and paneth cells (green) in the very bottom of the crypt and goblet cells (yellow) and enterocytes (blue) in the upper part. The cells move in a conveyor belt like motion form the bottom upwards, with the stem cells producing new cells by cell division. In the Lumen, additionally a Wnt gradient is present (b) The dynamics of the cell types is modelled by an probabilistic CA with 4 cell types - stem cells, paneth cells, enterocytes and goblet cells. Each input for the probabilistic CA updating cell i at time t consists of the state of the cell x_i^t itself, its left and right neighbors x_{i-1}^t, x_{i+1}^t, and a binary variable for the external Wnt signal. This input gets mapped to each of the 4 possible cell types with a certain probability $p\left(x_{i+1}^t | x_{i-1}^t, x_i^t, x_{i+1}^t, Wnt\left(i,t\right)\right)$. (c) Half of the cross section of the crypt is modeled as a 1D grid of cells with size 25 and the cell types as described in (b). The Wnt gradient is discretized to be "on" for the first 5 cells from the left and "off" for the rest of the cells. The dynamics is determined by the probabilistic CA described in (b). (d, e) Additionally to the standard probabilistic CA, paneth and stem cells can exchange positions with a probability p_{move}, and stem cells can divide with a probability p_{divide}.

9.2 Constructing a Quantitative Model for Small Intestine Ephithelial Dynamics

The small intestine ephithelium has been the target of serveral modeling approaches [38], for example quantitative 3D Models have been developed by Buske et al. [232] and Pin et al. [233]. Here, we strive for a simplifcation of these two models in the context of cellular automata. As modelling framework 1D probabilistic CA are picked (fig. 9.1 (b, c)). The automaton consists of a 1D grid of size $L = 25$, modelling solely the crypt. To simplify the dynamics of different cell types, only 4 cell types are considered: stem cells, enterocytes, paneth cells and goblet cells. The CA rule takes as inputs the state of the cell itself and its two neighbors, as well as a value for the external Wnt gradient, assumed to be 1 if $i \leq 4$ and 0 otherwise. Since it is a probabilistic CA, for each possible combination of inputs a probability for each possible output state is assigned, yielding 384 parameters[2] to be determined for the model. Further, introducing the possibility of an exchange of positions between stem and paneth cells with probability p_{move} at each step, ensures that the paneth cells can reside in the crypt (fig. 9.1 (d)), and stem cell proliferation, which happens with probability p_{divide} at each time step (fig. 9.1 (e)). Upon stem cell proliferation, the cell with the largest index leaves the simulated system, corresponding to leaving the crypt and moving now along the villus. Since along the villus no more differentiation dynamics happens this omits unnecessary simulation overhead. As boundary conditions, a stem cell at $i = -1$ and an enterocyte at $i = 25$ are chosen.

Choosing the above modeling framework implies some key assumptions about the biological system as well as introduces some strong simplifications:

- The one dimensional representation of an effectively 2 dimensional epithelial sheet is chosen to reflect the cylindrical symmetry of the crypt.
- In order to simplify as much as possible for an initial approach, the number of cell types is quite restricted. Most crucially, the existence of intermediate cell types prominent in the transient amplifying zone and their capability of proliferation are omitted, as well as secretory cell types found more seldom like for example enteroentrocytes. Also, quiescent stem cells acting as reserve [234] are not part of the model.
- The Wnt signal is assumed to be a completely external signal independent of simulation dynamics fixed in time, whose maintenance is not an intrinsic part of the model. This is justified by the many possible sources of Wnt signal [235] and its stable appearance in the crypt.
- The number of parameters is large, even for a biological model. This is due to the fact that the interactions between the different cell types as well as with the Wnt signal is not specified a priori apart from the interaction ranges. Introducing additional assumption, like for example assuming stem

[2] Using conservation of probability, for the 4 states of each possible input 3 parameters have to be determined. There are $4^3 = 64$ inputs due to the nearest neighbor CA, which double to 128 when taking the Wnt signaling into account

cells and paneth cells are restricted to the region $i \leq 4$ and enterocytes and goblet cells to the region $i > 4$ apart from random appearances by noise with probability η reduces the number of variables to $128 + 2$. Further assuming isotropic local interactions lets the number of free parameters drop to $84 + 2$.

To fit values for the remaining parameters, relative occurences of paneth and goblet cells as a function of cell position from Chwalinski and Potten [236] as presented in Buske et al. [232] were used. The fit was carried out by using a particle swarm optimization algorithm [237, 238] using the Canberra distance [239] as distance measure. Running 91876 fit simulations for the Wnt off regime and 128157 for the Wnt on, taking the 10 best each and combining them yields 100 high performing solutions. For more details on the fitting procedure see [226].

The results are shown in fig. 9.2. As expected, the model fits the data it was adjusted to quite well (fig. 9.2 (a)) and quickly reaches a steady state (fig. 9.2 (b)). The different cell types induced by Wnt on and Wnt off regime can clearly be seen in exemplary simulations (c). Restricting the number of variables as discussed above yields similar results. The model is also able to qualitatively and quantitatively reproduce some results not included in the fit: p_{move} and p_{divide} effectively introduce two new timescales into the system additionally to the vanilla probabilistic CA dynamics. Their ratio is of the order $10 - 25$, which is the same order of magnitude found in literature [240, 230]. Further, the percentage of single cells, pairs, and triplets of goblet cells adjacent to each other was in the same order of magnitude in the simulation (74%, 19%, 6%) as in experiments (80%, 12%, 5%) [241]. On the other hand, lateral inhibition would expect to produce at least sometimes a goblet cell in a homogeneous field of enterocytes, but the corresponding probability was close to zero in all fits. Also, seldomly occuring configurations have nearly arbitrary values (since they do not contribute to the fit effectively).

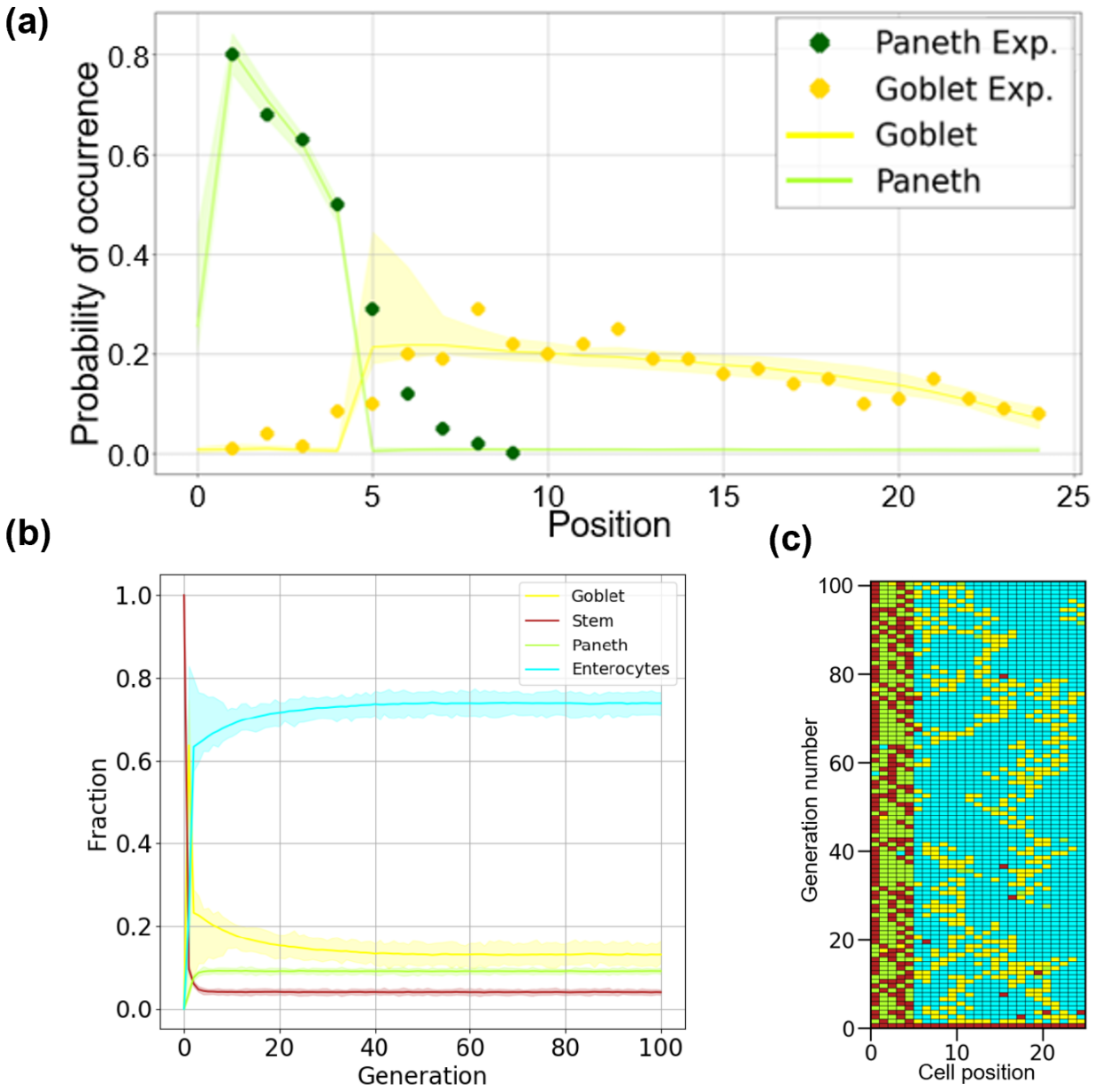


Figure 9.2 Results for the Small Intestine Crypt. Cell type distribution as a function of position for paneth and goblet cells - experimental data from Chwalinski and Potten [236] as presented in Buske et al. [232] as dark markers, mean (dark line) as well as maximum and minimum values as spanning the light colored, filled areas for a simulation for 100 fit solutions. (b) Fraction of cell types for goblet, paneth and stem cells as well as enterocytes as a function of generations for the 100 best solutions of the fitting procedure. As in (a), mean as dark line, with maximum and minimum spanning the light colored, filled areas. (c) Exemplary kymograph. Figure adapted from [226] with permission.

10 Summary and Outlook

10.1 Summary

Motivated by developmental pattern formation by direct contact interaction, in this thesis cellular automata were used as a modeling tool to investigate different possible mechanisms of pattern formation facilitated by such local interactions.

First, the main focus was finding rules that create a fixed target pattern, a French Flag, from random initial conditions. Rules using different patterning strategies were found. Rules based on (partially) erasing the pattern and recreating it showed a great robustness to changes in the initial conditions or the number of different states in the final pattern. Rules based on sorting the initial distribution of states were able to withstand perturbations due to noise, asynchronous updates and growth of the system, as well as being tunable in nearly continuous fashion with respect the width of the stripes of the flag.

As a next step, the objective was changed from finding a rule that creates a single target pattern to find rules that generate any desired target pattern from any initial pattern given inputs from a small number of organizer cells which follow an appropriate protocol of cell state changes. Through exhaustive numerical investigations a limited number of rules was identified that allow for this type of programmable pattern formation, but in an qualitatively different manner. A sufficient information-conservation principle was identified which is employed by all the numerically found rules, and is independent of the number of states and the size of the system.

While during the majority of this thesis the CA updates were synchronous, an asynchronous version with stochastic changes of update times, communication delay and a correction possibility was introduced. It was shown that for intermediate connection strengths the local interactions are able to keep the synchrony of the system for a long time, for different topologies, system sizes, correction algorithms, growth, simultaneous pattern formation, and types of update noise processes.

Coupling CA with vertex models allows studying pattern formation with local signaling in the presence of realistic mechanical movement, growth, and death of cells. In a preliminary exploration it was shown that “Salt and Pepper” patterns can be generated in a growing 2D system based on vertex models.

Supplementing a long range signal with nearest-neighbor signaling demonstrated the possibilities to reduce the fuzziness in boundaries in equilibrium and non-equilibrium systems, with the best performance for a coupling via a SUM logic followed by thresholding. This is independent of system size and model parameters, and the boundary position again shows a scaling with the size of the system similar to systems just based on long range signaling.

For specific biological systems, CA can also be inferred from experimental data and verbal biological model descriptions as shown for a stochastic CA for small intestine homeostasis. In that function, it can serve as a mathematical framework to check if the verbal models are able to explain the pattern formation process accurately.

10.2 Future Avenues

10.2.1 2D and 3D Extension

Most of the investigated pattern formation scenarios have been on a 1D or very restricted regular 2D topology, which has serious drawbacks. Regular topologies allow for a clear distinction of direction, for example, when buffering for errors in 2D programmable systems (section 5.5) or increasing the fitness in 2D French Flags by coupling several 1D patterning processes across a second dimension (section 4.11.4).

However, regularity is a simplification that allows for solution that neglect that in irregular topologies also interactions in “diagonal” direction might appear which may not allow for a clear separation of a horizontal or vertical direction. Similarly, the neglect of a third dimension makes the models not applicable for many potential use-cases. Thus, expanding to more dimensions and irregular topologies seems to be a key component for extending the models to more realistic cases. In the 2D case, the already or currently explored approaches of irregular grids sampled with the same statistics as measured in biological tissue (section 7.2) or the use of vertex models with a similar dynamics as measured in specific biological tissue (chapter 8) are a promising avenue worth exploring further. For the 3D case, vertex models have also been established to model the mechanical aspects [222] and could be used as model foundation comparable to the 2D case above. Another approach of modeling growing tissues with CA has been made by Basanta et al. [242], introducing cell division, death and movement and may be more suitable for a primary exploration.

10.2.2 Tissue as Network

A further path to systematically increase the complexity of the CA models is to represent the tissue as a network. On the intercellular level, cells could be modeled as the nodes of the network, while interactions could be represented as weighted edges of the network. This would allow to include short and long range interactions into the same framework. From a network science point of view, such a network would be constrained insofar that the network is embedded in real space, i.e., a spatial network [243]. “Connectomes” mapping the spatial relationship between cells exist for some cases and could be used as direct experimental data input [244]. Due to possible cell rearrangements, division and death, the network topology is dynamic, thus a temporal network is necessary [244]. If several different interaction types are of interest, multilayer networks could be used [245]. Leaning on ideas from the statistical physics of networks [171], defining a stochastic process to create networks with the above described properties could simplify their investigation. Modeling the intercellular communication via a network approach could help finding the answers to questions like for example:

- How can information be transmitted optimally through such a network and does nature do the same?
- Does the network topology facilitate the pattern formed in biological systems or does it play against them?
- Are there constraints on the possible dynamics happening on these networks?

CA represent the internal dynamics of cells with a single memory state and an information processing instruction. The introduction of an intracellular network to explicitly model gene regulation that can be combined with possibly several intercellular interaction networks could be another increase in complexity that in some cases has shown interesting results [160, 246], but needs a more thorough exploration. Because the possible state, parameter and network topology spaces are potentially huge, exploration strategies apart from exhaustive search or random sampling might be necessary. Evolutionary approaches have shown promising results reinventing strategies known from developmental biology for pattern formation of cells modeled with gene regulatory networks but in simple 1D topologies and without any coupling between the cells, only to an external gradient [247].

10.2.3 Integration of Experimental Data

The advent of large scale “-omics” datasets provides a tempting resource for the contruction of quantitative models. Information about cellular states as a function of time may be provided by genetic lineage tracing [248]: Prospective genetic lineage tracing uses the induction of a genetic tracer into a cell. Later, after cell division, differentiation, etc., a sample of the tissue is collected and analysed, revealing which cells originated from the same cellular ancestors. Retrospective genetic lineage tracing uses a similar idea, but instead of an induced genetic trace, somatic mutations are leveraged to reconstruct relationships between cells.

Spatial transcriptomics is used to generate snapshots of cell states in the spatial dimension [249]. Techniques include in situ hybridisation based tools, which visualize RNAs by hybridizing labeled probes in the original environment, with the key limitation of spectral overlap of the labels; in situ sequencing tools sequence the RNA in place, but are restricted by the size of the DNA balls needed for signal amplification; in situ capturing methods capture the RNA in situ with analysis steps performed ex situ – here capturing efficiency is the main limiting factor; in silico reconstruction methods use either reference atlases or de novo assumptions about the expression characteristic in physical space. To investigate which cells interact with each other, gene expression of known receptor ligand pairs is measured and their interaction determined, e.g., by threshold or correlation based methods [250]. Cellular signals may also be recorded with DNA-based systems, converting cellular signaling events into records on DNA strands, which can be read out later [251].

Exemplary datasets of the above kind include single cell surveys of the mouse small intestine epithelium [252], transcriptome data of *Drosophila* [253], and an atlas of cell types and lineage tracing for the freshwater planarian *Schmidtea mediterranea* [254].

How to integrate this amount of data in a CA model? Usually, a rough idea of what is happening in the system of interest already exists in biological literature and can be taken as an initial verbal model that needs to be put into a quantitative context. For example, to extract cell states from transcriptomics data, the expression of cell type markers can be used. Alternatively, if no dependency on predetermined notions of cell types is wished, or the thought of having certain cell types in the system needs to be checked, or knowledge about cell types is not available, clustering algorithms can be used to define cell states [255]. If a fixed geometry is chosen, the exact geometry can, e.g., be chosen to fit experimental data as exemplified for the small intestine crypt [232]. Having determined geometry and possible cell states, the general form of the transition rules has to be determined, i.e., what general types of interactions and their associated ranges. This can be done either from the verbal model, as in chapter 9, or by using the transcriptomics data as described above. In a next step, the exact parameters of these transition rules have to be fitted. Using a probabilistic model as in chapter 9, values can be fitted using the spatial snapshots of the spatial transcriptomics data or the temporal snapshots from lineage tracing. As fit methods for deterministic models evolutionary algorithms are an option, compare chapter 4. For probabilistic CA, similar to chapter 9, particle swarm optimization can be used, but also gradient based methods are in principle possible. The spatial transcriptomics data may also be used to generate initial conditions as well as boundary conditions.

What use does such a model have? Following Mogilner et al. [36], showing that a quantitative model can reproduce the biological phenomenon indicates that all “necessary players and interactions” [36] have been identified or, in case of not being able to reproduce the data, the underlying mechanisms may not be understood well. Furthermore, determining sensitivities of the parameters, stiff and sloppy parameter combinations can be extracted, indicating which parameter combinations need to be determined more precisely [256].

Leaving the world of CA models, one can speculate that it might even be possible to arrive at a more generic class of models that in principle can include frameworks for all relevant cell-cell interaction types and outcomes. As first examples, an approach was already undertaken to predict the state trajectory of cells by learning models for transcriptomic vector fields in the context of hematopoiesis [257], and focusing on image analysis and cell population estimation [258]. Feeding such a generic class of models with transcriptomics data, quantitative models could be produced fast, potentially even without having a verbal model first. Since in the transcriptomics data, in principle, also the expression of proteins determining mechanical interactions is recorded, these may potentially be included as well. In some sense, this whole approach would be close to having a “machine learning model” for morphogenesis, which has enough realistic and interpret-able components to adapt to different biological situations when fitted to large scale experimental data.

A Appendix

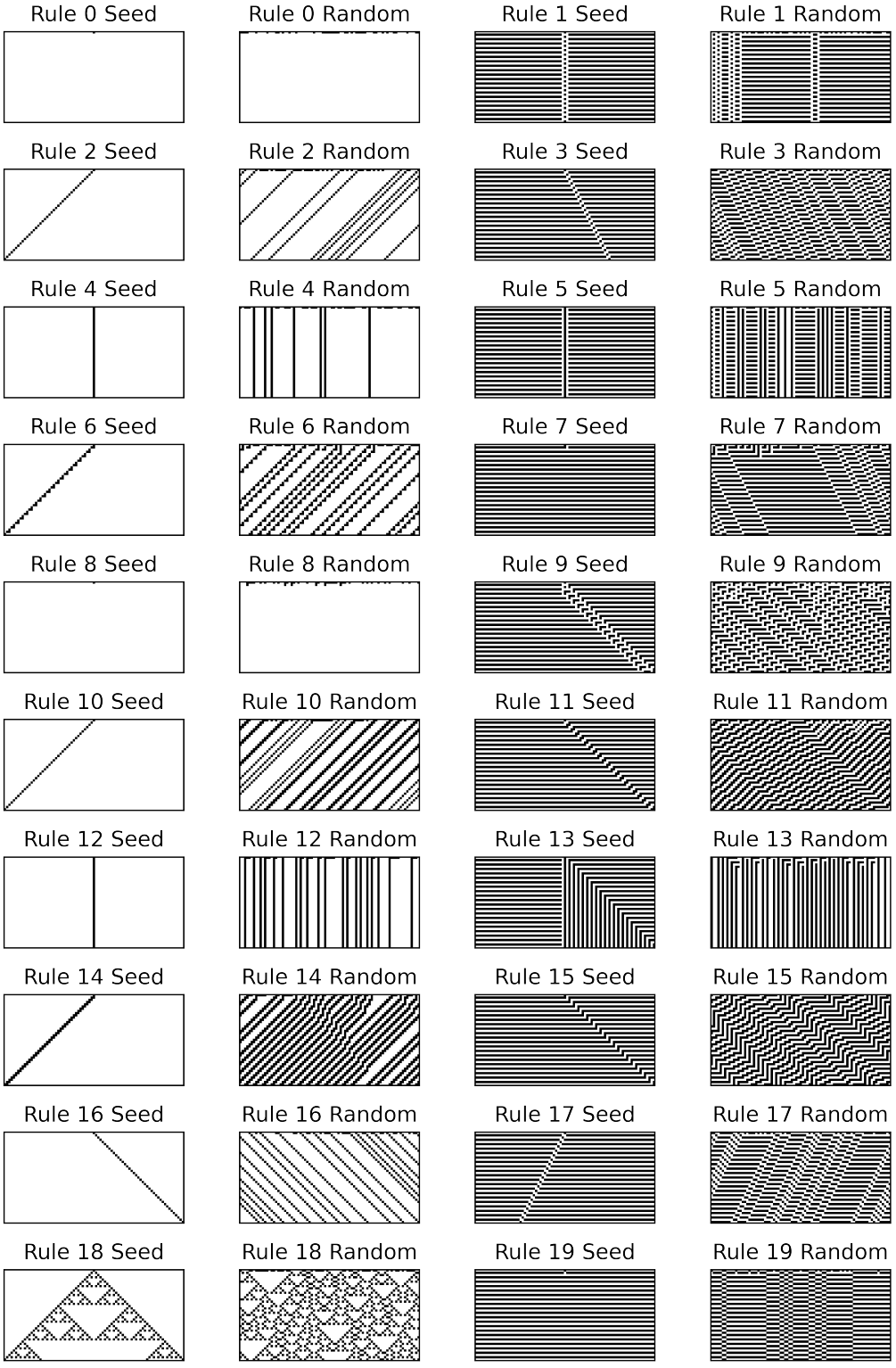
A.1 Autonomous DNA Cellular Automata

An implementation of Cellular Automata has been proposed by Yin et al. for purely DNA based systems [190]. On a track made of DNA lattice dangling DNA molecules can be mounted, engineered to have some flexibility to allow for contact with the neighbors and a sticky end allowing connection to floating molecules or their neighbors. In the solution, free floating DNA molecules again with sticky ends perform helper action or act as rule molecules. Information about states is encoded into the DNA sequence. Through binding with the neighbors and subsequent cleavage, a dangling DNA molecule can first incorporate information about the state of the neighbors in its strand. Then, it can bind to an rule molecules which leads to the assignment of a new state, with a subsequent restoration of a sticky end. This process an be engineered in a way that it propagates as wave across the lattice, leading to an effectively synchronous CA update. For more details see [190]. Organizer cells can in principle also implemented in this set-up by modifying the initiator molecules necessary to start the waves such to also encode temporal information [167].

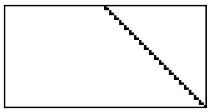
A.2 Cellular Automata

A.2.1 Simulation of Elementary Cellular Automata

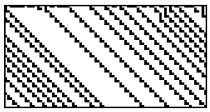
Periodic Boundary Conditions



Rule 20 Seed



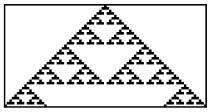
Rule 20 Random



Rule 21 Seed

Rule 21 Random

Rule 22 Seed

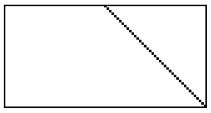


Rule 22 Random

Rule 23 Seed

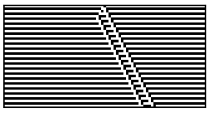
Rule 23 Random

Rule 24 Seed



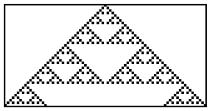
Rule 24 Random

Rule 25 Seed

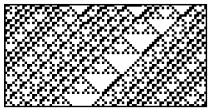


Rule 25 Random

Rule 26 Seed



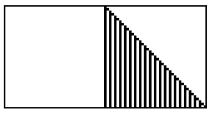
Rule 26 Random



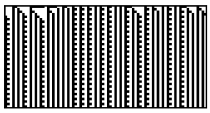
Rule 27 Seed

Rule 27 Random

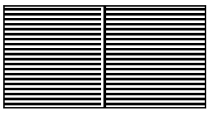
Rule 28 Seed



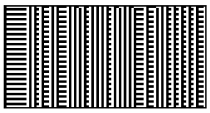
Rule 28 Random



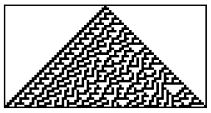
Rule 29 Seed



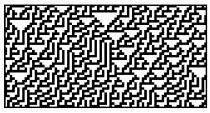
Rule 29 Random



Rule 30 Seed



Rule 30 Random

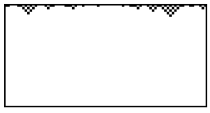


Rule 31 Seed

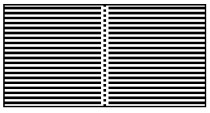
Rule 31 Random

Rule 32 Seed

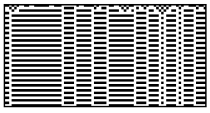
Rule 32 Random



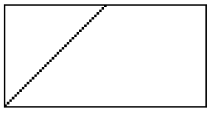
Rule 33 Seed



Rule 33 Random

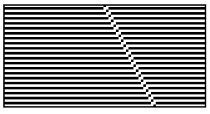


Rule 34 Seed

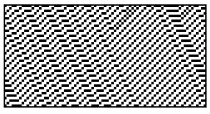


Rule 34 Random

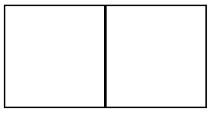
Rule 35 Seed



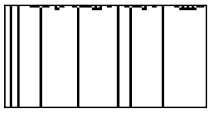
Rule 35 Random



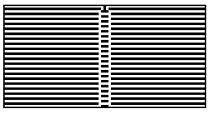
Rule 36 Seed



Rule 36 Random

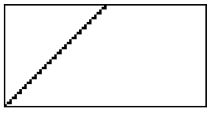


Rule 37 Seed

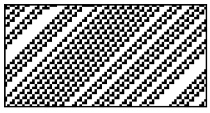


Rule 37 Random

Rule 38 Seed

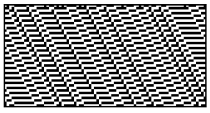


Rule 38 Random

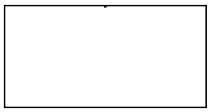


Rule 39 Seed

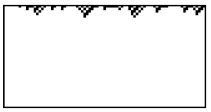
Rule 39 Random



Rule 40 Seed

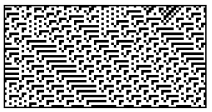


Rule 40 Random

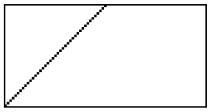


Rule 41 Seed

Rule 41 Random



Rule 42 Seed

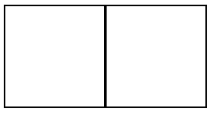


Rule 42 Random

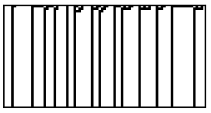
Rule 43 Seed

Rule 43 Random

Rule 44 Seed

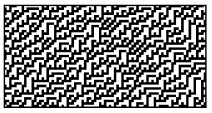


Rule 44 Random



Rule 45 Seed

Rule 45 Random

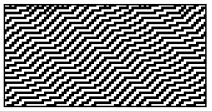


Rule 46 Seed

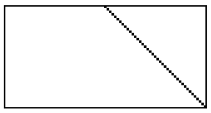
Rule 46 Random

Rule 47 Seed

Rule 47 Random

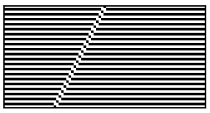


Rule 48 Seed

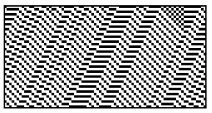


Rule 48 Random

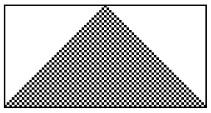
Rule 49 Seed



Rule 49 Random

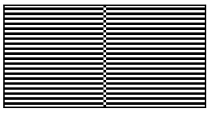


Rule 50 Seed

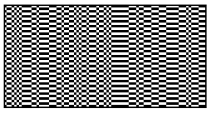


Rule 50 Random

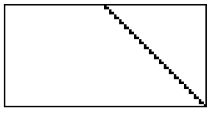
Rule 51 Seed



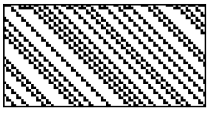
Rule 51 Random



Rule 52 Seed

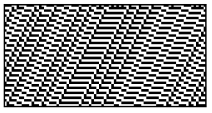


Rule 52 Random

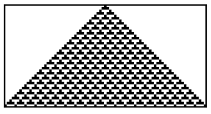


Rule 53 Seed

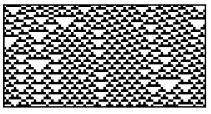
Rule 53 Random



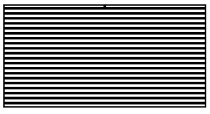
Rule 54 Seed



Rule 54 Random

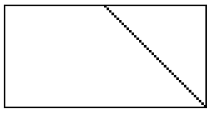


Rule 55 Seed

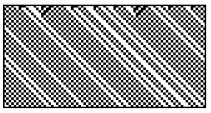


Rule 55 Random

Rule 56 Seed



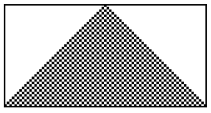
Rule 56 Random



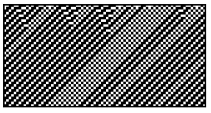
Rule 57 Seed

Rule 57 Random

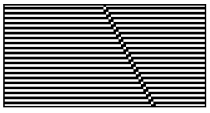
Rule 58 Seed



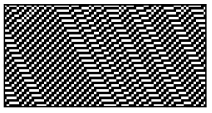
Rule 58 Random



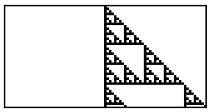
Rule 59 Seed



Rule 59 Random

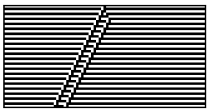


Rule 60 Seed



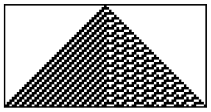
Rule 60 Random

Rule 61 Seed

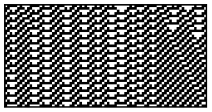


Rule 61 Random

Rule 62 Seed



Rule 62 Random

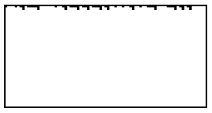


Rule 63 Seed

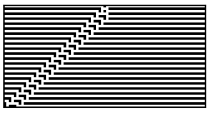
Rule 63 Random

Rule 64 Seed

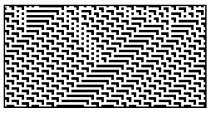
Rule 64 Random



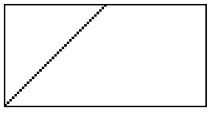
Rule 65 Seed



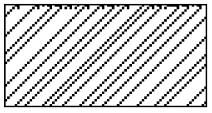
Rule 65 Random



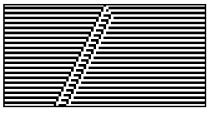
Rule 66 Seed



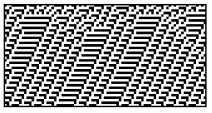
Rule 66 Random



Rule 67 Seed

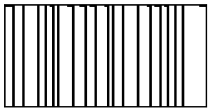


Rule 67 Random

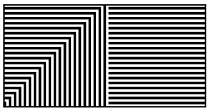


Rule 68 Seed

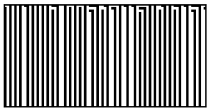
Rule 68 Random



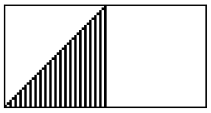
Rule 69 Seed



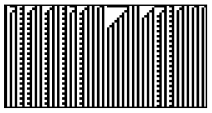
Rule 69 Random



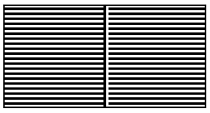
Rule 70 Seed



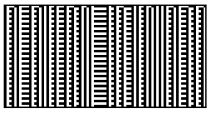
Rule 70 Random



Rule 71 Seed



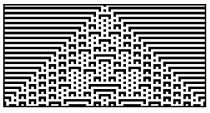
Rule 71 Random



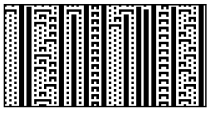
Rule 72 Seed

Rule 72 Random

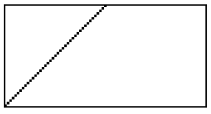
Rule 73 Seed



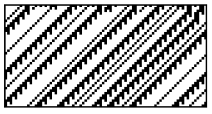
Rule 73 Random



Rule 74 Seed

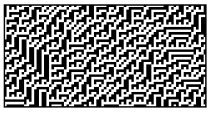


Rule 74 Random

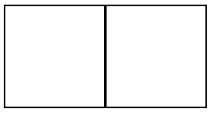


Rule 75 Seed

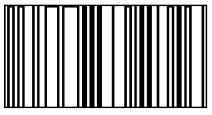
Rule 75 Random



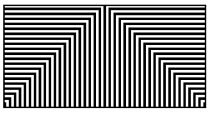
Rule 76 Seed



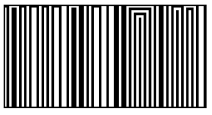
Rule 76 Random



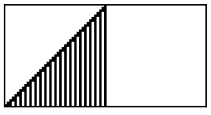
Rule 77 Seed



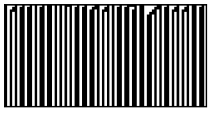
Rule 77 Random



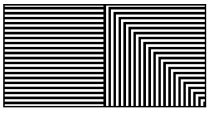
Rule 78 Seed



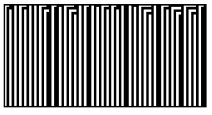
Rule 78 Random



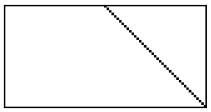
Rule 79 Seed



Rule 79 Random



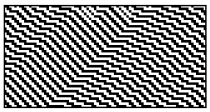
Rule 80 Seed



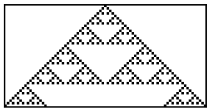
Rule 80 Random

Rule 81 Seed

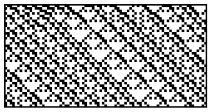
Rule 81 Random



Rule 82 Seed



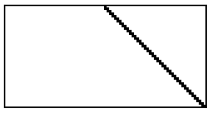
Rule 82 Random



Rule 83 Seed

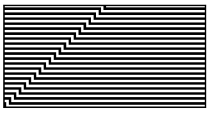
Rule 83 Random

Rule 84 Seed

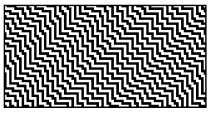


Rule 84 Random

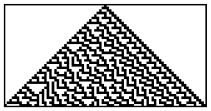
Rule 85 Seed



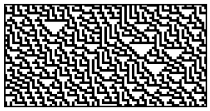
Rule 85 Random



Rule 86 Seed



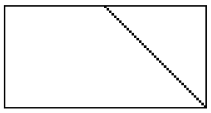
Rule 86 Random



Rule 87 Seed

Rule 87 Random

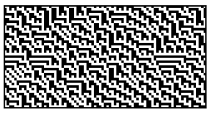
Rule 88 Seed



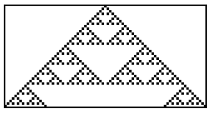
Rule 88 Random

Rule 89 Seed

Rule 89 Random

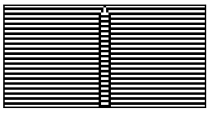


Rule 90 Seed



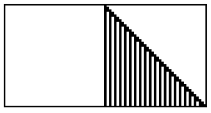
Rule 90 Random

Rule 91 Seed

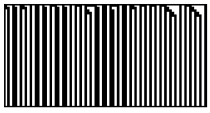


Rule 91 Random

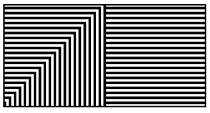
Rule 92 Seed



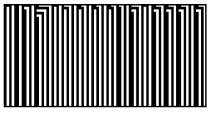
Rule 92 Random



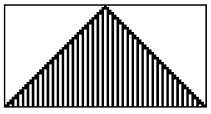
Rule 93 Seed



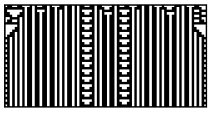
Rule 93 Random



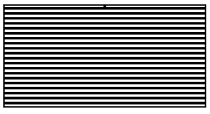
Rule 94 Seed



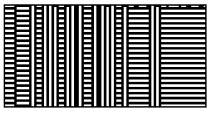
Rule 94 Random



Rule 95 Seed



Rule 95 Random

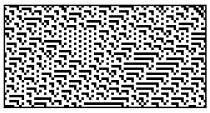


Rule 96 Seed

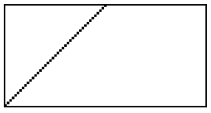
Rule 96 Random

Rule 97 Seed

Rule 97 Random



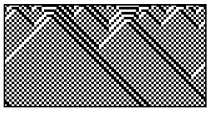
Rule 98 Seed



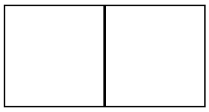
Rule 98 Random

Rule 99 Seed

Rule 99 Random



Rule 100 Seed

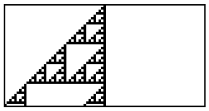


Rule 100 Random

Rule 101 Seed

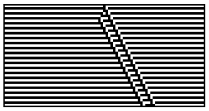
Rule 101 Random

Rule 102 Seed



Rule 102 Random

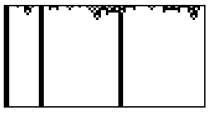
Rule 103 Seed



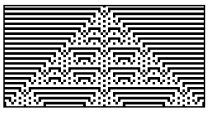
Rule 103 Random

Rule 104 Seed

Rule 104 Random

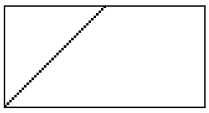


Rule 105 Seed



Rule 105 Random

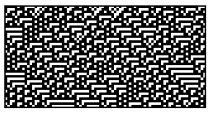
Rule 106 Seed



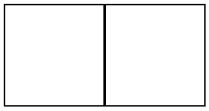
Rule 106 Random

Rule 107 Seed

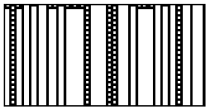
Rule 107 Random



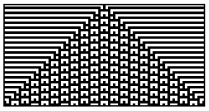
Rule 108 Seed



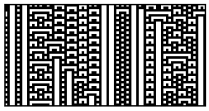
Rule 108 Random



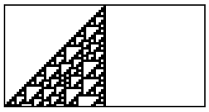
Rule 109 Seed



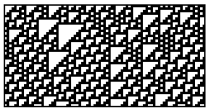
Rule 109 Random



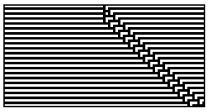
Rule 110 Seed



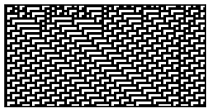
Rule 110 Random



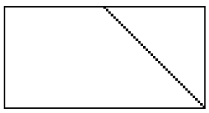
Rule 111 Seed



Rule 111 Random



Rule 112 Seed

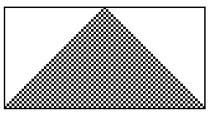


Rule 112 Random

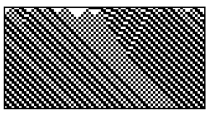
Rule 113 Seed

Rule 113 Random

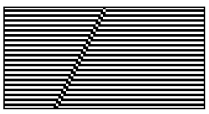
Rule 114 Seed



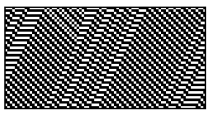
Rule 114 Random



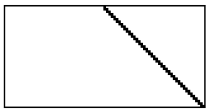
Rule 115 Seed



Rule 115 Random



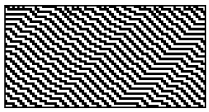
Rule 116 Seed



Rule 116 Random

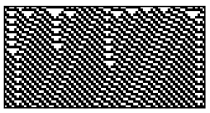
Rule 117 Seed

Rule 117 Random

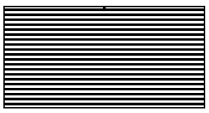


Rule 118 Seed

Rule 118 Random

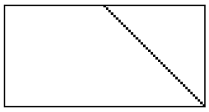


Rule 119 Seed



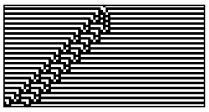
Rule 119 Random

Rule 120 Seed

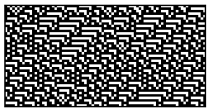


Rule 120 Random

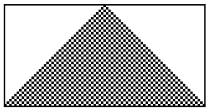
Rule 121 Seed



Rule 121 Random

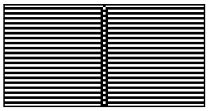


Rule 122 Seed

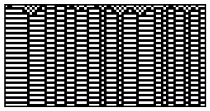


Rule 122 Random

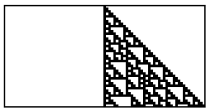
Rule 123 Seed



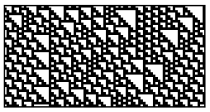
Rule 123 Random



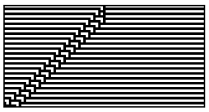
Rule 124 Seed



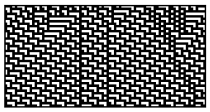
Rule 124 Random



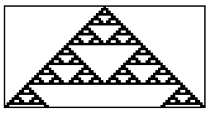
Rule 125 Seed



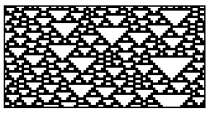
Rule 125 Random



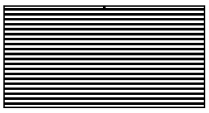
Rule 126 Seed



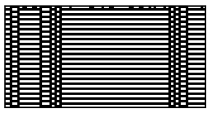
Rule 126 Random



Rule 127 Seed



Rule 127 Random

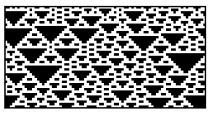


Rule 128 Seed

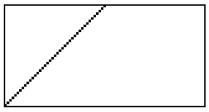
Rule 128 Random

Rule 129 Seed

Rule 129 Random

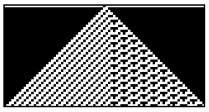


Rule 130 Seed

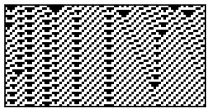


Rule 130 Random

Rule 131 Seed



Rule 131 Random

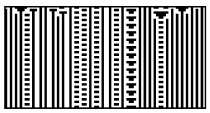


Rule 132 Seed

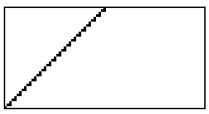
Rule 132 Random

Rule 133 Seed

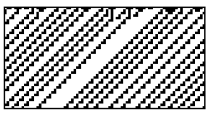
Rule 133 Random



Rule 134 Seed



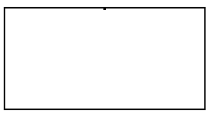
Rule 134 Random



Rule 135 Seed

Rule 135 Random

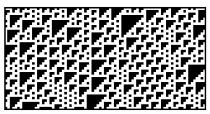
Rule 136 Seed



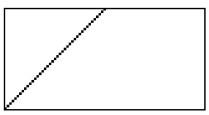
Rule 136 Random

Rule 137 Seed

Rule 137 Random



Rule 138 Seed

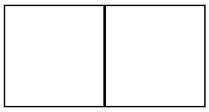


Rule 138 Random

Rule 139 Seed

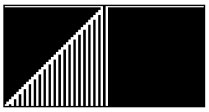
Rule 139 Random

Rule 140 Seed

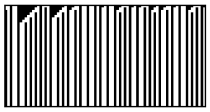


Rule 140 Random

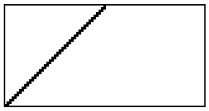
Rule 141 Seed



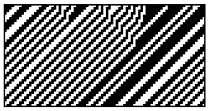
Rule 141 Random



Rule 142 Seed



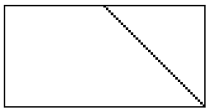
Rule 142 Random



Rule 143 Seed

Rule 143 Random

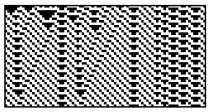
Rule 144 Seed



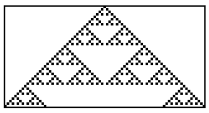
Rule 144 Random

Rule 145 Seed

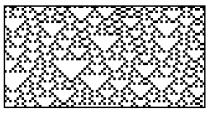
Rule 145 Random



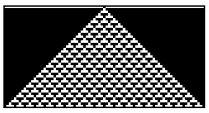
Rule 146 Seed



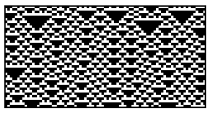
Rule 146 Random



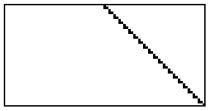
Rule 147 Seed



Rule 147 Random



Rule 148 Seed

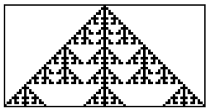


Rule 148 Random

Rule 149 Seed

Rule 149 Random

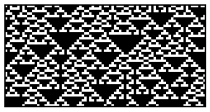
Rule 150 Seed



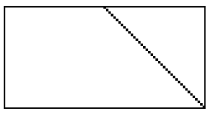
Rule 150 Random

Rule 151 Seed

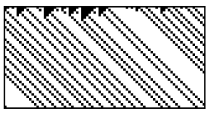
Rule 151 Random



Rule 152 Seed



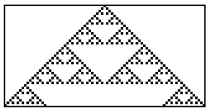
Rule 152 Random



Rule 153 Seed

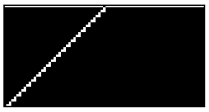
Rule 153 Random

Rule 154 Seed



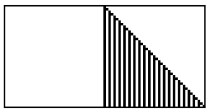
Rule 154 Random

Rule 155 Seed

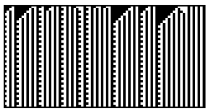


Rule 155 Random

Rule 156 Seed

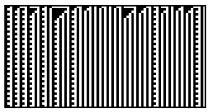


Rule 156 Random

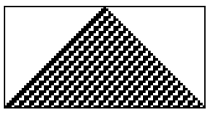


Rule 157 Seed

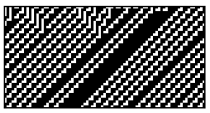
Rule 157 Random



Rule 158 Seed



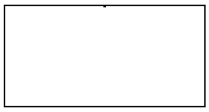
Rule 158 Random



Rule 159 Seed

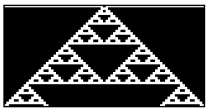
Rule 159 Random

Rule 160 Seed

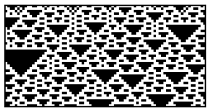


Rule 160 Random

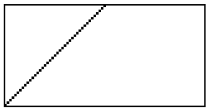
Rule 161 Seed



Rule 161 Random



Rule 162 Seed



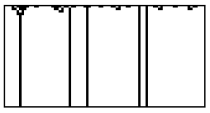
Rule 162 Random

Rule 163 Seed

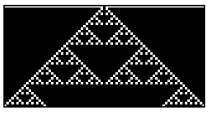
Rule 163 Random

Rule 164 Seed

Rule 164 Random

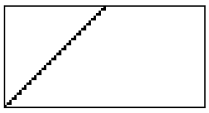


Rule 165 Seed

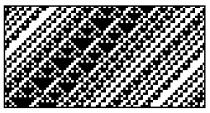


Rule 165 Random

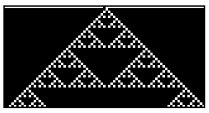
Rule 166 Seed



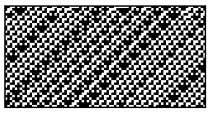
Rule 166 Random



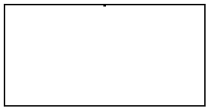
Rule 167 Seed



Rule 167 Random



Rule 168 Seed

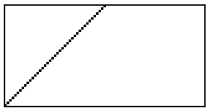


Rule 168 Random

Rule 169 Seed

Rule 169 Random

Rule 170 Seed



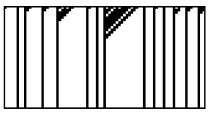
Rule 170 Random

Rule 171 Seed

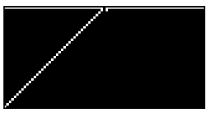
Rule 171 Random

Rule 172 Seed

Rule 172 Random

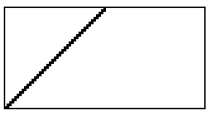


Rule 173 Seed



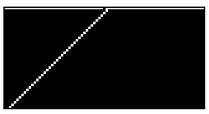
Rule 173 Random

Rule 174 Seed



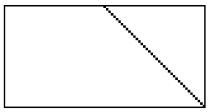
Rule 174 Random

Rule 175 Seed



Rule 175 Random

Rule 176 Seed

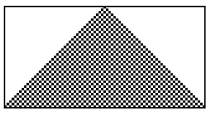


Rule 176 Random

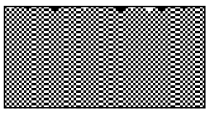
Rule 177 Seed

Rule 177 Random

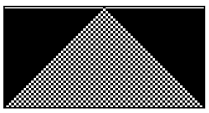
Rule 178 Seed



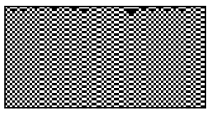
Rule 178 Random



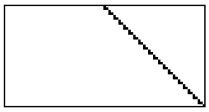
Rule 179 Seed



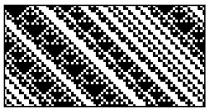
Rule 179 Random



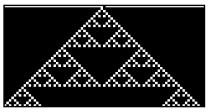
Rule 180 Seed



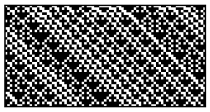
Rule 180 Random



Rule 181 Seed

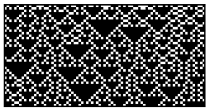


Rule 181 Random



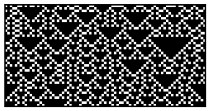
Rule 182 Seed

Rule 182 Random

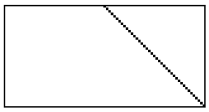


Rule 183 Seed

Rule 183 Random



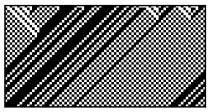
Rule 184 Seed



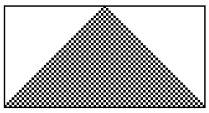
Rule 184 Random

Rule 185 Seed

Rule 185 Random



Rule 186 Seed

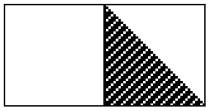


Rule 186 Random

Rule 187 Seed

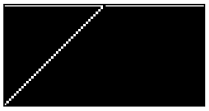
Rule 187 Random

Rule 188 Seed

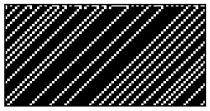


Rule 188 Random

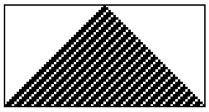
Rule 189 Seed



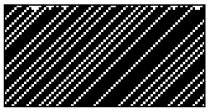
Rule 189 Random



Rule 190 Seed



Rule 190 Random



Rule 191 Seed

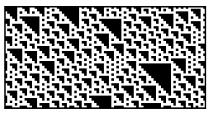
Rule 191 Random

Rule 192 Seed

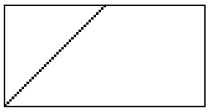
Rule 192 Random

Rule 193 Seed

Rule 193 Random

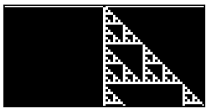


Rule 194 Seed



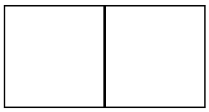
Rule 194 Random

Rule 195 Seed



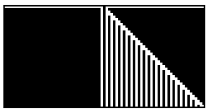
Rule 195 Random

Rule 196 Seed

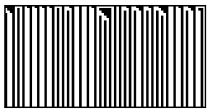


Rule 196 Random

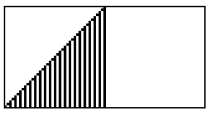
Rule 197 Seed



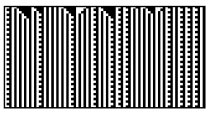
Rule 197 Random



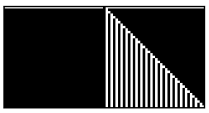
Rule 198 Seed



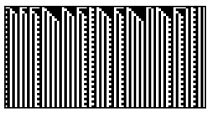
Rule 198 Random



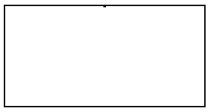
Rule 199 Seed



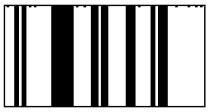
Rule 199 Random



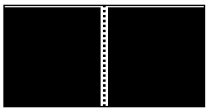
Rule 200 Seed



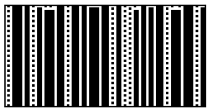
Rule 200 Random



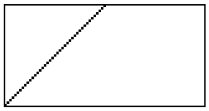
Rule 201 Seed



Rule 201 Random

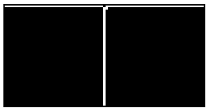


Rule 202 Seed

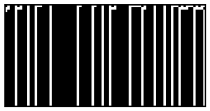


Rule 202 Random

Rule 203 Seed

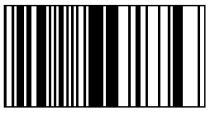


Rule 203 Random

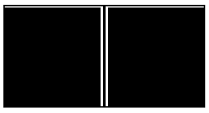


Rule 204 Seed

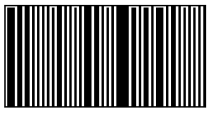
Rule 204 Random



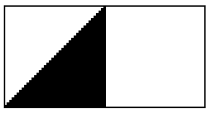
Rule 205 Seed



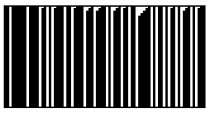
Rule 205 Random



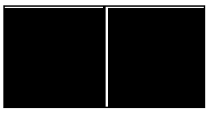
Rule 206 Seed



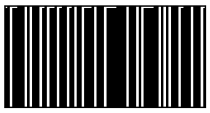
Rule 206 Random



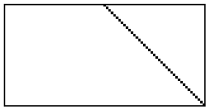
Rule 207 Seed



Rule 207 Random



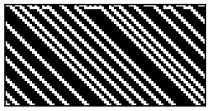
Rule 208 Seed



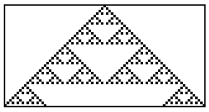
Rule 208 Random

Rule 209 Seed

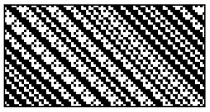
Rule 209 Random



Rule 210 Seed

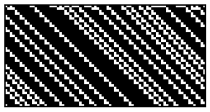


Rule 210 Random



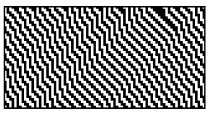
Rule 211 Seed

Rule 211 Random



Rule 212 Seed

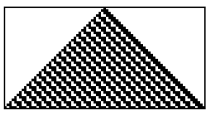
Rule 212 Random



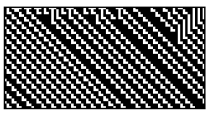
Rule 213 Seed

Rule 213 Random

Rule 214 Seed

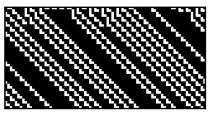


Rule 214 Random

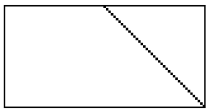


Rule 215 Seed

Rule 215 Random

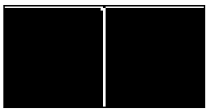


Rule 216 Seed



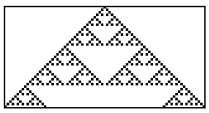
Rule 216 Random

Rule 217 Seed



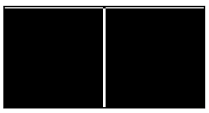
Rule 217 Random

Rule 218 Seed

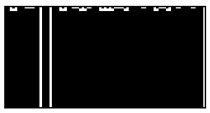


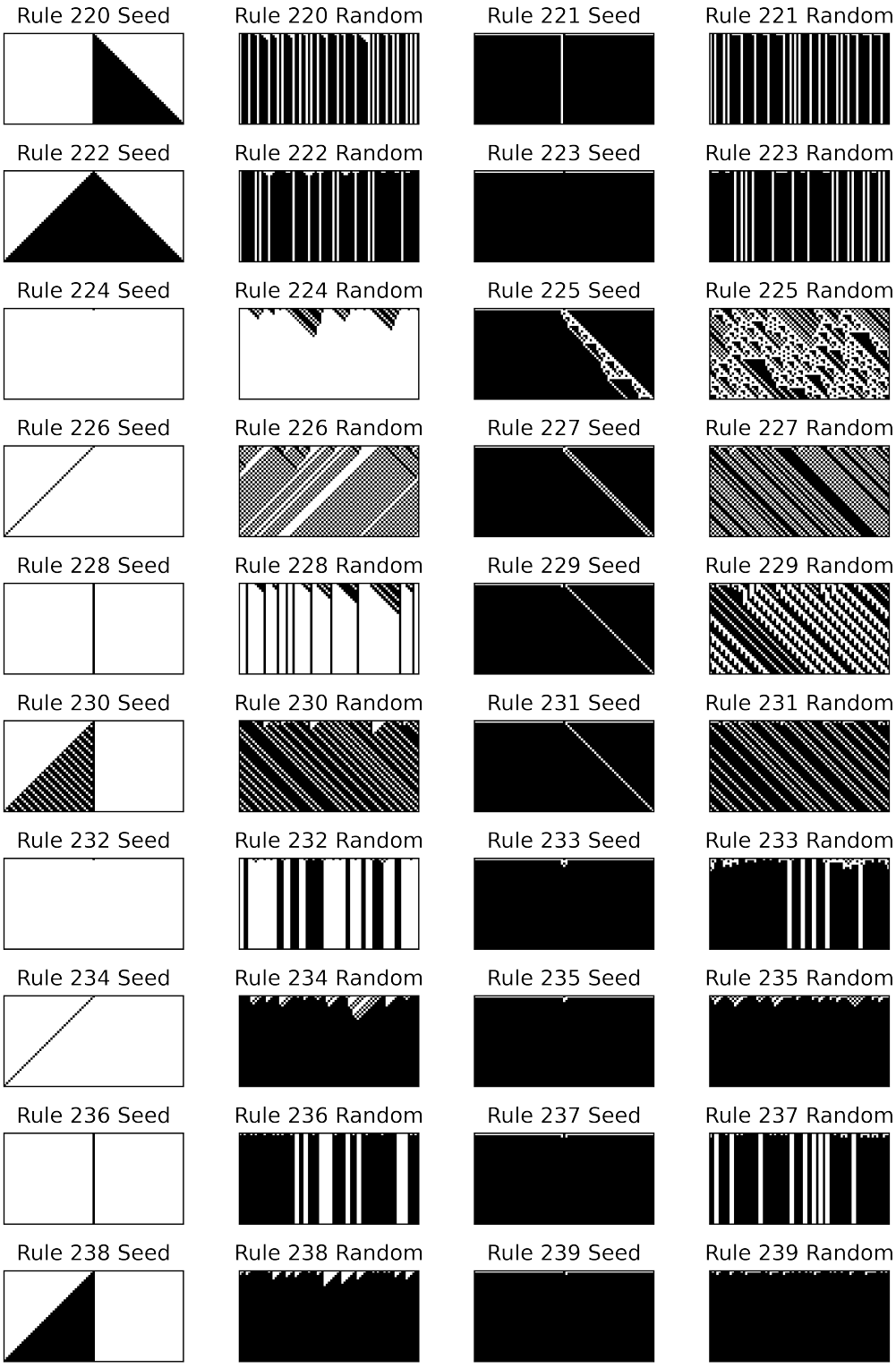
Rule 218 Random

Rule 219 Seed



Rule 219 Random




Rule 220 Seed
Rule 220 Random
Rule 221 Seed
Rule 221 Random
Rule 222 Seed
Rule 222 Random
Rule 223 Seed
Rule 223 Random
Rule 224 Seed
Rule 224 Random
Rule 225 Seed
Rule 225 Random
Rule 226 Seed
Rule 226 Random
Rule 227 Seed
Rule 227 Random
Rule 228 Seed
Rule 228 Random
Rule 229 Seed
Rule 229 Random
Rule 230 Seed
Rule 230 Random
Rule 231 Seed
Rule 231 Random
Rule 232 Seed
Rule 232 Random
Rule 233 Seed
Rule 233 Random
Rule 234 Seed
Rule 234 Random
Rule 235 Seed
Rule 235 Random
Rule 236 Seed
Rule 236 Random
Rule 237 Seed
Rule 237 Random
Rule 238 Seed
Rule 238 Random
Rule 239 Seed
Rule 239 Random

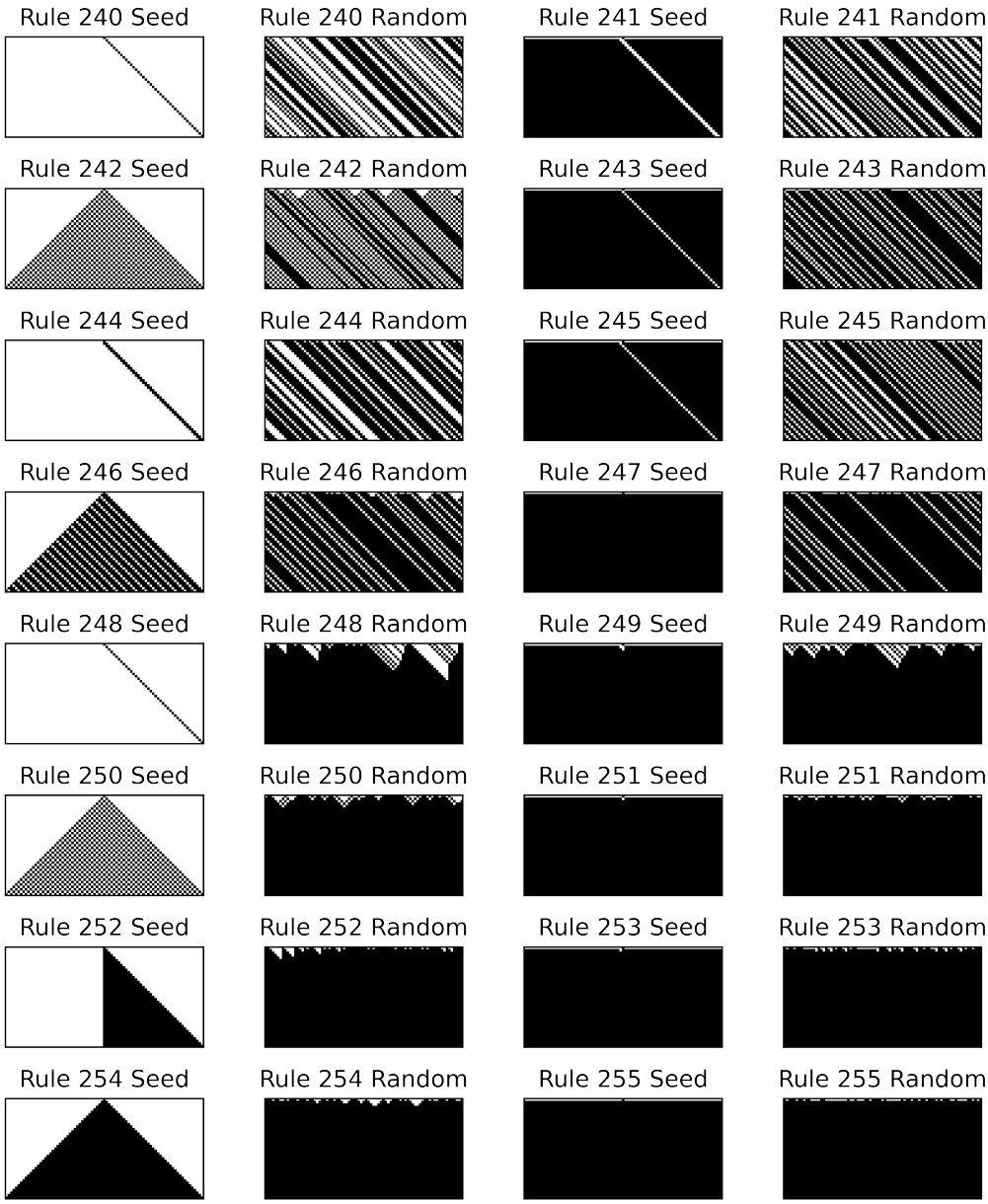
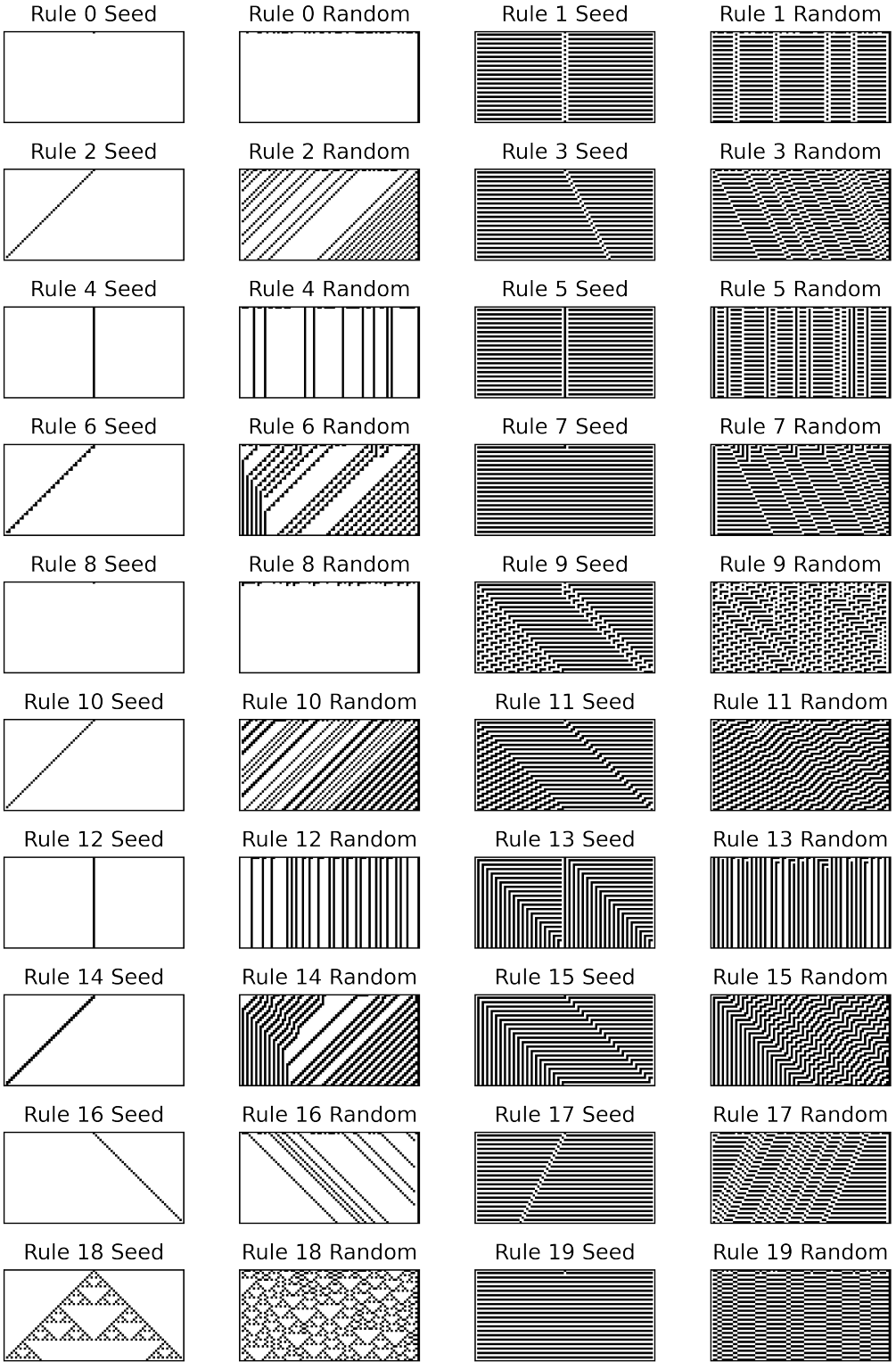
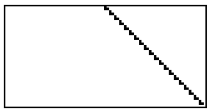


Figure A.1 Simulation of Elementary Cellular Automata Rules for Length $L = 80$ and Seed or Random Initial Conditions and Periodic Boundary Conditions.

Fixed Boundary Conditions



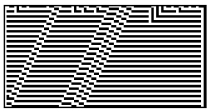
Rule 20 Seed



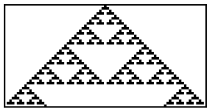
Rule 20 Random

Rule 21 Seed

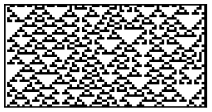
Rule 21 Random



Rule 22 Seed



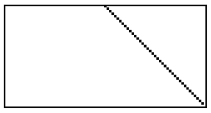
Rule 22 Random



Rule 23 Seed

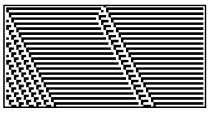
Rule 23 Random

Rule 24 Seed



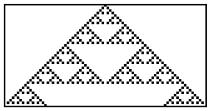
Rule 24 Random

Rule 25 Seed

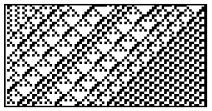


Rule 25 Random

Rule 26 Seed



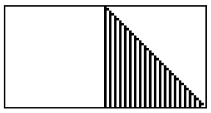
Rule 26 Random



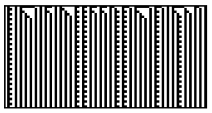
Rule 27 Seed

Rule 27 Random

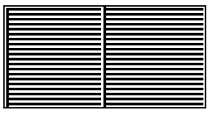
Rule 28 Seed



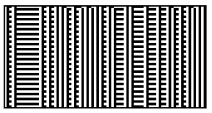
Rule 28 Random



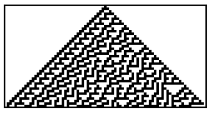
Rule 29 Seed



Rule 29 Random



Rule 30 Seed



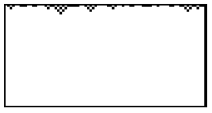
Rule 30 Random

Rule 31 Seed

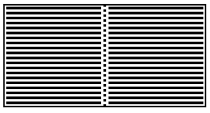
Rule 31 Random

Rule 32 Seed

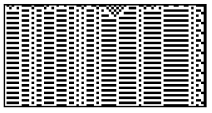
Rule 32 Random



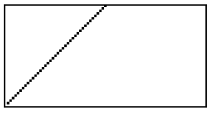
Rule 33 Seed



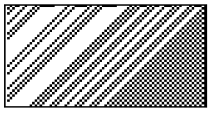
Rule 33 Random



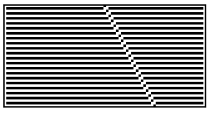
Rule 34 Seed



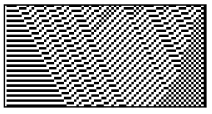
Rule 34 Random



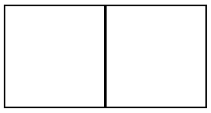
Rule 35 Seed



Rule 35 Random

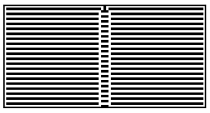


Rule 36 Seed

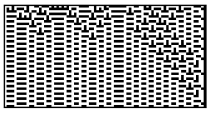


Rule 36 Random

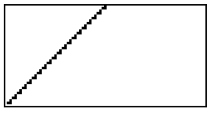
Rule 37 Seed



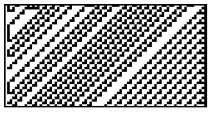
Rule 37 Random



Rule 38 Seed



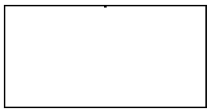
Rule 38 Random



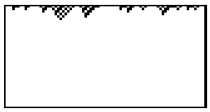
Rule 39 Seed

Rule 39 Random

Rule 40 Seed

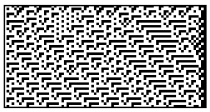


Rule 40 Random

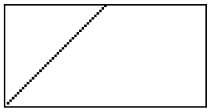


Rule 41 Seed

Rule 41 Random



Rule 42 Seed

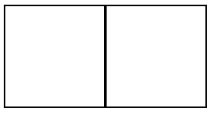


Rule 42 Random

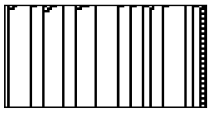
Rule 43 Seed

Rule 43 Random

Rule 44 Seed

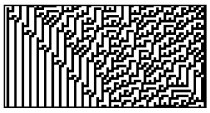


Rule 44 Random

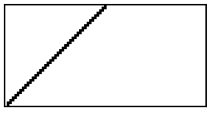


Rule 45 Seed

Rule 45 Random



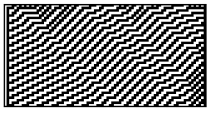
Rule 46 Seed



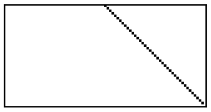
Rule 46 Random

Rule 47 Seed

Rule 47 Random



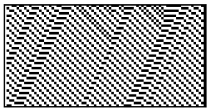
Rule 48 Seed



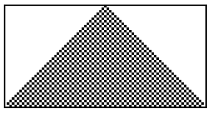
Rule 48 Random

Rule 49 Seed

Rule 49 Random

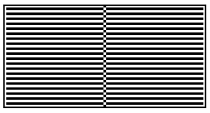


Rule 50 Seed

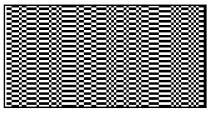


Rule 50 Random

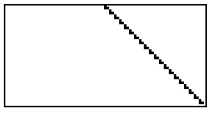
Rule 51 Seed



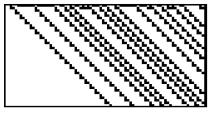
Rule 51 Random



Rule 52 Seed

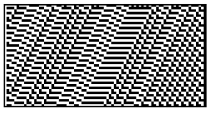


Rule 52 Random

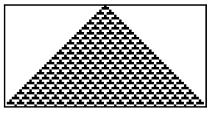


Rule 53 Seed

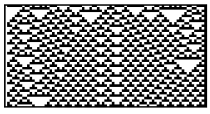
Rule 53 Random



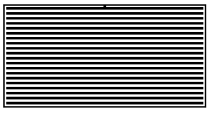
Rule 54 Seed



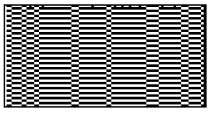
Rule 54 Random



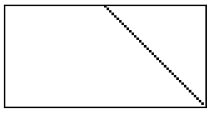
Rule 55 Seed



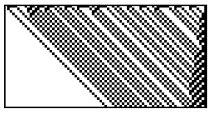
Rule 55 Random



Rule 56 Seed



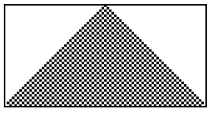
Rule 56 Random



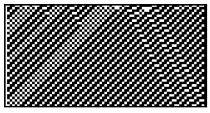
Rule 57 Seed

Rule 57 Random

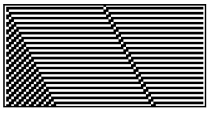
Rule 58 Seed



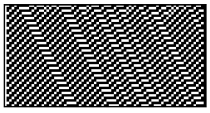
Rule 58 Random



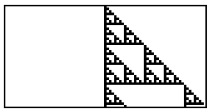
Rule 59 Seed



Rule 59 Random

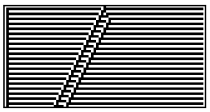


Rule 60 Seed



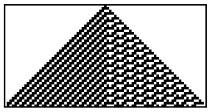
Rule 60 Random

Rule 61 Seed



Rule 61 Random

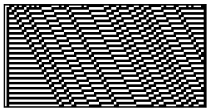
Rule 62 Seed



Rule 62 Random

Rule 63 Seed

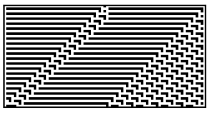
Rule 63 Random



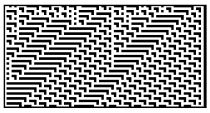
Rule 64 Seed

Rule 64 Random

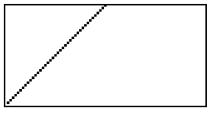
Rule 65 Seed



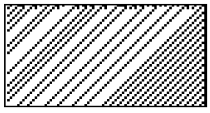
Rule 65 Random



Rule 66 Seed



Rule 66 Random



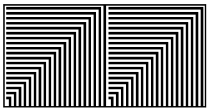
Rule 67 Seed

Rule 67 Random

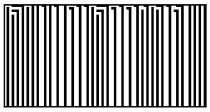
Rule 68 Seed

Rule 68 Random

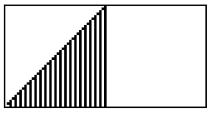
Rule 69 Seed



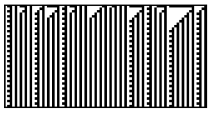
Rule 69 Random



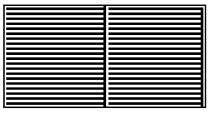
Rule 70 Seed



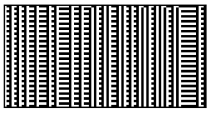
Rule 70 Random



Rule 71 Seed

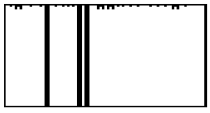


Rule 71 Random

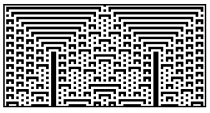


Rule 72 Seed

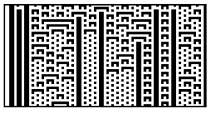
Rule 72 Random



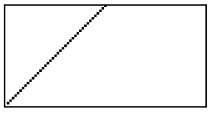
Rule 73 Seed



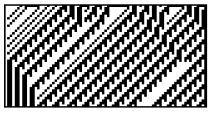
Rule 73 Random



Rule 74 Seed

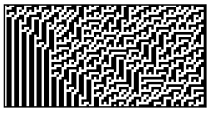


Rule 74 Random

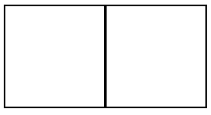


Rule 75 Seed

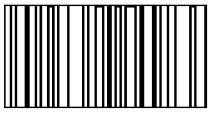
Rule 75 Random



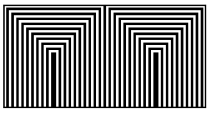
Rule 76 Seed



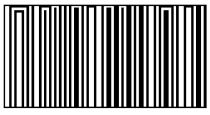
Rule 76 Random



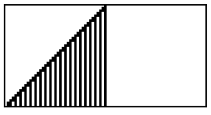
Rule 77 Seed



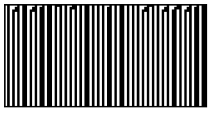
Rule 77 Random



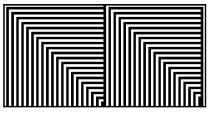
Rule 78 Seed



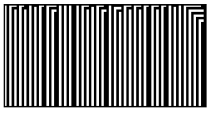
Rule 78 Random



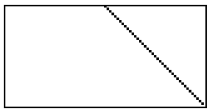
Rule 79 Seed



Rule 79 Random



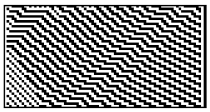
Rule 80 Seed



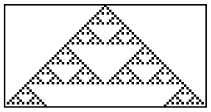
Rule 80 Random

Rule 81 Seed

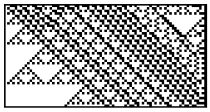
Rule 81 Random



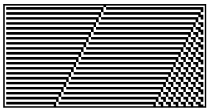
Rule 82 Seed



Rule 82 Random

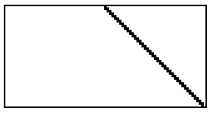


Rule 83 Seed

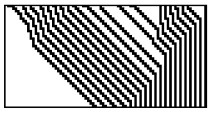


Rule 83 Random

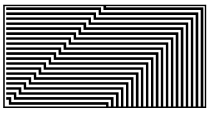
Rule 84 Seed



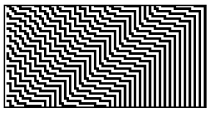
Rule 84 Random



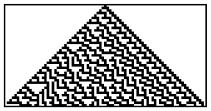
Rule 85 Seed



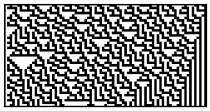
Rule 85 Random



Rule 86 Seed



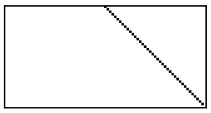
Rule 86 Random



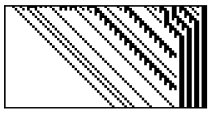
Rule 87 Seed

Rule 87 Random

Rule 88 Seed

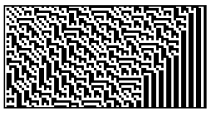


Rule 88 Random

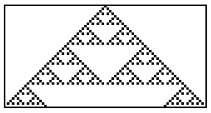


Rule 89 Seed

Rule 89 Random

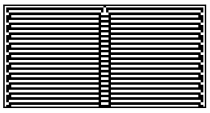


Rule 90 Seed



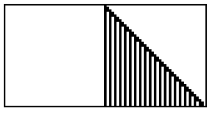
Rule 90 Random

Rule 91 Seed

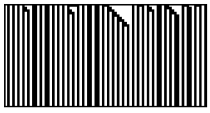


Rule 91 Random

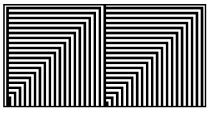
Rule 92 Seed



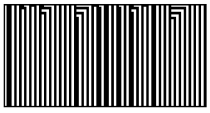
Rule 92 Random



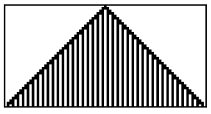
Rule 93 Seed



Rule 93 Random

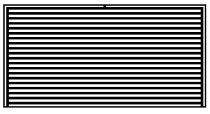


Rule 94 Seed

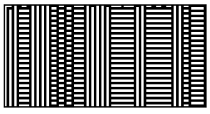


Rule 94 Random

Rule 95 Seed

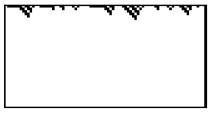


Rule 95 Random



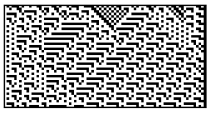
Rule 96 Seed

Rule 96 Random

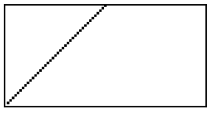


Rule 97 Seed

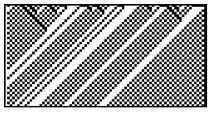
Rule 97 Random



Rule 98 Seed



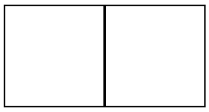
Rule 98 Random



Rule 99 Seed

Rule 99 Random

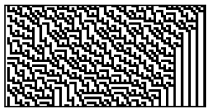
Rule 100 Seed



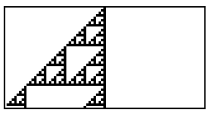
Rule 100 Random

Rule 101 Seed

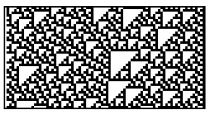
Rule 101 Random



Rule 102 Seed



Rule 102 Random



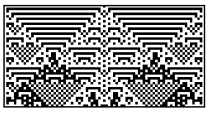
Rule 103 Seed

Rule 103 Random

Rule 104 Seed

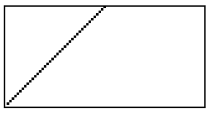
Rule 104 Random

Rule 105 Seed

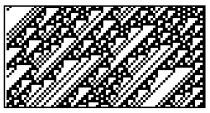


Rule 105 Random

Rule 106 Seed

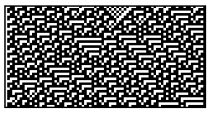


Rule 106 Random



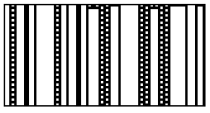
Rule 107 Seed

Rule 107 Random

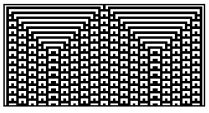


Rule 108 Seed

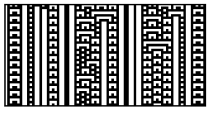
Rule 108 Random



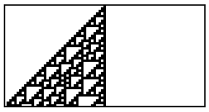
Rule 109 Seed



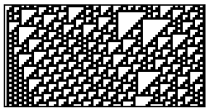
Rule 109 Random



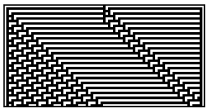
Rule 110 Seed



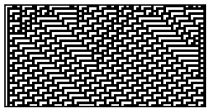
Rule 110 Random



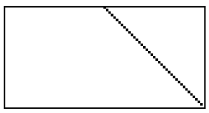
Rule 111 Seed



Rule 111 Random



Rule 112 Seed

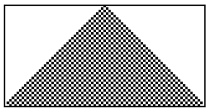


Rule 112 Random

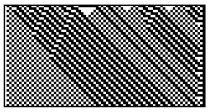
Rule 113 Seed

Rule 113 Random

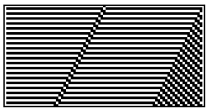
Rule 114 Seed



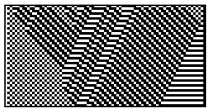
Rule 114 Random



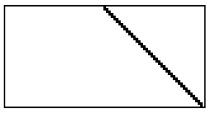
Rule 115 Seed



Rule 115 Random



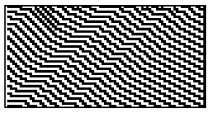
Rule 116 Seed



Rule 116 Random

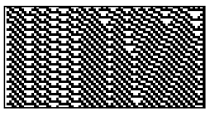
Rule 117 Seed

Rule 117 Random

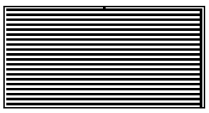


Rule 118 Seed

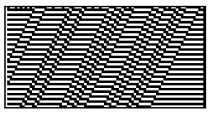
Rule 118 Random



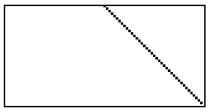
Rule 119 Seed



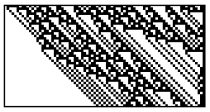
Rule 119 Random



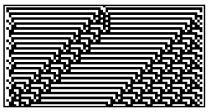
Rule 120 Seed



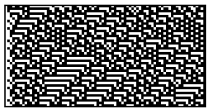
Rule 120 Random



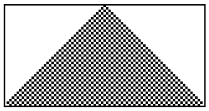
Rule 121 Seed



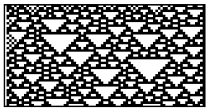
Rule 121 Random



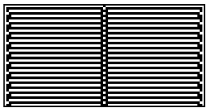
Rule 122 Seed



Rule 122 Random

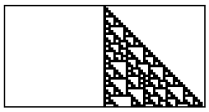


Rule 123 Seed

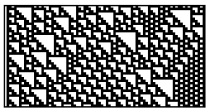


Rule 123 Random

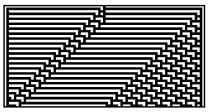
Rule 124 Seed



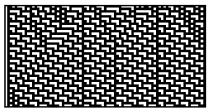
Rule 124 Random



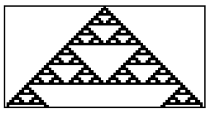
Rule 125 Seed



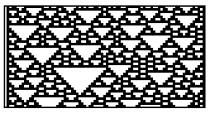
Rule 125 Random



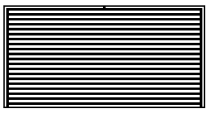
Rule 126 Seed



Rule 126 Random

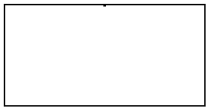


Rule 127 Seed



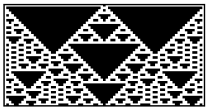
Rule 127 Random

Rule 128 Seed

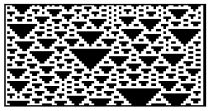


Rule 128 Random

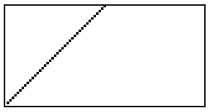
Rule 129 Seed



Rule 129 Random

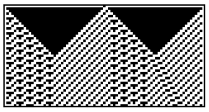


Rule 130 Seed

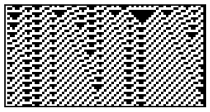


Rule 130 Random

Rule 131 Seed



Rule 131 Random

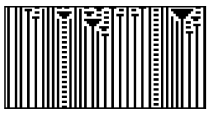


Rule 132 Seed

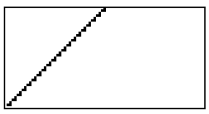
Rule 132 Random

Rule 133 Seed

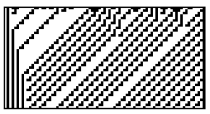
Rule 133 Random



Rule 134 Seed



Rule 134 Random



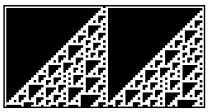
Rule 135 Seed

Rule 135 Random

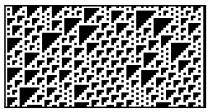
Rule 136 Seed

Rule 136 Random

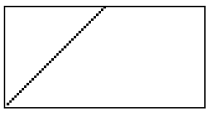
Rule 137 Seed



Rule 137 Random



Rule 138 Seed

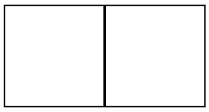


Rule 138 Random

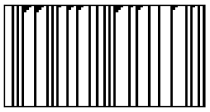
Rule 139 Seed

Rule 139 Random

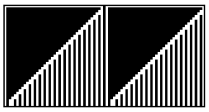
Rule 140 Seed



Rule 140 Random



Rule 141 Seed



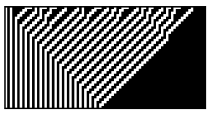
Rule 141 Random

Rule 142 Seed

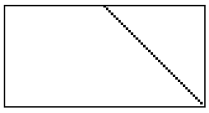
Rule 142 Random

Rule 143 Seed

Rule 143 Random

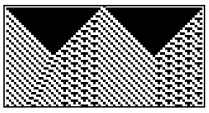


Rule 144 Seed

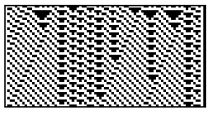


Rule 144 Random

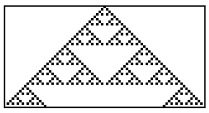
Rule 145 Seed



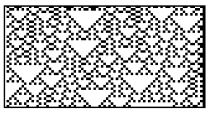
Rule 145 Random



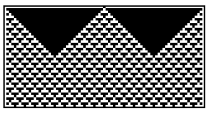
Rule 146 Seed



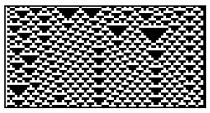
Rule 146 Random



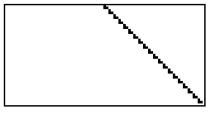
Rule 147 Seed



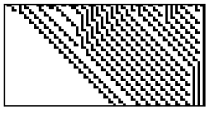
Rule 147 Random



Rule 148 Seed

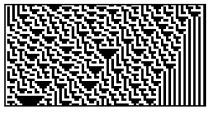


Rule 148 Random

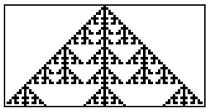


Rule 149 Seed

Rule 149 Random



Rule 150 Seed

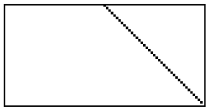


Rule 150 Random

Rule 151 Seed

Rule 151 Random

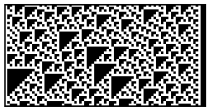
Rule 152 Seed



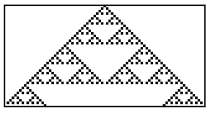
Rule 152 Random

Rule 153 Seed

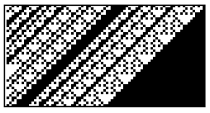
Rule 153 Random



Rule 154 Seed

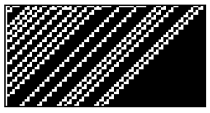


Rule 154 Random

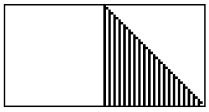


Rule 155 Seed

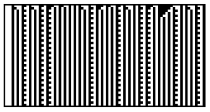
Rule 155 Random



Rule 156 Seed

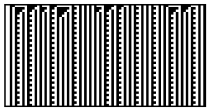


Rule 156 Random

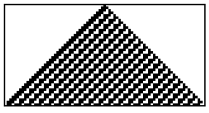


Rule 157 Seed

Rule 157 Random



Rule 158 Seed

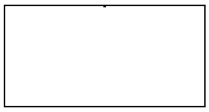


Rule 158 Random

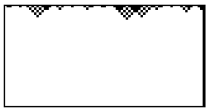
Rule 159 Seed

Rule 159 Random

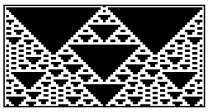
Rule 160 Seed



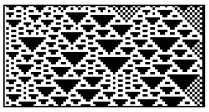
Rule 160 Random



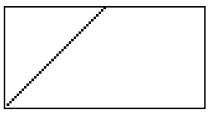
Rule 161 Seed



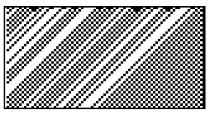
Rule 161 Random



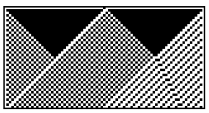
Rule 162 Seed



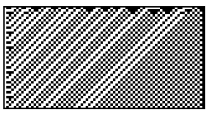
Rule 162 Random



Rule 163 Seed

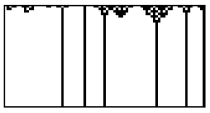


Rule 163 Random

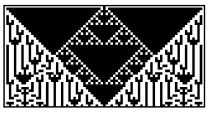


Rule 164 Seed

Rule 164 Random

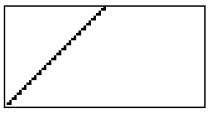


Rule 165 Seed

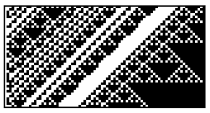


Rule 165 Random

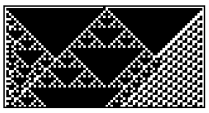
Rule 166 Seed



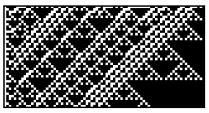
Rule 166 Random



Rule 167 Seed



Rule 167 Random



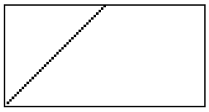
Rule 168 Seed

Rule 168 Random

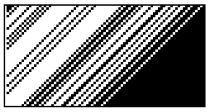
Rule 169 Seed

Rule 169 Random

Rule 170 Seed



Rule 170 Random



Rule 171 Seed

Rule 171 Random

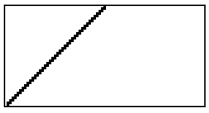
Rule 172 Seed

Rule 172 Random

Rule 173 Seed

Rule 173 Random

Rule 174 Seed

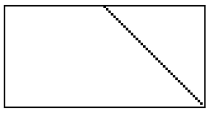


Rule 174 Random

Rule 175 Seed

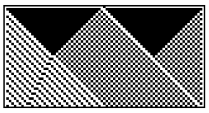
Rule 175 Random

Rule 176 Seed

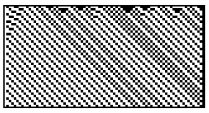


Rule 176 Random

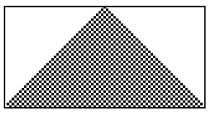
Rule 177 Seed



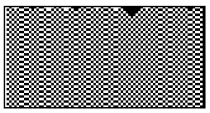
Rule 177 Random



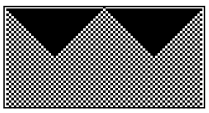
Rule 178 Seed



Rule 178 Random

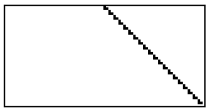


Rule 179 Seed

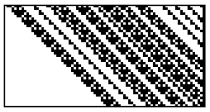


Rule 179 Random

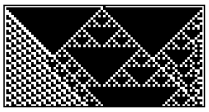
Rule 180 Seed



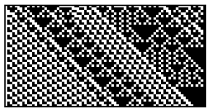
Rule 180 Random



Rule 181 Seed

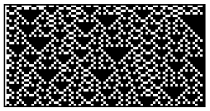


Rule 181 Random

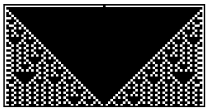


Rule 182 Seed

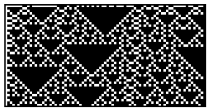
Rule 182 Random



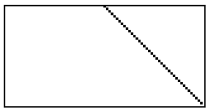
Rule 183 Seed



Rule 183 Random



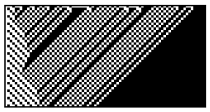
Rule 184 Seed



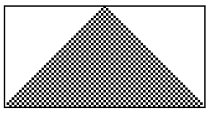
Rule 184 Random

Rule 185 Seed

Rule 185 Random



Rule 186 Seed

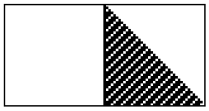


Rule 186 Random

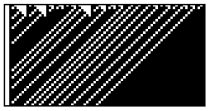
Rule 187 Seed

Rule 187 Random

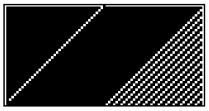
Rule 188 Seed



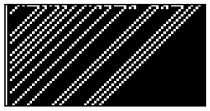
Rule 188 Random



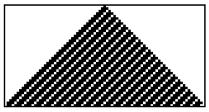
Rule 189 Seed



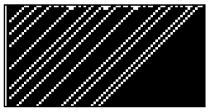
Rule 189 Random



Rule 190 Seed

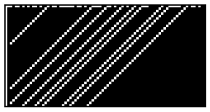


Rule 190 Random



Rule 191 Seed

Rule 191 Random

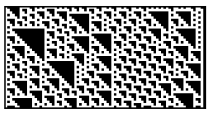


Rule 192 Seed

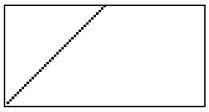
Rule 192 Random

Rule 193 Seed

Rule 193 Random

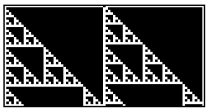


Rule 194 Seed



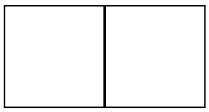
Rule 194 Random

Rule 195 Seed

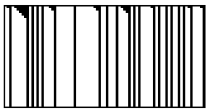


Rule 195 Random

Rule 196 Seed

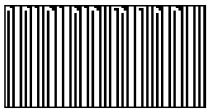


Rule 196 Random

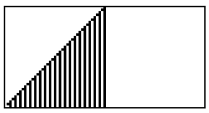


Rule 197 Seed

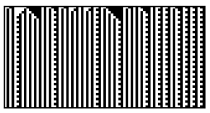
Rule 197 Random



Rule 198 Seed

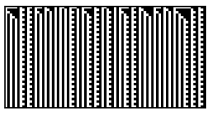


Rule 198 Random

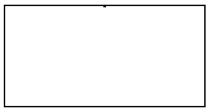


Rule 199 Seed

Rule 199 Random

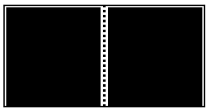


Rule 200 Seed

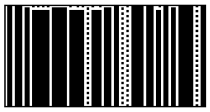


Rule 200 Random

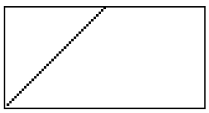
Rule 201 Seed



Rule 201 Random



Rule 202 Seed



Rule 202 Random

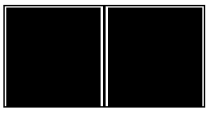
Rule 203 Seed

Rule 203 Random

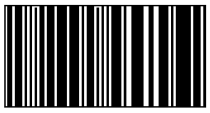
Rule 204 Seed

Rule 204 Random

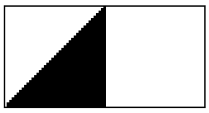
Rule 205 Seed



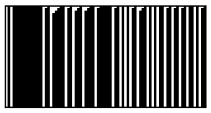
Rule 205 Random



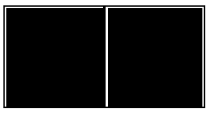
Rule 206 Seed



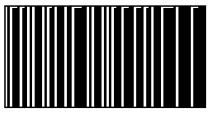
Rule 206 Random



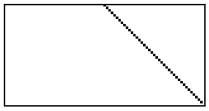
Rule 207 Seed



Rule 207 Random

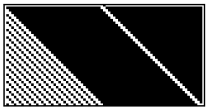


Rule 208 Seed



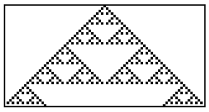
Rule 208 Random

Rule 209 Seed



Rule 209 Random

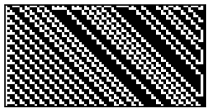
Rule 210 Seed



Rule 210 Random

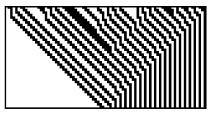
Rule 211 Seed

Rule 211 Random



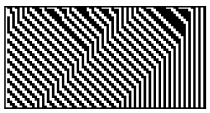
Rule 212 Seed

Rule 212 Random

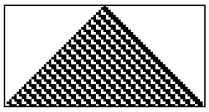


Rule 213 Seed

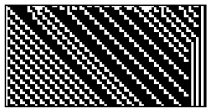
Rule 213 Random



Rule 214 Seed



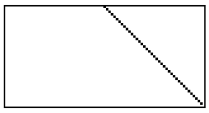
Rule 214 Random



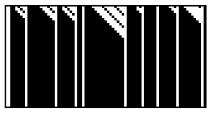
Rule 215 Seed

Rule 215 Random

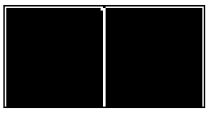
Rule 216 Seed



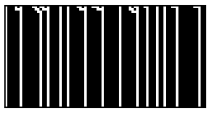
Rule 216 Random



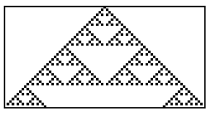
Rule 217 Seed



Rule 217 Random

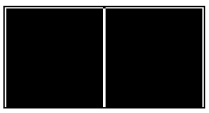


Rule 218 Seed



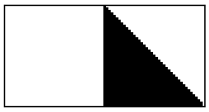
Rule 218 Random

Rule 219 Seed

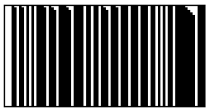


Rule 219 Random

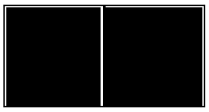
Rule 220 Seed



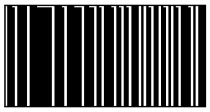
Rule 220 Random



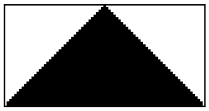
Rule 221 Seed



Rule 221 Random



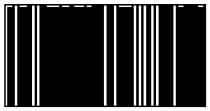
Rule 222 Seed



Rule 222 Random

Rule 223 Seed

Rule 223 Random



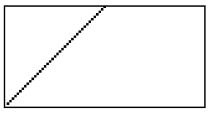
Rule 224 Seed

Rule 224 Random

Rule 225 Seed

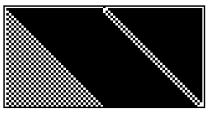
Rule 225 Random

Rule 226 Seed

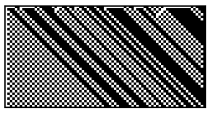


Rule 226 Random

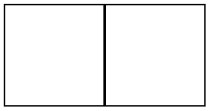
Rule 227 Seed



Rule 227 Random



Rule 228 Seed

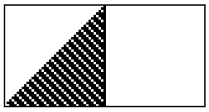


Rule 228 Random

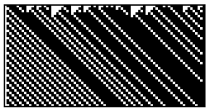
Rule 229 Seed

Rule 229 Random

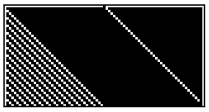
Rule 230 Seed



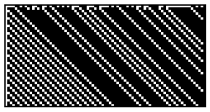
Rule 230 Random



Rule 231 Seed



Rule 231 Random



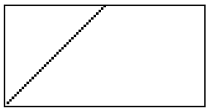
Rule 232 Seed

Rule 232 Random

Rule 233 Seed

Rule 233 Random

Rule 234 Seed

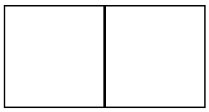


Rule 234 Random

Rule 235 Seed

Rule 235 Random

Rule 236 Seed

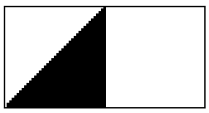


Rule 236 Random

Rule 237 Seed

Rule 237 Random

Rule 238 Seed



Rule 238 Random

Rule 239 Seed

Rule 239 Random

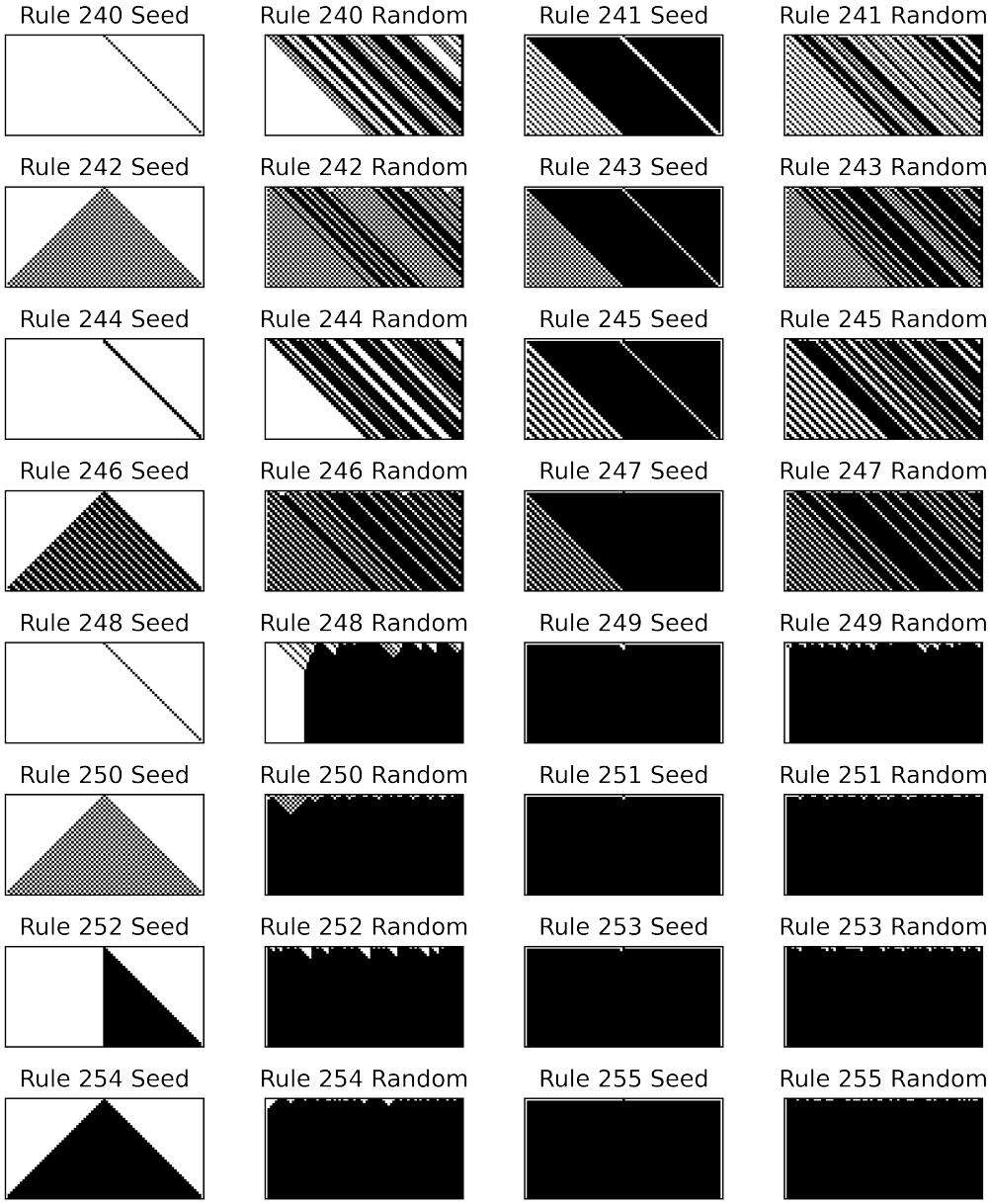


Figure A.2 Simulation of Elementary Cellular Automata Rules for Length $L = 80$ and Seed or Random Initial Conditions and Fixed Boundary Conditions.

A.2.2 Quantities to Classify Elementary Cellular Automata

Rule	Equivalents	Algebraic Form	W	λ_0	λ_1	μ	K_R	K_S	K_T	H_S	H_T	S_1	S_2
0	255	0	1	0	1	0.000	0.000	0.000	0.000	0.001	0.003	0.000	0.000
1	127	(1+p)(1+q)(1+r)	2	1/8	7/8	0.125	0.000	0.446	0.005	0.543	0.561	0.130	0.345
2	16, 191, 247	(1+p)(1+q)r	2	1/8	7/8	0.125	0.200	0.502	0.112	0.423	0.423	0.130	0.130
3	17, 63, 119	(1+p)(1+q)	2	1/4	3/4	0.167	0.200	0.728	0.153	0.595	0.686	0.189	0.308
4	223	(1+p)q(1+r)	2	1/8	7/8	0.125	0.400	0.512	0.000	0.367	0.002	0.130	0.130
5	95	(1+p)(1+r)	2	1/4	3/4	0.167	0.400	0.755	0.003	0.582	0.430	0.189	0.308
6	20, 159, 215	(1+p)(q+r)	2	1/4	3/4	0.250	0.400	0.629	0.132	0.520	0.528	0.500	0.547
7	21, 31, 87	(1+p)(1+qr)	2	3/8	5/8	0.208	0.400	0.490	0.108	0.682	0.691	0.308	0.461
8	64, 239, 253	(1+p)qr	1	1/8	7/8	0.125	0.600	0.000	0.000	0.001	0.004	0.130	0.000
9	65, 111, 125	(1+p)(1+q+r)	2	1/4	3/4	0.250	0.600	0.425	0.077	0.673	0.675	0.500	0.802
10	80, 175, 245	(1+p)r	2	1/4	3/4	0.167	0.600	0.734	0.150	0.592	0.592	0.189	0.189
11	47, 81, 117	(1+p)(1+q+qr)	2	3/8	5/8	0.208	0.600	0.577	0.122	0.688	0.693	0.308	0.510
12	68, 207, 221	(1+p)q	2	1/4	3/4	0.167	0.600	0.681	0.000	0.588	0.002	0.189	0.189
13	69, 79, 93	(1+p)(1+r+qr)	2	3/8	5/8	0.208	0.600	0.393	0.000	0.685	0.004	0.308	0.510
14	84, 143, 213	(1+p)(q+r+qr)	2	3/8	5/8	0.208	0.600	0.307	0.074	0.679	0.679	0.308	0.549
15	85	1+p	2	1/2	1/2	0.167	0.600	0.967	0.194	0.681	0.691	0.000	0.000
16	2, 191, 247	p(1+q)(1+r)	2	1/8	7/8	0.125	0.600	0.509	0.114	0.396	0.396	0.130	0.130
17	3, 63, 119	(1+q)(1+r)	2	1/4	3/4	0.167	0.600	0.750	0.154	0.552	0.685	0.189	0.308
18	183	(1+q)(p+r)	3	1/4	3/4	0.250	0.600	0.653	0.566	0.564	0.568	0.500	0.594
19	55	(1+q)(1+pr)	2	3/8	5/8	0.208	0.600	0.693	0.005	0.688	0.693	0.308	0.419
20	6, 159, 215	(p+q)(1+r)	2	1/4	3/4	0.250	0.600	0.642	0.138	0.518	0.521	0.500	0.547
21	7, 31, 87	(1+pq)(1+r)	2	3/8	5/8	0.208	0.600	0.569	0.120	0.685	0.691	0.308	0.461
22	151	p+q+r+pqr	3	3/8	5/8	0.375	1.000	0.837	0.857	0.638	0.647	0.808	0.913
23		1+pq+pr+qr	2	1/2	1/2	0.250	1.000	0.644	0.005	0.691	0.693	0.434	0.466
24	66, 189, 231	p+pq+pr+qr	2	1/4	3/4	0.250	1.000	0.613	0.128	0.470	0.471	0.811	0.696
25	61, 67, 103	1+q+r+pqr	2	3/8	5/8	0.292	1.000	0.450	0.098	0.688	0.688	0.808	0.848
26	82, 167, 181	p+pq+r+pqr	2	3/8	5/8	0.292	1.000	0.828	0.242	0.670	0.672	0.808	0.818
27	39, 53, 83	1+q+pr+qr	2	1/2	1/2	0.250	1.000	0.862	0.180	0.684	0.689	0.623	0.471
28	70, 157, 199	p+q+pr+pqr	2	3/8	5/8	0.292	1.000	0.541	0.001	0.688	0.101	0.808	0.750
29	71	1+pq+r+qr	2	1/2	1/2	0.250	1.000	0.850	0.002	0.691	0.320	0.623	0.521
30	86, 135, 149	p+q+r+qr	3	1/2	1/2	0.333	0.600	0.970	0.999	0.688	0.693	0.811	0.954
31	7, 21, 87	1+pq+pr+pqr	2	5/8	3/8	0.208	0.600	0.530	0.112	0.626	0.690	0.308	0.461
32	251	p(1+q)r	1	1/8	7/8	0.125	0.600	0.000	0.000	0.001	0.005	0.130	0.041
33	123	(1+q)(1+p+r)	2	1/4	3/4	0.250	0.600	0.653	0.004	0.591	0.576	0.500	0.860
34	48, 187, 243	(1+q)r	2	1/4	3/4	0.167	0.600	0.700	0.145	0.555	0.555	0.189	0.189
35	49, 59, 115	(1+q)(1+p+pr)	2	3/8	5/8	0.208	0.600	0.707	0.147	0.641	0.665	0.308	0.454
36	219	q+pq+pr+qr	2	1/4	3/4	0.250	0.400	0.390	0.000	0.167	0.005	0.811	0.337
37	91	1+p+r+pqr	2	3/8	5/8	0.292	0.400	0.660	0.004	0.659	0.532	0.808	0.915
38	52, 155, 211	q+pq+r+pqr	2	3/8	5/8	0.292	0.600	0.778	0.160	0.610	0.611	0.808	0.365
39	27, 53, 83	1+p+pr+qr	2	1/2	1/2	0.250	0.600	0.820	0.183	0.681	0.691	0.623	0.471
40	96, 235, 249	(p+q)r	1	1/4	3/4	0.250	0.600	0.000	0.000	0.002	0.007	0.500	0.379
41	97, 107, 121	1+p+q+pq+r+pqr	2	3/8	5/8	0.375	0.600	0.728	0.250	0.654	0.659	0.808	0.889
42	112, 171, 241	(1+pq)r	2	3/8	5/8	0.208	0.400	0.863	0.173	0.670	0.671	0.308	0.308
43	113	1+p+q+pq+pr+qr	2	1/2	1/2	0.250	0.400	0.270	0.065	0.693	0.693	0.434	0.573
44	100, 203, 217	q+pq+pr+pqr	2	3/8	5/8	0.292	1.000	0.561	0.000	0.440	0.005	0.808	0.669
45	75, 89, 101	1+p+r+qr	3	1/2	1/2	0.333	1.000	0.969	0.999	0.688	0.693	0.811	1.000
46	116, 139, 209	q+pq+r+qr	2	1/2	1/2	0.250	1.000	0.648	0.135	0.641	0.641	0.623	0.625
47	11, 81, 117	1+p+pr+pqr	2	5/8	3/8	0.208	1.000	0.666	0.138	0.692	0.692	0.308	0.510
48	34, 187, 243	p(1+q)	2	1/4	3/4	0.167	1.000	0.723	0.149	0.542	0.542	0.189	0.189
49	35, 59, 115	(1+q)(1+r+pr)	2	3/8	5/8	0.208	1.000	0.643	0.135	0.652	0.663	0.308	0.454
50	179	(1+q)(p+r+pr)	2	3/8	5/8	0.208	0.600	0.583	0.005	0.693	0.693	0.308	0.461
51		1+q	2	1/2	1/2	0.167	0.600	1.000	0.005	0.692	0.693	0.000	0.000
52	38, 155, 211	p+q+qr+pqr	2	3/8	5/8	0.292	1.000	0.736	0.155	0.616	0.616	0.808	0.365
53	27, 39, 83	1+pq+r+pr	2	1/2	1/2	0.250	1.000	0.853	0.184	0.689	0.692	0.623	0.471
54	147	p+q+r+pr	4	1/2	1/2	0.333	0.600	0.698	0.423	0.675	0.690	0.811	0.931
55	19	1+pq+qr+pqr	2	5/8	3/8	0.208	0.600	0.620	0.005	0.647	0.693	0.308	0.419
56	98, 185, 227	p+pq+qr+pqr	2	3/8	5/8	0.292	1.000	0.619	0.131	0.671	0.671	0.808	0.930
57	99	1+q+r+pr	2	1/2	1/2	0.333	1.000	0.188	0.056	0.693	0.693	0.811	0.909
58	114, 163, 177	p+pq+r+pr	2	1/2	1/2	0.250	1.000	0.368	0.087	0.664	0.664	0.623	0.656
59	35, 49, 115	1+q+qr+pqr	2	5/8	3/8	0.208	1.000	0.672	0.136	0.656	0.662	0.308	0.454
60	102, 153, 195	p+q	3	1/2	1/2	0.333	0.600	0.972	1.000	0.688	0.693	1.000	1.000
61	25, 67, 103	1+pq+r+pr+qr+pqr	2	5/8	3/8	0.292	0.600	0.416	0.105	0.686	0.688	0.808	0.848
62	118, 131, 145	p+q+r+pr+qr+pqr	2	5/8	3/8	0.292	0.400	0.371	0.027	0.651	0.639	0.808	0.868
63	3, 17, 119	1+pq	2	3/4	1/4	0.167	0.400	0.724	0.153	0.599	0.683	0.189	0.308
64	8, 239, 253	pq(1+r)	1	1/8	7/8	0.125	0.400	0.000	0.000	0.001	0.004	0.130	0.000
65	9, 111, 125	(1+p+q)(1+r)	2	1/4	3/4	0.250	0.400	0.437	0.088	0.673	0.670	0.500	0.802
66	24, 189, 231	pq+r+pr+qr	2	1/4	3/4	0.250	0.600	0.608	0.129	0.478	0.479	0.811	0.696
67	25, 61, 103	1+p+q+pqr	2	3/8	5/8	0.292	0.600	0.416	0.091	0.687	0.689	0.808	0.848

Continued on next page

Continued from previous page

Rule	Equivalents	Algebraic Form	W	λ_0	λ_1	μ	K_R	K_S	K_T	H_S	H_T	S_1	S_2
68	12, 207, 221	q(1+r)	2	1/4	3/4	0.167	0.600	0.696	0.000	0.551	0.002	0.189	0.189
69	13, 79, 93	(1+p+pq)(1+r)	2	3/8	5/8	0.208	0.600	0.417	0.000	0.687	0.004	0.308	0.510
70	28, 157, 199	q+r+pr+pqr	2	3/8	5/8	0.292	1.000	0.523	0.001	0.693	0.081	0.808	0.750
71	29	1+p+pq+qr	2	1/2	1/2	0.250	1.000	0.886	0.002	0.686	0.244	0.623	0.521
72	237	q(p+r)	2	1/4	3/4	0.250	0.400	0.319	0.000	0.404	0.004	0.500	0.352
73	109	1+p+q+r+pr+pqr	2	3/8	5/8	0.375	0.400	0.841	0.024	0.685	0.258	0.808	0.827
74	88, 173, 229	pq+r+pr+pqr	2	3/8	5/8	0.292	0.600	0.830	0.158	0.631	0.630	0.808	0.810
75	45, 89, 101	1+p+q+qr	3	1/2	1/2	0.333	0.600	0.970	0.999	0.688	0.693	0.811	1.000
76	205	q(1+pr)	2	3/8	5/8	0.208	1.000	0.887	0.000	0.643	0.001	0.308	0.308
77		1+p+pq+r+pr+qr	2	1/2	1/2	0.250	1.000	0.577	0.000	0.693	0.002	0.434	0.466
78	92, 141, 197	q+r+pr+qr	2	1/2	1/2	0.250	1.000	0.380	0.000	0.682	0.006	0.623	0.656
79	13, 69, 93	1+p+pq+pqr	2	5/8	3/8	0.208	1.000	0.383	0.000	0.686	0.003	0.308	0.510
80	10, 175, 245	p(1+r)	2	1/4	3/4	0.167	0.600	0.743	0.152	0.562	0.562	0.189	0.189
81	11, 47, 117	(1+q+pq)(1+r)	2	3/8	5/8	0.208	0.600	0.651	0.139	0.693	0.692	0.308	0.510
82	26, 167, 181	p+r+qr+pqr	2	3/8	5/8	0.292	0.600	0.829	0.242	0.675	0.673	0.808	0.818
83	27, 39, 53	1+q+pq+pr	2	1/2	1/2	0.250	0.600	0.848	0.183	0.687	0.690	0.623	0.471
84	14, 143, 213	(p+q+pq)(1+r)	2	3/8	5/8	0.208	0.200	0.324	0.079	0.693	0.691	0.308	0.549
85	15	1+r	2	1/2	1/2	0.167	0.200	0.960	0.190	0.692	0.693	0.000	0.000
86	30, 135, 149	p+q+pq+r	3	1/2	1/2	0.333	0.400	0.971	0.999	0.688	0.693	0.811	0.954
87	7, 21, 31	1+pr+qr+pqr	2	5/8	3/8	0.208	0.400	0.492	0.107	0.651	0.691	0.308	0.461
88	74, 173, 229	p+pr+qr+pqr	2	3/8	5/8	0.292	1.000	0.780	0.151	0.627	0.626	0.808	0.810
89	45, 75, 101	1+q+pq+r	3	1/2	1/2	0.333	1.000	0.969	0.998	0.688	0.693	0.811	1.000
90	165	p+r	3	1/2	1/2	0.333	0.600	0.970	0.999	0.688	0.692	1.000	1.000
91	37	1+q+pq+pr+qr+pqr	2	5/8	3/8	0.292	0.600	0.618	0.004	0.657	0.533	0.808	0.915
92	78, 141, 197	p+q+pq+pr	2	1/2	1/2	0.250	1.000	0.359	0.000	0.684	0.004	0.623	0.656
93	13, 69, 79	1+r+qr+pqr	2	5/8	3/8	0.208	1.000	0.408	0.000	0.687	0.004	0.308	0.510
94	133	p+q+pq+r+qr+pqr	2	5/8	3/8	0.292	0.600	0.599	0.001	0.676	0.045	0.808	0.868
95	5	1+pr	2	3/4	1/4	0.167	0.600	0.690	0.003	0.580	0.395	0.189	0.308
96	40, 235, 249	p(q+r)	1	1/4	3/4	0.250	0.600	0.000	0.000	0.002	0.006	0.500	0.379
97	41, 107, 121	1+p+q+r+qr+pqr	2	3/8	5/8	0.375	0.600	0.764	0.259	0.657	0.658	0.808	0.889
98	56, 185, 227	pq+r+qr+pqr	2	3/8	5/8	0.292	1.000	0.648	0.135	0.647	0.648	0.808	0.930
99	57	1+p+q+pr	2	1/2	1/2	0.333	1.000	0.101	0.027	0.693	0.693	0.811	0.909
100	44, 203, 217	q+pr+qr+pqr	2	3/8	5/8	0.292	1.000	0.586	0.000	0.472	0.005	0.808	0.669
101	45, 75, 89	1+p+pq+r	3	1/2	1/2	0.333	1.000	0.971	1.000	0.688	0.693	0.811	1.000
102	60, 153, 195	q+r	3	1/2	1/2	0.333	0.600	0.971	0.999	0.689	0.693	1.000	1.000
103	25, 61, 67	1+p+pq+pr+qr+pqr	2	5/8	3/8	0.292	0.600	0.433	0.103	0.686	0.689	0.808	0.848
104	233	pq+pr+qr+pqr	2	3/8	5/8	0.375	1.000	0.276	0.000	0.199	0.007	0.808	0.719
105		1+p+q+r	3	1/2	1/2	0.500	1.000	0.970	0.999	0.688	0.693	1.000	1.000
106	120, 169, 225	pq+r	4	1/2	1/2	0.333	0.600	0.971	0.999	0.688	0.693	0.811	0.811
107	41, 97, 121	1+p+q+pr+qr+pqr	2	5/8	3/8	0.375	0.600	0.741	0.270	0.658	0.660	0.808	0.889
108	201	q+pr	2	1/2	1/2	0.333	0.400	0.779	0.000	0.623	0.050	0.811	0.424
109	73	1+p+pq+r+qr+pqr	2	5/8	3/8	0.375	0.400	0.856	0.016	0.690	0.175	0.808	0.827
110	124, 137, 193	q+r+qr+pqr	4	5/8	3/8	0.292	0.600	0.612	0.298	0.682	0.659	0.808	0.960
111	9, 65, 125	1+p+pq+pr	2	3/4	1/4	0.250	0.600	0.380	0.073	0.672	0.641	0.500	0.802
112	42, 171, 241	p(1+qr)	2	3/8	5/8	0.208	1.000	0.876	0.174	0.662	0.662	0.308	0.308
113	43	1+q+pq+r+pr+qr	2	1/2	1/2	0.250	1.000	0.369	0.083	0.693	0.693	0.434	0.573
114	58, 163, 177	p+r+pr+qr	2	1/2	1/2	0.250	1.000	0.392	0.093	0.656	0.656	0.623	0.656
115	35, 49, 59	1+q+pq+pqr	2	5/8	3/8	0.208	1.000	0.672	0.139	0.665	0.670	0.308	0.454
116	46, 139, 209	p+q+pq+qr	2	1/2	1/2	0.250	1.000	0.647	0.134	0.653	0.653	0.623	0.625
117	11, 47, 81	1+r+pr+pqr	2	5/8	3/8	0.208	1.000	0.652	0.133	0.686	0.693	0.308	0.510
118	62, 131, 145	p+q+pq+r+pr+pqr	2	5/8	3/8	0.292	0.600	0.415	0.022	0.644	0.640	0.808	0.868
119	3, 17, 63	1+qr	2	3/4	1/4	0.167	0.600	0.736	0.155	0.579	0.689	0.189	0.308
120	106, 169, 225	p+qr	4	1/2	1/2	0.333	0.600	0.970	1.000	0.688	0.693	0.811	0.811
121	41, 97, 107	1+q+pq+r+pr+pqr	2	5/8	3/8	0.375	0.600	0.755	0.264	0.652	0.650	0.808	0.889
122	161	p+r+pr+pqr	3	5/8	3/8	0.292	0.600	0.662	0.843	0.683	0.693	0.808	0.894
123	33	1+q+pq+qr	2	3/4	1/4	0.250	0.600	0.669	0.004	0.645	0.548	0.500	0.860
124	110, 137, 193	p+q+pq+pqr	4	5/8	3/8	0.292	0.400	0.575	0.316	0.681	0.670	0.808	0.960
125	9, 65, 111	1+r+pr+qr	2	3/4	1/4	0.250	0.400	0.488	0.096	0.670	0.609	0.500	0.802
126	129	p+q+pq+r+pr+qr	3	3/4	1/4	0.250	0.200	0.669	0.845	0.683	0.692	0.811	0.989
127	1	1+pqr	2	7/8	1/8	0.125	0.200	0.480	0.005	0.462	0.610	0.130	0.345
128	254	pqr	1	1/8	7/8	0.125	0.200	0.000	0.000	0.001	0.005	0.130	0.041
129	126	1+p+q+pq+r+pr+qr	3	1/4	3/4	0.250	0.200	0.666	0.812	0.683	0.693	0.811	0.989
130	144, 190, 246	(1+p+q)r	2	1/4	3/4	0.250	0.400	0.582	0.125	0.456	0.456	0.500	0.421
131	62, 118, 145	1+p+q+pq+pqr	2	3/8	5/8	0.292	0.400	0.459	0.017	0.645	0.640	0.808	0.868
132	222	q(1+p+r)	2	1/4	3/4	0.250	0.600	0.620	0.000	0.456	0.003	0.500	0.440
133	94	1+p+r+pr+pqr	2	3/8	5/8	0.292	0.600	0.611	0.001	0.677	0.074	0.808	0.868
134	148, 158, 214	q+pq+r+pr+pqr	2	3/8	5/8	0.375	0.600	0.595	0.132	0.605	0.610	0.808	0.733
135	30, 86, 149	1+p+qr	3	1/2	1/2	0.333	0.600	0.969	1.000	0.688	0.693	0.811	0.954
136	192, 238, 252	qr	1	1/4	3/4	0.167	0.600	0.000	0.000	0.002	0.006	0.189	0.130
137	110, 124, 193	1+p+q+pq+r+pr+pqr	4	3/8	5/8	0.292	0.600	0.609	0.303	0.681	0.669	0.808	0.960
138	174, 208, 244	(1+p+pq)r	2	3/8	5/8	0.208	1.000	0.849	0.170	0.667	0.667	0.308	0.308

Continued on next page

Continued from previous page

Rule	Equivalents	Algebraic Form	W	λ_0	λ_1	μ	K_R	K_S	K_T	H_S	H_T	S_1	S_2
139	46, 116, 209	1+p+q+pq+qr	2	1/2	1/2	0.250	1.000	0.641	0.134	0.669	0.669	0.623	0.625
140	196, 206, 220	q(1+p+pr)	2	3/8	5/8	0.208	1.000	0.706	0.000	0.568	0.003	0.308	0.327
141	78, 92, 197	1+p+r+pr+qr	2	1/2	1/2	0.250	1.000	0.448	0.000	0.683	0.005	0.623	0.656
142	212	q+pq+r+pr+qr	2	1/2	1/2	0.250	1.000	0.362	0.087	0.693	0.693	0.434	0.573
143	14, 84, 213	1+p+pqr	2	5/8	3/8	0.208	1.000	0.334	0.075	0.691	0.689	0.308	0.549
144	130, 190, 246	p(1+q+r)	2	1/4	3/4	0.250	0.600	0.563	0.121	0.414	0.414	0.500	0.421
145	62, 118, 131	1+q+r+qr+pqr	2	3/8	5/8	0.292	0.600	0.395	0.032	0.651	0.639	0.808	0.868
146	182	p+pq+r+qr+pqr	3	3/8	5/8	0.375	0.400	0.651	0.525	0.563	0.566	0.808	0.696
147	54	1+q+pr	4	1/2	1/2	0.333	0.400	0.621	0.335	0.637	0.686	0.811	0.931
148	134, 158, 214	p+q+pr+qr+pqr	2	3/8	5/8	0.375	0.600	0.609	0.131	0.596	0.595	0.808	0.733
149	30, 86, 135	1+pq+r	3	1/2	1/2	0.333	0.600	0.970	1.000	0.688	0.693	0.811	0.954
150		p+q+r	3	1/2	1/2	0.500	1.000	0.970	0.998	0.688	0.693	1.000	1.000
151	22	1+pq+pr+qr+pqr	3	5/8	3/8	0.375	1.000	0.838	0.857	0.638	0.647	0.808	0.913
152	188, 194, 230	p+pq+pr+qr+pqr	2	3/8	5/8	0.292	0.600	0.568	0.121	0.464	0.465	0.808	0.696
153	60, 102, 195	1+q+r	3	1/2	1/2	0.333	0.600	0.967	0.998	0.688	0.693	1.000	1.000
154	166, 180, 210	p+pq+r	2	1/2	1/2	0.333	1.000	0.976	0.234	0.688	0.690	0.811	0.544
155	38, 52, 211	1+q+pr+qr+pqr	2	5/8	3/8	0.292	1.000	0.778	0.162	0.581	0.581	0.808	0.365
156	198	p+q+pr	2	1/2	1/2	0.333	1.000	0.555	0.001	0.693	0.060	0.811	0.629
157	28, 70, 199	1+pq+r+qr+pqr	2	5/8	3/8	0.292	1.000	0.537	0.001	0.691	0.101	0.808	0.750
158	134, 148, 214	p+q+r+qr+pqr	2	5/8	3/8	0.375	0.600	0.599	0.130	0.596	0.594	0.808	0.733
159	6, 20, 215	1+pq+pr	2	3/4	1/4	0.250	0.600	0.612	0.131	0.532	0.533	0.500	0.547
160	250	pr	1	1/4	3/4	0.167	0.600	0.000	0.000	0.003	0.008	0.189	0.130
161	122	1+p+q+pq+r+qr+pqr	3	3/8	5/8	0.292	0.600	0.670	0.803	0.683	0.693	0.808	0.894
162	176, 186, 242	(1+q+pq)r	2	3/8	5/8	0.208	1.000	0.680	0.141	0.642	0.642	0.308	0.333
163	58, 114, 177	1+p+q+pq+pr	2	1/2	1/2	0.250	1.000	0.372	0.088	0.669	0.669	0.623	0.656
164	218	q+pq+pr+qr+pqr	2	3/8	5/8	0.292	0.600	0.411	0.000	0.303	0.007	0.808	0.688
165	90	1+p+r	3	1/2	1/2	0.333	0.600	0.969	0.997	0.688	0.693	1.000	1.000
166	154, 180, 210	q+pq+r	2	1/2	1/2	0.333	1.000	0.974	0.250	0.691	0.693	0.811	0.544
167	26, 82, 181	1+p+pr+qr+pqr	2	5/8	3/8	0.292	1.000	0.843	0.242	0.645	0.647	0.808	0.818
168	224, 234, 248	(p+q+pq)r	1	3/8	5/8	0.208	0.400	0.000	0.000	0.004	0.012	0.308	0.336
169	106, 120, 225	1+p+q+pq+r	4	1/2	1/2	0.333	0.400	0.971	0.999	0.688	0.693	0.811	0.811
170	240	r	2	1/2	1/2	0.167	0.200	0.970	0.192	0.688	0.688	0.000	0.000
171	42, 112, 241	1+p+q+pq+pr+qr+pqr	2	5/8	3/8	0.208	0.200	0.809	0.163	0.682	0.682	0.308	0.308
172	202, 216, 228	q+pq+pr	2	1/2	1/2	0.250	0.600	0.518	0.000	0.369	0.012	0.623	0.708
173	74, 88, 229	1+p+r+qr+pqr	2	5/8	3/8	0.292	0.600	0.792	0.152	0.619	0.619	0.808	0.810
174	138, 208, 244	q+pq+r+qr+pqr	2	5/8	3/8	0.208	0.600	0.830	0.166	0.653	0.653	0.308	0.308
175	10, 80, 245	1+p+pr	2	3/4	1/4	0.167	0.600	0.721	0.148	0.583	0.583	0.189	0.189
176	162, 186, 242	p(1+q+qr)	2	3/8	5/8	0.208	1.000	0.681	0.143	0.631	0.631	0.308	0.333
177	58, 114, 163	1+q+r+pr+qr	2	1/2	1/2	0.250	1.000	0.392	0.090	0.669	0.669	0.623	0.656
178		p+pq+r+pr+qr	2	1/2	1/2	0.250	1.000	0.629	0.005	0.693	0.693	0.434	0.466
179	50	1+q+pqr	2	5/8	3/8	0.208	1.000	0.663	0.005	0.693	0.693	0.308	0.461
180	154, 166, 210	p+q+qr	2	1/2	1/2	0.333	0.600	0.968	0.242	0.691	0.691	0.811	0.544
181	26, 82, 167	1+pq+r+pr+pqr	2	5/8	3/8	0.292	0.600	0.819	0.265	0.674	0.676	0.808	0.818
182	146	p+q+r+pr+pqr	3	5/8	3/8	0.375	0.400	0.660	0.557	0.562	0.567	0.808	0.696
183	18	1+pq+qr	3	3/4	1/4	0.250	0.400	0.649	0.609	0.564	0.567	0.500	0.594
184	226	p+pq+qr	2	1/2	1/2	0.250	1.000	0.255	0.065	0.693	0.693	0.623	0.747
185	56, 98, 227	1+q+r+pr+pqr	2	5/8	3/8	0.292	1.000	0.642	0.135	0.659	0.659	0.808	0.930
186	162, 176, 242	p+pq+r+pr+pqr	2	5/8	3/8	0.208	0.600	0.677	0.141	0.629	0.629	0.308	0.333
187	34, 48, 243	1+q+qr	2	3/4	1/4	0.167	0.600	0.697	0.144	0.557	0.557	0.189	0.189
188	152, 194, 230	p+q+pqr	2	5/8	3/8	0.292	0.600	0.587	0.126	0.501	0.501	0.808	0.696
189	24, 66, 231	1+pq+r+pr+qr	2	3/4	1/4	0.250	0.600	0.609	0.130	0.486	0.486	0.811	0.696
190	130, 144, 246	p+q+r+pr+qr	2	3/4	1/4	0.250	0.400	0.573	0.123	0.479	0.479	0.500	0.421
191	2, 16, 247	1+pq+pqr	2	7/8	1/8	0.125	0.400	0.464	0.105	0.395	0.395	0.130	0.130
192	136, 238, 252	pq	1	1/4	3/4	0.167	0.400	0.000	0.000	0.002	0.006	0.189	0.130
193	110, 124, 137	1+p+q+r+pr+qr+pqr	4	3/8	5/8	0.292	0.400	0.524	0.241	0.682	0.676	0.808	0.960
194	152, 188, 230	pq+r+pr+qr+pqr	2	3/8	5/8	0.292	0.600	0.576	0.123	0.479	0.479	0.808	0.696
195	60, 102, 153	1+p+q	3	1/2	1/2	0.333	0.600	0.970	0.998	0.688	0.693	1.000	1.000
196	140, 206, 220	q(1+r+pr)	2	3/8	5/8	0.208	1.000	0.755	0.000	0.583	0.003	0.308	0.327
197	78, 92, 141	1+p+pq+r+pr	2	1/2	1/2	0.250	1.000	0.408	0.000	0.685	0.004	0.623	0.656
198	156	q+r+pr	2	1/2	1/2	0.333	1.000	0.506	0.001	0.693	0.093	0.811	0.629
199	28, 70, 157	1+p+pq+qr+pqr	2	5/8	3/8	0.292	1.000	0.517	0.001	0.690	0.088	0.808	0.750
200	236	q(p+r+pr)	2	3/8	5/8	0.208	0.600	0.739	0.000	0.660	0.001	0.308	0.308
201	108	1+p+q+r+pr	2	1/2	1/2	0.333	0.600	0.801	0.000	0.608	0.033	0.811	0.424
202	172, 216, 228	pq+r+pr	2	1/2	1/2	0.250	1.000	0.515	0.000	0.398	0.010	0.623	0.708
203	44, 100, 217	1+p+q+qr+pqr	2	5/8	3/8	0.292	1.000	0.571	0.000	0.440	0.005	0.808	0.669
204		q	2	1/2	1/2	0.167	0.600	0.957	0.000	0.691	0.000	0.000	0.000
205	76	1+p+pq+r+pr+qr+pqr	2	5/8	3/8	0.208	0.600	0.834	0.000	0.661	0.001	0.308	0.308
206	140, 196, 220	q+r+pr+qr+pqr	2	5/8	3/8	0.208	1.000	0.718	0.000	0.583	0.003	0.308	0.327
207	12, 68, 221	1+p+pq	2	3/4	1/4	0.167	1.000	0.709	0.000	0.578	0.002	0.189	0.189
208	138, 174, 244	p(1+r+qr)	2	3/8	5/8	0.208	1.000	0.820	0.166	0.667	0.667	0.308	0.308
209	46, 116, 139	1+q+pq+r+qr	2	1/2	1/2	0.250	1.000	0.631	0.132	0.659	0.659	0.623	0.625

Continued on next page

Continued from previous page

Rule	Equivalents	Algebraic Form	W	λ_0	λ_1	μ	K_R	K_S	K_T	H_S	H_T	S_1	S_2
210	154, 166, 180	p+r+qr	2	1/2	1/2	0.333	1.000	0.963	0.262	0.690	0.690	0.811	0.544
211	38, 52, 155	1+q+pq+pr+pqr	2	5/8	3/8	0.292	1.000	0.785	0.163	0.622	0.622	0.808	0.365
212	142	p+q+pq+pr+qr	2	1/2	1/2	0.250	0.400	0.325	0.081	0.693	0.693	0.434	0.573
213	14, 84, 143	1+r+pqr	2	5/8	3/8	0.208	0.400	0.331	0.079	0.693	0.692	0.308	0.549
214	134, 148, 158	p+q+pq+r+pqr	2	5/8	3/8	0.375	0.600	0.649	0.137	0.600	0.600	0.808	0.733
215	6, 20, 159	1+pr+qr	2	3/4	1/4	0.250	0.600	0.636	0.132	0.580	0.579	0.500	0.547
216	172, 202, 228	p+pr+qr	2	1/2	1/2	0.250	0.600	0.497	0.000	0.388	0.011	0.623	0.708
217	44, 100, 203	1+q+pq+r+pqr	2	5/8	3/8	0.292	0.600	0.586	0.000	0.464	0.003	0.808	0.669
218	164	p+r+pqr	2	5/8	3/8	0.292	0.400	0.408	0.000	0.292	0.007	0.808	0.688
219	36	1+q+pq+pr+qr	2	3/4	1/4	0.250	0.400	0.396	0.000	0.199	0.005	0.811	0.337
220	140, 196, 206	p+q+pq+pr+pqr	2	5/8	3/8	0.208	0.600	0.706	0.000	0.533	0.003	0.308	0.327
221	12, 68, 207	1+r+qr	2	3/4	1/4	0.167	0.600	0.724	0.000	0.567	0.002	0.189	0.189
222	132	p+q+pq+r+qr	2	3/4	1/4	0.250	0.600	0.623	0.000	0.456	0.002	0.500	0.440
223	4	1+pr+pqr	2	7/8	1/8	0.125	0.600	0.525	0.000	0.414	0.002	0.130	0.130
224	168, 234, 248	p(q+r+qr)	1	3/8	5/8	0.208	0.600	0.000	0.000	0.005	0.017	0.308	0.336
225	106, 120, 169	1+p+q+r+qr	4	1/2	1/2	0.333	0.600	0.969	0.998	0.688	0.693	0.811	0.811
226	184	pq+r+qr	2	1/2	1/2	0.250	1.000	0.348	0.082	0.681	0.681	0.623	0.747
227	56, 98, 185	1+p+q+pr+pqr	2	5/8	3/8	0.292	1.000	0.592	0.125	0.666	0.666	0.808	0.930
228	172, 202, 216	q+pr+qr	2	1/2	1/2	0.250	1.000	0.497	0.000	0.397	0.010	0.623	0.708
229	74, 88, 173	1+p+pq+r+pqr	2	5/8	3/8	0.292	1.000	0.807	0.154	0.647	0.645	0.808	0.810
230	152, 188, 194	q+r+pqr	2	5/8	3/8	0.292	1.000	0.565	0.120	0.479	0.480	0.808	0.696
231	24, 66, 189	1+p+pq+pr+qr	2	3/4	1/4	0.250	1.000	0.620	0.129	0.469	0.469	0.811	0.696
232		pq+pr+qr	2	1/2	1/2	0.250	1.000	0.638	0.000	0.681	0.002	0.434	0.466
233	104	1+p+q+r+pqr	2	5/8	3/8	0.375	1.000	0.264	0.000	0.298	0.008	0.808	0.719
234	168, 224, 248	pq+r+pqr	1	5/8	3/8	0.208	0.600	0.000	0.000	0.006	0.018	0.308	0.336
235	40, 96, 249	1+p+q+pr+qr	1	3/4	1/4	0.250	0.600	0.000	0.000	0.002	0.006	0.500	0.379
236	200	q+pr+pqr	2	5/8	3/8	0.208	0.600	0.739	0.000	0.681	0.001	0.308	0.308
237	72	1+p+pq+r+qr	2	3/4	1/4	0.250	0.600	0.408	0.000	0.434	0.002	0.500	0.352
238	136, 192, 252	q+r-qr	1	3/4	1/4	0.167	0.600	0.000	0.000	0.002	0.007	0.189	0.130
239	8, 64, 253	1+p+pq+pr+pqr	1	7/8	1/8	0.125	0.600	0.000	0.000	0.001	0.004	0.130	0.000
240	170	p	2	1/2	1/2	0.167	0.600	0.977	0.192	0.693	0.693	0.000	0.000
241	42, 112, 171	1+q+pq+r+pr+qr+pqr	2	5/8	3/8	0.208	0.600	0.879	0.176	0.653	0.653	0.308	0.308
242	162, 176, 186	p+r+pr+qr+pqr	2	5/8	3/8	0.208	0.600	0.687	0.142	0.634	0.634	0.308	0.333
243	34, 48, 187	1+q+pq	2	3/4	1/4	0.167	0.600	0.715	0.152	0.561	0.562	0.189	0.189
244	138, 174, 208	p+q+pq+qr+pqr	2	5/8	3/8	0.208	0.600	0.831	0.166	0.647	0.647	0.308	0.308
245	10, 80, 175	1+r+pr	2	3/4	1/4	0.167	0.600	0.729	0.150	0.556	0.556	0.189	0.189
246	130, 144, 190	p+q+pq+r+pr	2	3/4	1/4	0.250	0.600	0.539	0.120	0.456	0.456	0.500	0.421
247	2, 16, 191	1+qr+pqr	2	7/8	1/8	0.125	0.600	0.516	0.114	0.376	0.376	0.130	0.130
248	168, 224, 234	p+qr+pqr	1	5/8	3/8	0.208	0.400	0.000	0.000	0.004	0.011	0.308	0.336
249	40, 96, 235	1+q+pq+r+pr	1	3/4	1/4	0.250	0.400	0.000	0.000	0.002	0.007	0.500	0.379
250	160	p+r-pr	1	3/4	1/4	0.167	0.400	0.000	0.000	0.002	0.007	0.189	0.130
251	32	1+q+pq+qr+pqr	1	7/8	1/8	0.125	0.400	0.000	0.000	0.001	0.004	0.130	0.041
252	136, 192, 238	p+q-pq	1	3/4	1/4	0.167	0.200	0.000	0.000	0.002	0.007	0.189	0.130
253	8, 64, 239	1+r+pr+qr+pqr	1	7/8	1/8	0.125	0.200	0.000	0.000	0.001	0.005	0.130	0.000
254	128	q+p(-1+q)(-1+r)+r-qr	1	7/8	1/8	0.125	0.000	0.000	0.000	0.001	0.005	0.130	0.041
255	0	1	1	1	0	0.000	0.000	0.000	0.000	0.001	0.003	0.000	0.000

Table A.1 Values of Classification Quantities for ECA. Properties listed are: Rules in the equivalence class; Algebraic form in short notation $x_{i-1}^t = p$, $x_i^t = q$, $x_{i+1}^t = r$, resulting values have to be taken modulo 2, expressions from [100]; Wolfram rule class, values from Wolfram Alpha [259]; Langtons λ with 0 (λ_0) and 1 (λ_1), respectively, as "quiescent" state; μ sensitivity; Kolmogorov complexity of the rule table K_R, of the pattern along the spatial axis (length $L = 100$), averaged over 1100 time steps and 10 samples with random initial conditions, and of the pattern along the temporal axis averaged over the spatial axis with similar parameters as above; Single site entropy H calculated along the spatial H_S and temporal axis H_T, respectively, and averaged over the other, with same parameters as above; Numercial calculation of the the synergy for 1 or 2 time steps, values from M. Bojer [116].

A.3 Fitness Calculation, Evolutionary Algorithm, Consensus Procedure

Algorithm 1 Fitness Calculation (taken from [146])

function $FitnessSingleRule(f, T_{Max}, L, N_{Sample}, Grid_{Target})$ ▷ Fitness of a single rule
  $j \leftarrow 0$
  $F_{total} \leftarrow 0$
  for $j < N_{Sample}$ **do** ▷ Iterate through requested number of samples
    $Grid \leftarrow$ InitialState (L) ▷ Array containing the initial state
    $Grid, t_{SteadyState} \leftarrow$ CA-Simulation $(f, Grid, T_{Max}, L)$
    $F_{total} \leftarrow F_{total} +$ Differences $(Grid, Grid_{Target})$ ▷ Add fitness for current iteration
    $j \leftarrow j + 1$
  $F_{average} \leftarrow F_{total}/N_{Sample}$
  return $Grid$, $t_{SteadyState}$

function $FitnessRuleList(L_f, T_{Max}, L, N_{Sample}, Grid_{Target})$
  $L_{Fitness}$
  for $f\ in\ L_f$ **do**
    Append $(L_{Fitness},$ FitnessSingleRule $(f, T_{Max}, L, N_{Sample}, Grid_{Target}))$
  return $L_{Fitness}$

Algorithm 2 Evolutionary Algorithm Simulation (taken from [146])

$L_f \leftarrow$ randomCA$(N = 100, k = 3)$ ▷ Initial list of 100 random, 3-state rules
function EA-SIMULATION(L_f, $N_{Iteration}$, s)
  $l \leftarrow 0$
  while $l < N_{Iterations}$ **do** ▷ Loop over iterations
    $L_{Fitness} \leftarrow$ FitnessRuleList(L_f)

    Mutate rules and replace if mutated are better
    $L_f^m \leftarrow$ Mutate(Mutate(L_f)) ▷ New List by mutating each rule in the list twice
    $L_{Fitness}^m \leftarrow$ FitnessRuleList(L_f^m)
    $L_f, L_{Fitness} \leftarrow$ SelectFittestPairwise$(L_f, L_f^m, L_{Fitness}, L_{Fitness}^m)$ ▷ New list by selecting pairwise fittest rules

    Replace worst rules by random rules
    $L_f \leftarrow$ Sort$(L_f, L_{Fitness})$ ▷ Sort rules by their fitness
    $L_f \leftarrow$ Delete(L_f, s) ▷ Delete s worst rules from set
    $L_f \leftarrow L_f +$ randomCA$(s, 3)$ ▷ Add s random rules to set

    $l \leftarrow l + 1$
  return L_f

Starting rule	Maximum Dist	Fitness threshold	Consensus Rule
21102222220111012221110000	5	0.9	**21122222220111012211110000**
21102222220111012221110000	5	0.85	2112222222011101122**2**1110000
21102222220111012221110000	5	0.95	21122222220111012211110000
21102222220111012221110000	6	0.9	21122222220111012211110000
21102222220111012221110000	6	0.85	21122222220111012211110000
21102222220111012221110000	6	0.95	2112222222**0**1101122111110000
20022212220211012211110200	5	0.9	2**00**2222**1**22202**2**1101122111110**2**00
20022212220211012211110200	5	0.85	2**00**2222**1**22202**2**1101122111110**2**00
20022212220211012211110200	5	0.95	2**00**2222**1**22202**2**1101122111110**2**00
20022212220211012211110200	6	0.9	2**00**2222**1**22200**1**1011221111102**2**00
20022212220211012211110200	6	0.85	2**00**2222**1**22200**1**1011221111102**2**00
20022212220211012211110200	6	0.95	2**00**2222**1**22202**2**1101122111110000

Table A.2 Consensus Rule for Different Parameters Consensus rule for 2 different starting rules, 2 different neighborhood sizes and 3 different fitness thresholds. All consensus rules differed from the one selected by at most 5 mutations. Taken from [146].

A.4 4 State Rules

A.4.1 Rule Tables

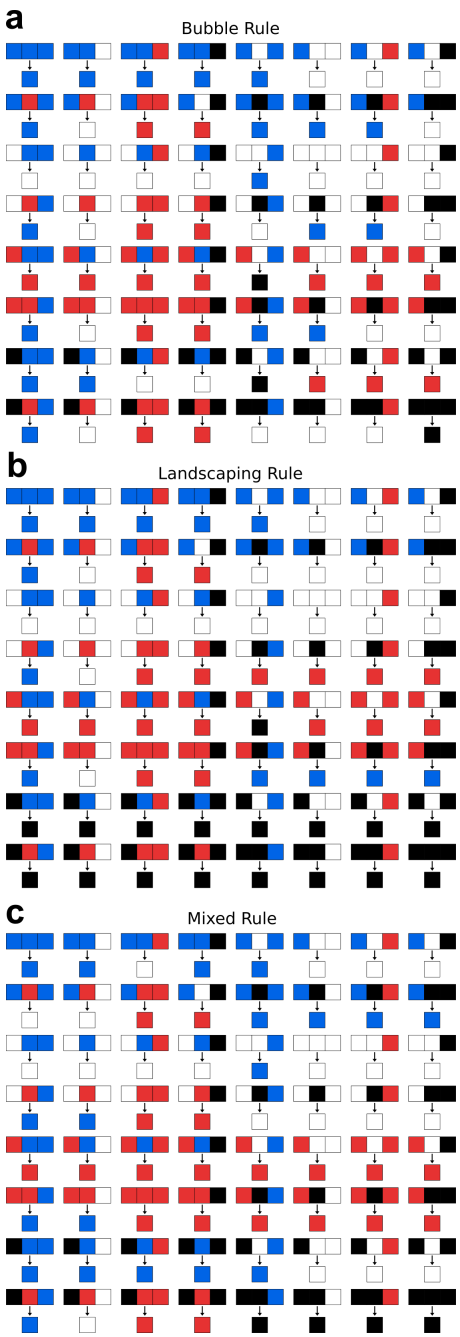


Figure A.3 Rule Table of Bubble, Landscaping, and Mixed Rule

A.4.2 Growth

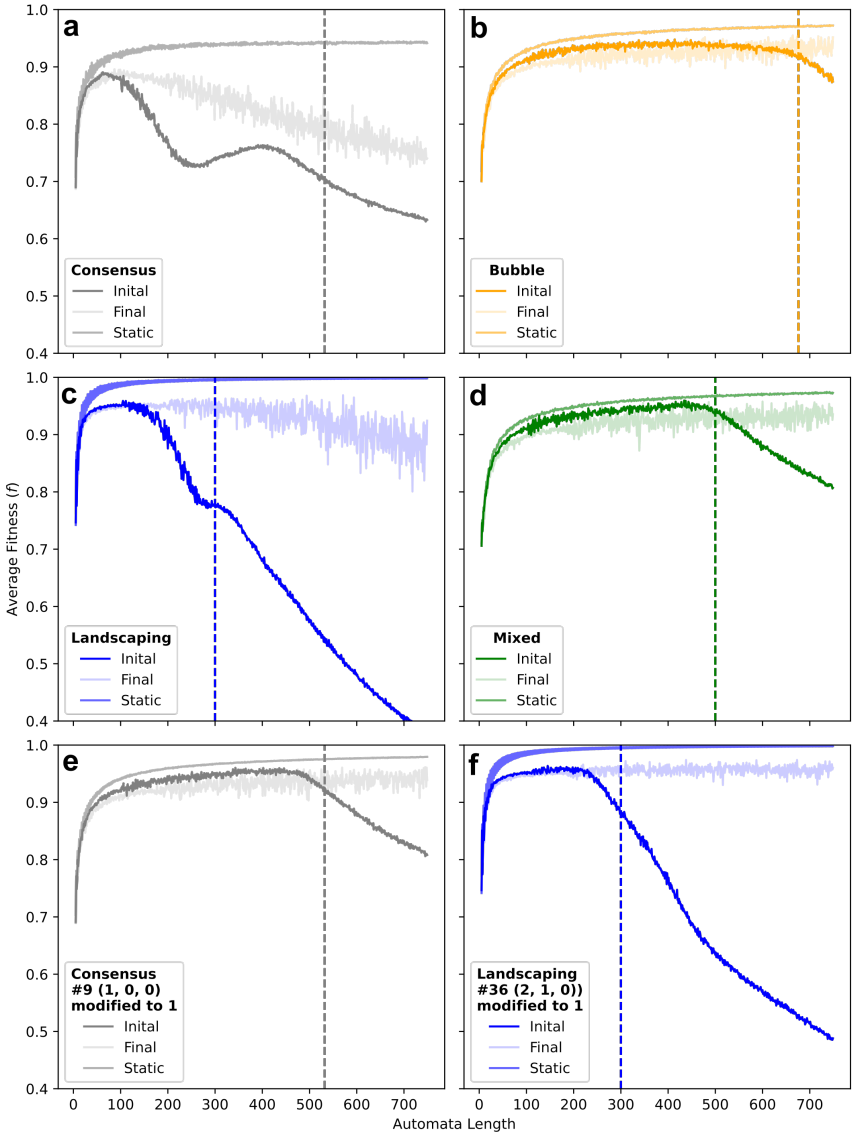


Figure A.4 Fitness During Growth Average fitness Φ as function of the system size L for a growing system for the prototypical rules in 3 and 4 state space, as well as modifications to entries 9 (from 2 to 1) and 36 (from 3 to 1) for the Consensus and Landscaping rule, respectively. The fitness for initial and final length and a static system without growth are depicted. Growth rate was set to a probability of 0.2% per time step and cell. Ensemble sizes for the average are 10000 ($L \leq 10$), 1000 ($10 < L \leq 100$), 100 ($L > 100$) for the growing system and 10000 ($L \leq 10$), 1000 ($L > 100$) for the static system. The dashed vertical lines correspond to the length at which the growth rate of the system exceeds the speed of the pattern formation process, defined by the equation $0.002aL_{crit} = 1$, with a being the average number of steps needed to complete pattern formation.

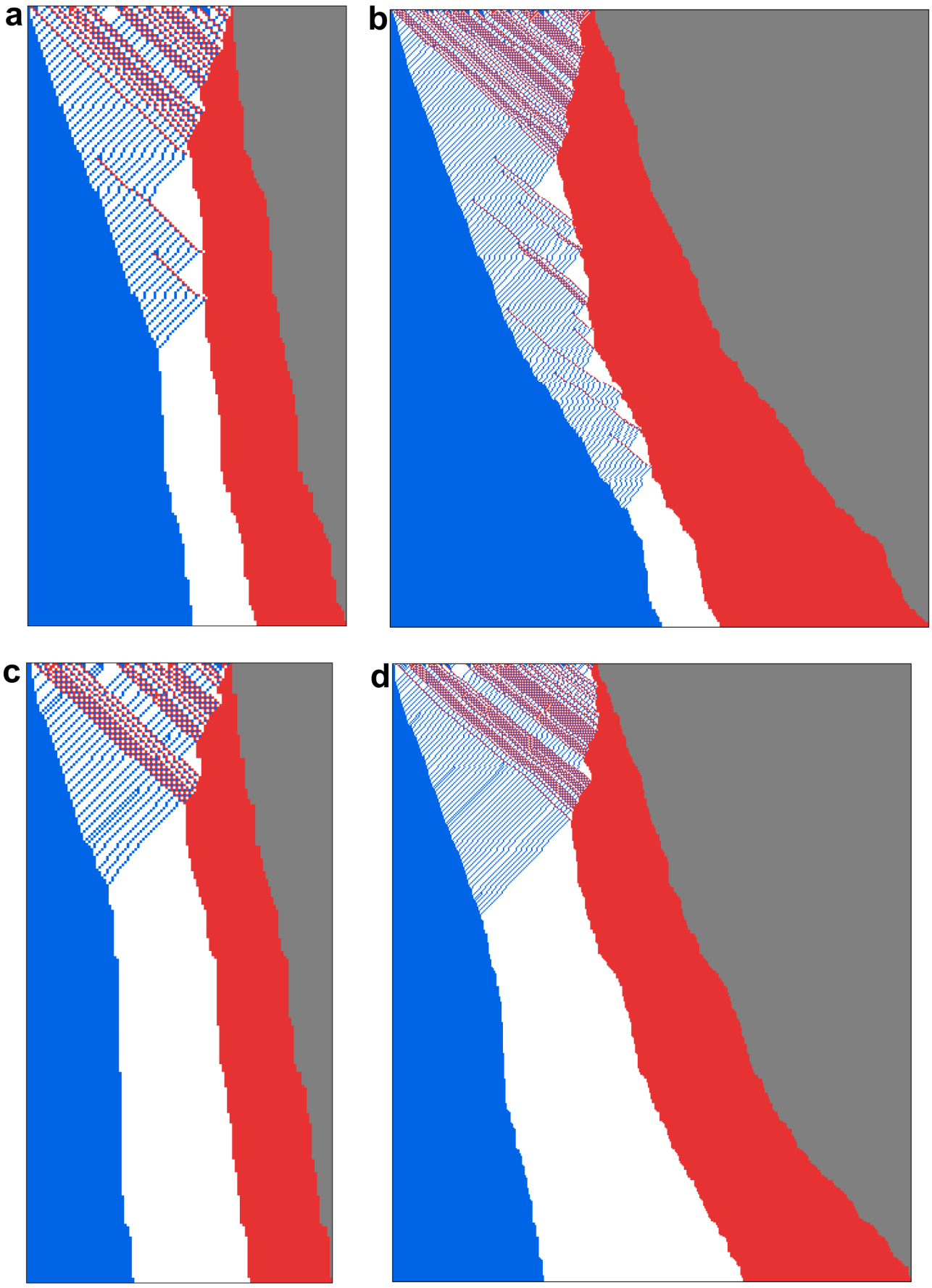


Figure A.5 Kymographs for Consensus Rule During Growth. Kymograph for a growing system for the Consensus rule (a, b) and modified Consensus rule modified at position 9 (to 1) (c, d) for initial sizes $L = 80$ (a, c) and $L = 160$ (b, d) with growth rate of 0.2% per cell and time step.

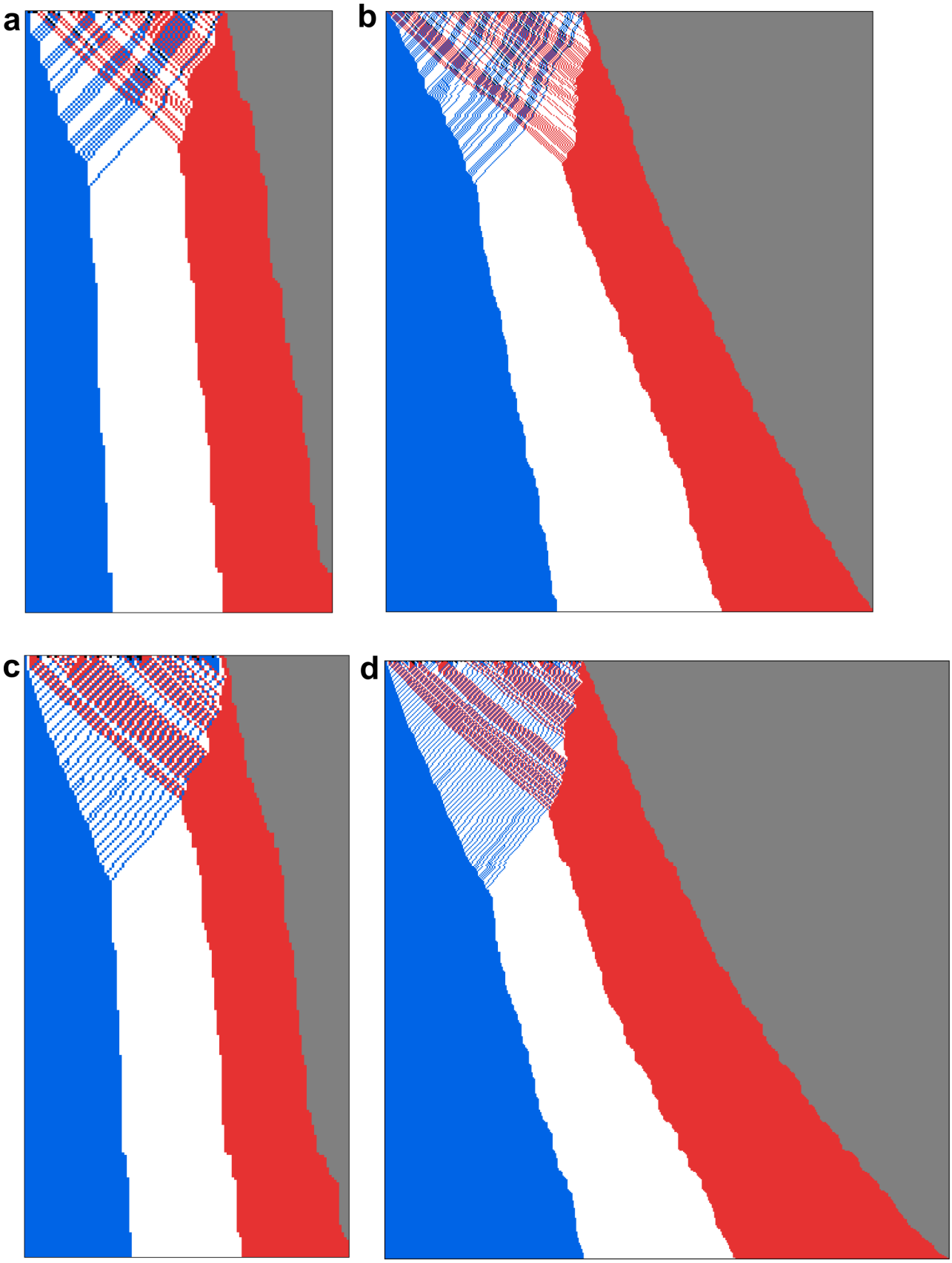


Figure A.6 Kymographs for Bubble and Mixed Rule During Growth. Kymograph for a growing system for the Mixed rule (a, b) and Bubble rule (c, d) for initial sizes $L = 80$ (a, c) and $L = 160$ (b, d) with growth rate of 0.2% per cell and time step.

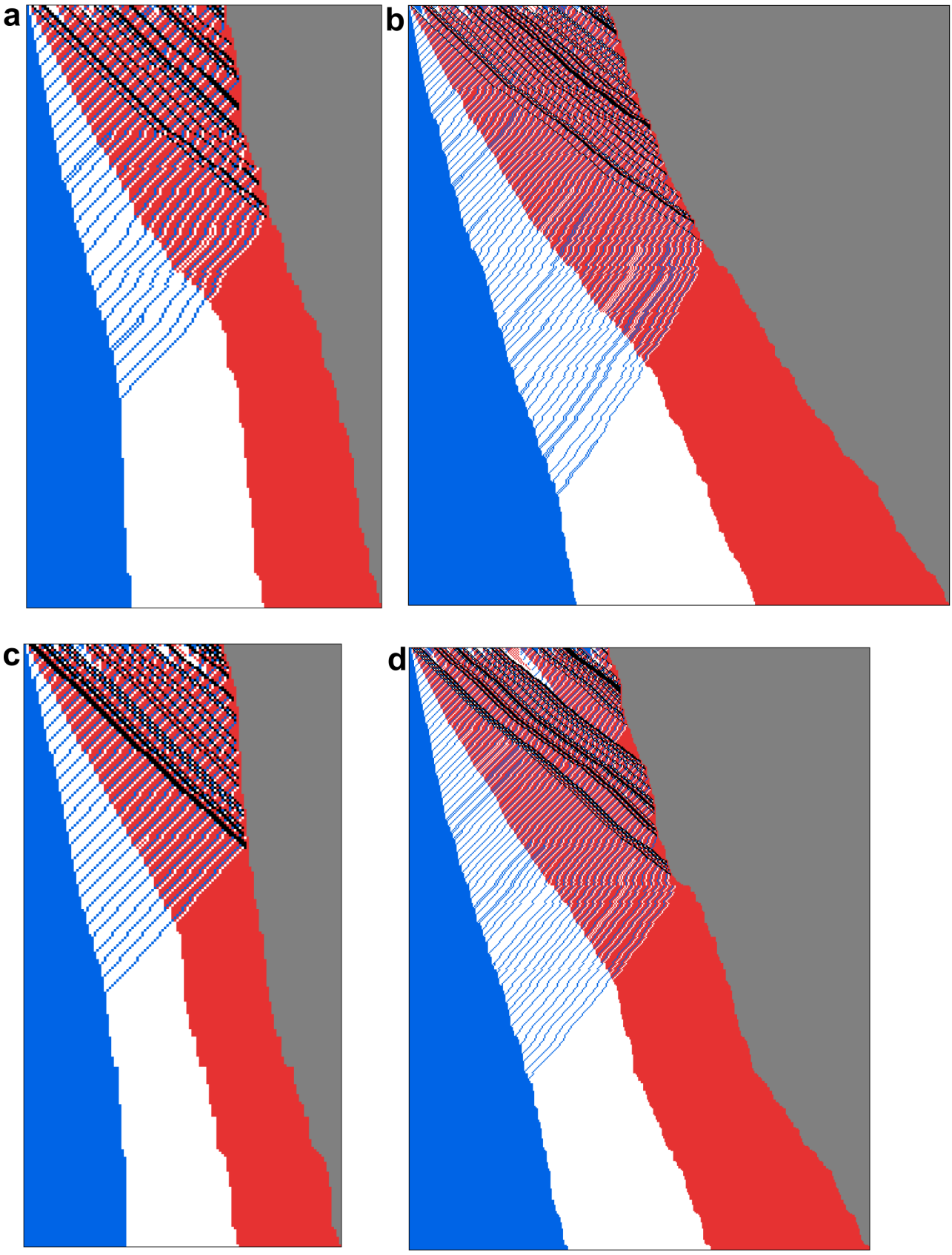
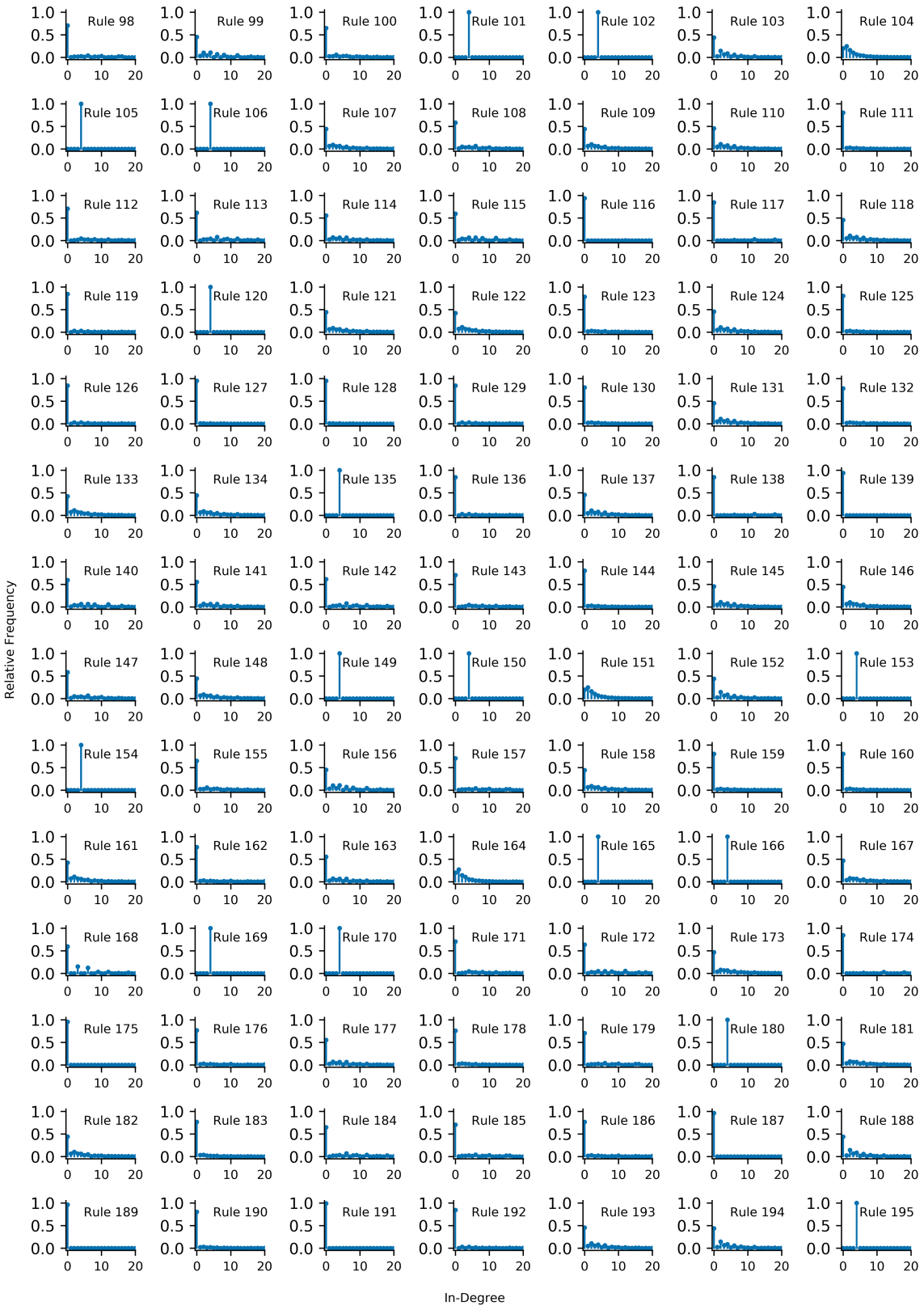


Figure A.7 Kymographs for Landscaping Rule During Growth. Kymograph for a growing system for the Landscaping rule (a, b) and modified Landscaping rule modified at position 36 (to 1) (c, d) for initial sizes $L = 80$ (a, c) and $L = 160$ (b, d) with growth rate of 0.2% per cell and time step.

A.5 Programmability

A.5.1 In-Degree Distributions


Rule 98
Rule 99
Rule 100
Rule 101
Rule 102
Rule 103
Rule 104
Rule 105
Rule 106
Rule 107
Rule 108
Rule 109
Rule 110
Rule 111
Rule 112
Rule 113
Rule 114
Rule 115
Rule 116
Rule 117
Rule 118
Rule 119
Rule 120
Rule 121
Rule 122
Rule 123
Rule 124
Rule 125
Rule 126
Rule 127
Rule 128
Rule 129
Rule 130
Rule 131
Rule 132
Rule 133
Rule 134
Rule 135
Rule 136
Rule 137
Rule 138
Rule 139
Rule 140
Rule 141
Rule 142
Rule 143
Rule 144
Rule 145
Rule 146
Rule 147
Rule 148
Rule 149
Rule 150
Rule 151
Rule 152
Rule 153
Rule 154
Rule 155
Rule 156
Rule 157
Rule 158
Rule 159
Rule 160
Rule 161
Rule 162
Rule 163
Rule 164
Rule 165
Rule 166
Rule 167
Rule 168
Rule 169
Rule 170
Rule 171
Rule 172
Rule 173
Rule 174
Rule 175
Rule 176
Rule 177
Rule 178
Rule 179
Rule 180
Rule 181
Rule 182
Rule 183
Rule 184
Rule 185
Rule 186
Rule 187
Rule 188
Rule 189
Rule 190
Rule 191
Rule 192
Rule 193
Rule 194
Rule 195
Relative Frequency
In-Degree

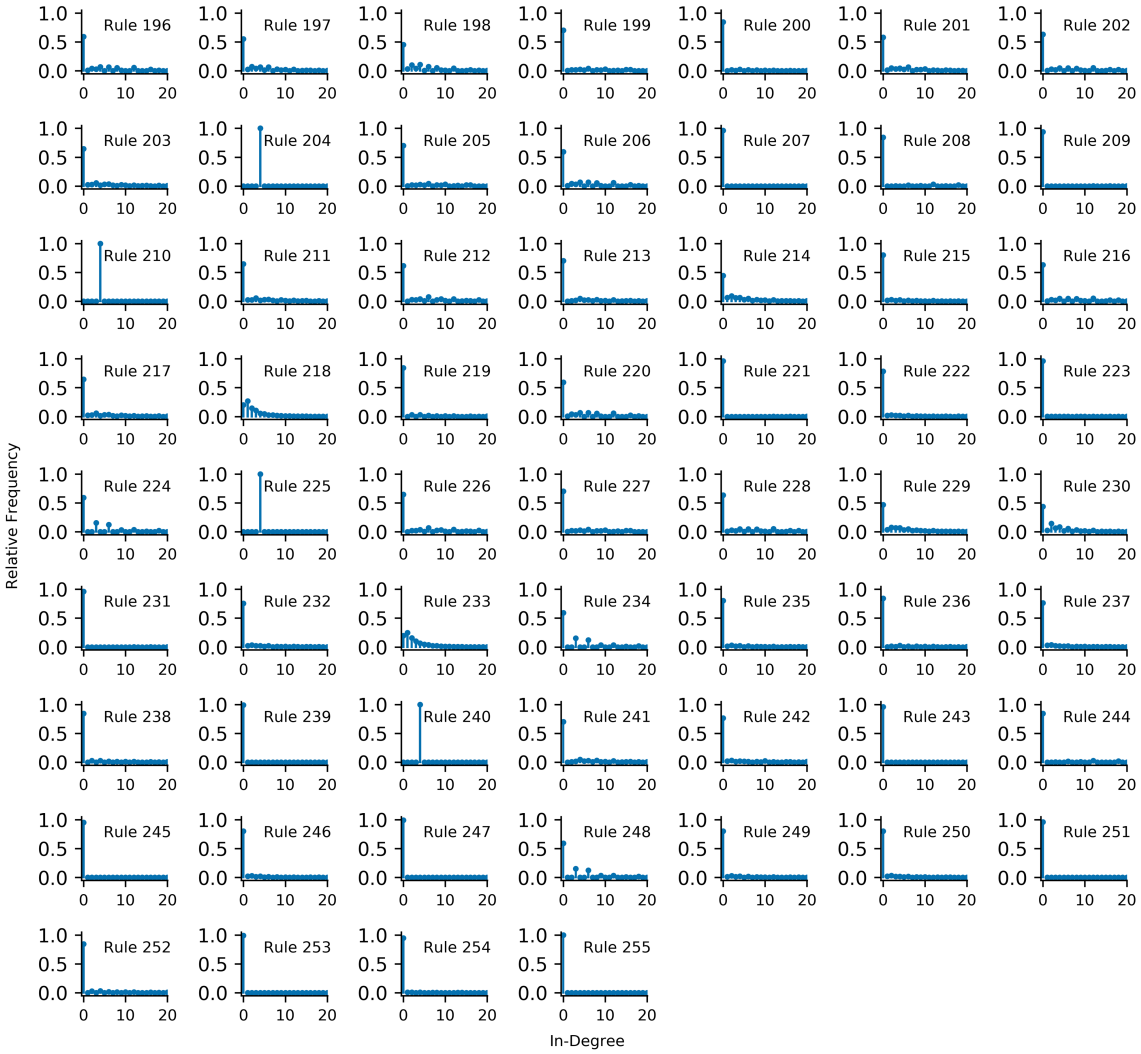


Figure A.8 In Degree Distributions. In degree distributions for linear topology with two organizer cells and system size $L = 16$. Distributions are cut at 20. *Reproduced from [168] under the Creative Commons Attribution 4.0 International License [170].*

A.6 Synchronization

A.6.1 Synchronization Schemes

Exemplary list of (a)synchronous CA update schemes compiled from [193, 194].

- **Synchronous Update**: All cells are update at the same time [193].
- **Random Independent**: At each step in time a cell is chosen at random and updated [193].
- **Random Order**: All cells are updated in random order. After each cell has performed an update, the order is again randomly chosen [193].
- **Cyclic-sequential ordered**: The update order is fixed in the automaton definition and cycled through until all cells have updated once, then repeated [193].
- **Generic cyclic-sequential ordered**: As above, but now the update can take some time until it is communicated to the neighbors, maximally until the cell updates again [193].
- **Clocked schemes**: Timers are associated with each cell such that they update autonomously and at possibly different and possibly variable frequencies and with different delays until the update is communicated to the neighbors. [193].
- α**-Synchronism**: At each time-step a subset of cells is chosen, each cell is with probability α a member of this subset, and then updated [194].
- β**-Synchronism**: Each cell is updated at the same time step, but the information about that update is transmitted to the whole neighborhood only with probability β [194].
- γ**-synchronism**: Each cell is updated at the same time step, but the information about that update is transmitted to each neighbor with probability γ [194].

A.7 Generation of Irregular Grids

The goal of this section is to describe an algorithm to generate 2D irregular grids whose irregularity can smoothly be tuned with a parameter r, including regular grids for certain parameter values and grids which closely resemble neighborhood configurations observed experimentally in quasi 2D tissue. To construct a grid of $M \cdot N$ points on a surface with dimensions L_x and L_y, the points are first distributed according to the midpoints a $M \cdot N$ hexagonal pattern. Then each point is shifted by δx and δy, both random variables drawn from a triangular distribution with zero mean and standard deviation r. Periodic boundary conditions are assumed for this as well as for the rest of the procedure. To construct neighborhood relationships for each point, a Voronoi tessellation using the SciPy python package [201] is performed. The edge and area distributions as well as exemplary tessellations for different values of the standard deviation r are shown in fig. A.9. As expected, both distributions as well as the visual appearance of the grids tend to a more diverse phenomenology for increasing r. For large enough r, both distributions come close to the experimental distributions for the *Drosophila* wing epithelium extracted from Farhadifar et al. [220], indicating that the procedure can be used as a proxy to generate grids which are statistically approximate representation of specific experimental tissues.

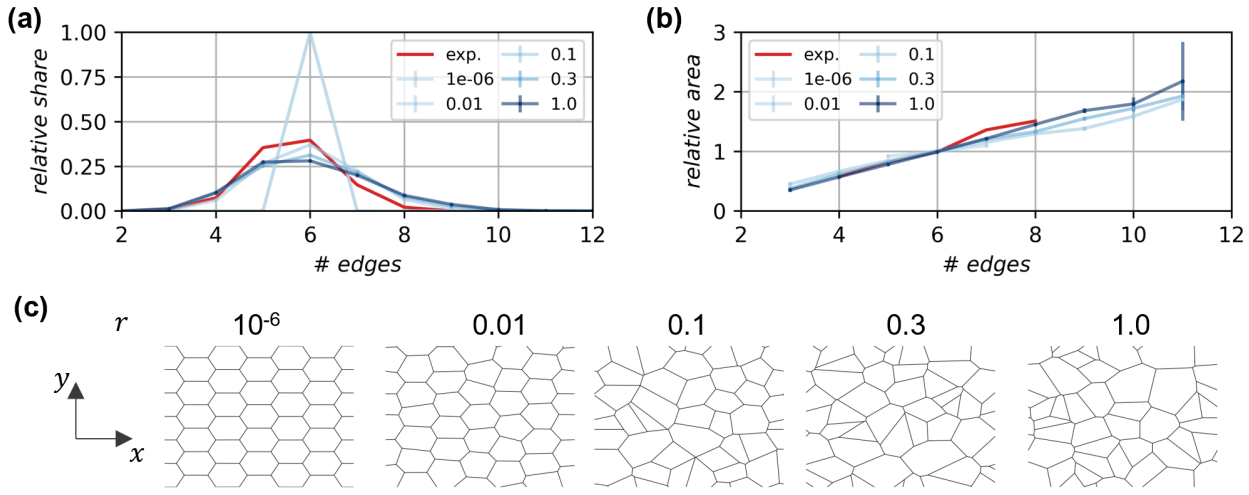


Figure A.9 In Silico Generated Artificial Grids. Grids are generated by placing midpoints of cells according to a regular hexagonal grid, shifting them by adding independent random variables drawn from a triangular distribution with zero mean and standard deviations r to the midpoint coordinates, and subsequently performing a Voronoi tessellation [201]. Examples are shown for a Grid of $M \cdot N = 6 \cdot 6 = 36$ cells on a surface of size $L_x = L_y = 1$ with periodic boundary conditions and an ensemble size of 100. (a) Frequency distribution of the number of edges and relative area distribution $\frac{\langle A_{\#edges} \rangle_{Ensemble}}{\langle A \rangle_{Ensemble,\ \#Edges}}$ of the generated grids different standard deviations r, together with the experimental distribution for the *Drosophila* wing epithelium extracted from Farhadifar et al. [220]. (c) Exemplary grids for different standard deviations r of the triangular distribution. Visualization procedure adapted from the corresponding SciPy routine [201].

B Acknowledgements

First and foremost I want to thank *Uli* for his excellent supervision of my thesis. Your dedication to teaching your students reflected in your open mind toward the ideas of your students and your support when we pursued them. Your office door was always open to me, and an idea always left your office better than it went in.

Second, I want to thank my fellow PhD students for the great time at the chair, the fruitful scientific discussions, interesting conversations during lunch and fun on group retreats.

This thesis would probably not exist in this form without *Mareike*, who accompanied me during this PhD journey from the beginning to the end with countless discussions, numerous invaluable feedback and great camaraderie. I will dearly miss our fun meetings and the stories of your thrilling travel adventures.

Nanni, you are the heart and soul of the group, a constant source of positivity, helpful feedback and new ideas.

Tobias, I salute the intensity and personal investment you put into creating a beautiful, complex model and code package and going all in in your research about the origins of life.

Joachim, your ability to repair cars yourself while being a theoretical physicist was always fascinating to me.

Johannes, your open mind and vision when discussing scientific questions in a multitude of contexts and your ability to crunch pages of equations I always admired deeply.

If you wanted to get a scientific reality check, a simple talk about your research with *Severin* and *Florian* was usually enough. Thank you for your critical thinking and numerous feedback.

Zara and *Elena* I admired for keeping me humble towards the hardships an experimentalist in the biological sciences endures.

Michael I thank for showcasing me how to work in an extremely structured and well-organized manner.

Julio, seldomly I had the pleasure to listen to more enjoyable talks than your presentations.

My ability to throw frisbees improved greatly when *Johannes* joined the group – I dearly miss our regular small sport breaks. Your broad knowledge and interest together with your quick thinking made all our conversation a delight for me. I especially cherished the book club you organized together with *Katharina*.

If anyone needs an example of how to successfully deal with great changes in working mindset when switching fields from biology to physics, look no further than *Frauke*.

Ludwig, I admire you for setting a high standard producing scientific ideas and results as well as summoning the support of others for your cause. Together with Frauke you elevated the mood of any room you two went into.

Mariel, *Zoë*, *Sebastian*, *Ludwig*, *Yasmin* and *Sabina* I want to thank for being awesome office companions. I particularly enjoyed our discussions about science while sharing the same “thinking cell”.

A shout-out to the PostDoc Squad of our Chair - *Johannes*, *Vladimir*, *Filipe*, *Maryam*, *Bernhard*, and *Hamid*. Your feedback during my talks and your perspective on and knowledge about science was always incredibly helpful and valuable. Especially I want to thank *Bernhard* for elevating my understanding of statistical physics to a new level.

I want to thank all the students I had the pleasure of supervising. *Zoë, Yasmin, Ludwig, Gabriel, Max, Sabina, Maria, Lukas, Valèria*. Thank you for your hard work, perseverance, and the fun time while accompanying you on a small part of your journey in the scientific world. I hope you could learn as much from

me as I could from you.

I thank *Mareike*, *Giovanni*, *Ludwig*, *Frauke*, *Johannes*, and *Stephanie* for critical proofreading of this thesis.

Finally, I want to thank *Stephanie* for her loving support. Her dedication to science and her rigorosity in holding herself accountable to the scientific standards have always been a model to me.

Bibliography

[1] Ron Sender, Shai Fuchs, and Ron Milo. Revised estimates for the number of human and bacteria cells in the body. *PLoS Biol.*, 14(8):e1002533, 2016. doi: 10.1371/journal.pbio.1002533.

[2] Lewis Wolpert. *Principles of development*. Oxford University Press, Oxford, 5. ed. edition, 2015. ISBN 9780198709886.

[3] Scott F. Gilbert. *Developmental biology*. Sinauer Associates, Sunderland, Mass., 10. ed. edition, 2014. ISBN 978-0878939787.

[4] Sarah J. Bray. Notch signalling in context. *Nat. Rev. Mol. Cell Biol.*, 17(11):722–735, 2016. doi: 10.1038/nrm.2016.94.

[5] Jonathan M. W. Slack. *Essential developmental biology*. Blackwell Publishing, Malden, Mass., 2. ed. edition, 2006. ISBN 978-1-4051-2216-0.

[6] Peter Nick, Reinhard Fischer, Dietmar Gradl, Mathias Gutmann, Jörg Kämper, Tilman Lamparter, and Michael Riemann. *Modellorganismen*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2019. ISBN 978-3-662-54867-7. doi: 10.1007/978-3-662-54868-4.

[7] Michael J.F. Barresi and Scott F. Gilbert. *Developmental Biology*. Oxford University Press, New York, 12 (int. ed.) edition, 2020. ISBN 9781605358741.

[8] Lewis Wolpert, Cheryll Tickle, Alfonso Martinez Arias, Peter Lawrence, and James Locke. *Principles of Development*. Oxford University Press, New York, 6 edition, 2019. ISBN 978-0-19-880056-9.

[9] Camilo Mora, Derek P. Tittensor, Sina Adl, Alastair G. B. Simpson, and Boris Worm. How many species are there on earth and in the ocean? *PLoS Biol.*, 9(8):e1001127, 2011. doi: 10.1371/journal.pbio.1001127.

[10] Norbert Perrimon, Chrysoula Pitsouli, and Ben-Zion Shilo. Signaling mechanisms controlling cell fate and embryonic patterning. *Cold Spring Harb. Perspect. Biol.*, 4(8):a005975, 2012. doi: 10.1101/cshperspect.a005975.

[11] Claudia Gerri, Sergio Menchero, Shantha K. Mahadevaiah, James M. A. Turner, and Kathy K. Niakan. Human embryogenesis: A comparative perspective. *Annu. Rev. Cell Dev.*, 36:411–440, 2020. ISSN 1081-0706. doi: 10.1146/annurev-cellbio-022020-024900.

[12] Francis Corson, Lydie Couturier, Herve Rouault, Khalil Mazouni, and Francois Schweisguth. Self-organized notch dynamics generate stereotyped sensory organ patterns in drosophila. *Science*, 356 (6337), 2017. ISSN 0036-8075. doi: 10.1126/science.aai7407.

[13] Andrew C. Oates, Luis G. Morelli, and Saul Ares. Patterning embryos with oscillations: structure, function and dynamics of the vertebrate segmentation clock. *Development*, 139(4):625–639, 2012. ISSN 0950-1991. doi: 10.1242/dev.063735.

[14] Arthur D. Lander. How cells know where they are. *Science*, 339(6122):923–927, 2013. doi: 10.1126/science.1224186.

[15] Jamie A. Davies and Fokion Glykofrydis. Engineering pattern formation and morphogenesis. *Biochem. Soc. Trans.*, 48(3):1177–1185, 2020. doi: 10.1042/BST20200013.

[16] Lukas Aufinger and Friedrich C. Simmel. Establishing communication between artificial cells. *Chemistry*, 25(55):12659–12670, 2019. doi: 10.1002/chem.201901726.

[17] Katherine W. Rogers and Alexander F. Schier. Morphogen gradients: from generation to interpretation. *Annu. Rev. Cell Dev.*, 27:377–407, 2011. ISSN 1081-0706. doi: 10.1146/annurev-cellbio-092910-154148.

[18] A. M. Turing. The chemical basis of morphogenesis. *Philos. Trans. R. Soc. Lond., B, Biol. Sci.*, 237 (641):37–72, 1952. ISSN 0962-8436. doi: 10.1098/rstb.1952.0012.

[19] Lewis Wolpert. The french flag problem: a contribution to the discussion on pattern development and regulation. In Conrad Hal Waddington, editor, *Towards a Theoretical Biology*, pages 125–133. Aldine Pub. Co., 1968.

[20] L. Wolpert. Positional information and the spatial pattern of cellular differentiation. *J. Theor. Biol.*, 25(1):1–47, 1969. ISSN 00225193. doi: 10.1016/S0022-5193(69)80016-0.

[21] Matthew P. Harris, Scott Williamson, John F. Fallon, Hans Meinhardt, and Richard O. Prum. Molecular evidence for an activator-inhibitor mechanism in development of embryonic feather branching. *PNAS*, 102(33):11734–11739, 2005. ISSN 0027-8424. doi: 10.1073/pnas.0500781102.

[22] J. Raspopovic, L. Marcon, L. Russo, and J. Sharpe. Modeling digits. digit patterning is controlled by a bmp-sox9-wnt turing network modulated by morphogen gradients. *Science*, 345(6196):566–570, 2014. ISSN 0036-8075. doi: 10.1126/science.1252960.

[23] Marcin Zagorski, Yoji Tabata, Nathalie Brandenberg, Matthias P. Lutolf, Gašper Tkačik, Tobias Bollenbach, James Briscoe, and Anna Kicheva. Decoding of position in the developing neural tube from antiparallel morphogen gradients. *Science*, 356(6345):1379–1383, 2017. doi: 10.1126/science.aam5887.

[24] Mariela D. Petkova, Gašper Tkačik, William Bialek, Eric F. Wieschaus, and Thomas Gregor. Optimal decoding of cellular identities in a genetic network. *Cell*, 176(4):844–855.e15, 2019. ISSN 0092-8674. doi: 10.1016/j.cell.2019.01.007.

[25] Gašper Tkačik and Thomas Gregor. The many bits of positional information. *Development*, 148(2), 2021. doi: 10.1242/dev.176065.

[26] Aurore Dupin and Friedrich C. Simmel. Signalling and differentiation in emulsion-based multicompartmentalized in vitro gene circuits. *Nat. Chem*, 11(1):32–39, 2019. doi: 10.1038/s41557-018-0174-9.

[27] Marcelo Boareto. Patterning via local cell-cell interactions in developing systems. *Dev. Biol.*, 2019. ISSN 0012-1606. doi: 10.1016/j.ydbio.2019.12.008.

[28] Marika Sjöqvist and Emma R. Andersson. Do as i say, not(ch) as i do: Lateral control of cell fate. *Dev. Biol.*, 447(1):58–70, 2019. ISSN 0012-1606. doi: 10.1016/j.ydbio.2017.09.032.

[29] Bo-Kai Liao and Andrew C. Oates. Delta-notch signalling in segmentation. *Arthropod Struct Dev .*, 46(3):429–447, 2017. ISSN 1873-5495. doi: 10.1016/j.asd.2016.11.007.

[30] Leonardo Morsut, Kole T. Roybal, Xin Xiong, Russell M. Gordley, Scott M. Coyle, Matthew Thomson, and Wendell A. Lim. Engineering customized cell sensing and response behaviors using synthetic notch receptors. *Cell*, 164(4):780–791, 2016. ISSN 0092-8674. doi: 10.1016/j.cell.2016.01.012.

[31] Satoshi Toda, Lucas R. Blauch, Sindy K. Y. Tang, Leonardo Morsut, and Wendell A. Lim. Programming self-organizing multicellular structures with synthetic cell-cell signaling. *Science*, 361(6398): 156–162, 2018. ISSN 0036-8075. doi: 10.1126/science.aat0271.

[32] Satoshi Toda, Wesley L. McKeithan, Teemu J. Hakkinen, Pilar Lopez, Ophir D. Klein, and Wendell A. Lim. Engineering synthetic morphogen systems that can program multicellular patterning. *Science*, 370(6514):327–331, 2020. doi: 10.1126/science.abc0033.

[33] Terren K. Niethamer and Jeffrey O. Bush. Getting direction(s): The eph/ephrin signaling system in cell positioning. *Dev. Biol.*, 447(1):42–57, 2019. ISSN 0012-1606. doi: 10.1016/j.ydbio.2018.01.012.

[34] Cyrille Alexandre, Alberto Baena-Lopez, and Jean-Paul Vincent. Patterning and growth control by membrane-tethered wingless. *Nature*, 505(7482):180–185, 2014. ISSN 0028-0836. doi: 10.1038/nature12879.

[35] Daniel A. Goodenough and David L. Paul. Gap junctions. *Cold Spring Harb. Perspect. Biol.*, 1(1): a002576, 2009. ISSN 1943-0264. doi: 10.1101/cshperspect.a002576.

[36] Alex Mogilner, Roy Wollman, and Wallace F. Marshall. Quantitative modeling in cell biology: what is it good for? *Dev. Cell*, 11(3):279–287, 2006. ISSN 15345807. doi: 10.1016/j.devcel.2006.08.004.

[37] Pascal O. Luthi, Bastien Chopard, Anette Preiss, and Jeremy J. Ramsden. A cellular automaton model for neurogenesis in drosophila. *Phys. D: Nonlinear Phenom.*, 118(1-2):151–160, 1998. ISSN 01672789. doi: 10.1016/S0167-2789(98)00008-6.

[38] Chad M. Glen, Melissa L. Kemp, and Eberhard O. Voit. Agent-based modeling of morphogenetic systems: Advantages and challenges. *PLoS Comput. Biol.*, 15(3):e1006577, 2019. ISSN 1553-734X. doi: 10.1371/journal.pcbi.1006577.

[39] Stanislaw Ulam et al. On some mathematical problems connected with patterns of growth of figures. In *Proceedings of Symposia in Applied Mathematics*, volume 14, pages 215–224. Am. Math. Soc. Vol. 14, Providence, 1962.

[40] János Neumann, Arthur W Burks, et al. *Theory of self-reproducing automata*, volume 1102024. University of Illinois press Urbana, 1966.

[41] John Conway. Mathematical games (m. gardner, ed.). *Sci. Am.*, 223(4), 1970.

[42] Stephen Wolfram. Statistical mechanics of cellular automata. *Rev. Mod. Phys.*, 55(3):601–644, 1983. ISSN 0034-6861. doi: 10.1103/RevModPhys.55.601.

[43] Stephen Wolfram. Universality and complexity in cellular automata. *Phys. D: Nonlinear Phenom.*, 10(1-2):1–35, 1984. ISSN 01672789. doi: 10.1016/0167-2789(84)90245-8.

[44] Stephen Wolfram. Computation theory of cellular automata. *Commun. Math. Phys.*, 96(1):15–57, 1984. ISSN 0010-3616. doi: 10.1007/BF01217347.

[45] Michael J. Casey, Patrick S. Stumpf, and Ben D. MacArthur. Theory of cell fate. *Wiley Interdiscip Rev Syst Biol Med*, 12(2):e1471, 2020. doi: 10.1002/wsbm.1471.

[46] Enzo Kopperger, Jonathan List, Sushi Madhira, Florian Rothfischer, Don C. Lamb, and Friedrich C. Simmel. A self-assembled nanoscale robotic arm controlled by electric fields. *Science*, 359(6373): 296–301, 2018. doi: 10.1126/science.aao4284.

[47] Friedrich C. Simmel. Crc 1032 report a02 sensing and control with nucleic acid hybrid nanoactuators. 2020.

[48] Friedrich C. (personal communication) Simmel. Emulating ca with nanorobotic arms, 2020.

[49] V. C. Luca, K. M. Jude, and K. C. Garcia. Complex of notch1 (egf11-13) bound to delta-like 4 (n-egf2), 2015.

[50] Vincent C. Luca, Kevin M. Jude, Nathan W. Pierce, Maxence V. Nachury, Suzanne Fischer, and K. Christopher Garcia. Structural biology. structural basis for notch1 engagement of delta-like 4. *Science*, 347(6224):847–853, 2015. ISSN 0036-8075. doi: 10.1126/science.1261093.

[51] William Humphrey, Andrew Dalke, and Klaus Schulten. Vmd: Visual molecular dynamics. *J. Mol. Graph.*, 14(1):33–38, 1996. ISSN 02637855. doi: 10.1016/0263-7855(96)00018-5.

[52] Federico Bocci, José Nelson Onuchic, and Mohit Kumar Jolly. Understanding the principles of pattern formation driven by notch signaling by integrating experiments and theoretical models. *Front. Physiol.*, 11:929, 2020. ISSN 1664-042X. doi: 10.3389/fphys.2020.00929.

[53] Brendan D'Souza, Laurence Meloty-Kapella, and Gerry Weinmaster. Canonical and non-canonical notch ligands. *Curr. Top. Dev. Biol.*, 92:73–129, 2010. doi: 10.1016/S0070-2153(10)92003-6.

[54] Humberto Contreras-Cornejo, Germán Saucedo-Correa, Javier Oviedo-Boyso, Juan José Valdez-Alarcón, Víctor Manuel Baizabal-Aguirre, Marcos Cajero-Juárez, and Alejandro Bravo-Patiño. The csl proteins, versatile transcription factors and context dependent corepressors of the notch signaling pathway. *Cell Div.*, 11:12, 2016. ISSN 1747-1028. doi: 10.1186/s13008-016-0025-2.

[55] David del Álamo, Hervé Rouault, and François Schweisguth. Mechanism and significance of cis-inhibition in notch signalling. *Curr. Biol.*, 21(1):R40–7, 2011. ISSN 0960-9822. doi: 10.1016/j.cub.2010.10.034.

[56] Michael Cohen, Marios Georgiou, Nicola L. Stevenson, Mark Miodownik, and Buzz Baum. Dynamic filopodia transmit intermittent delta-notch signaling to drive pattern refinement during lateral inhibition. *Dev. Cell*, 19(1):78–89, 2010. ISSN 15345807. doi: 10.1016/j.devcel.2010.06.006.

[57] Mohit Kumar Jolly, Marcelo Boareto, Mingyang Lu, Jose' N. Onuchic, Cecilia Clementi, and Eshel Ben-Jacob. Operating principles of notch–delta–jagged module of cell–cell communication. *New J. Phys.*, 17(5):055021, 2015. doi: 10.1088/1367-2630/17/5/055021.

[58] S. Artavanis-Tsakonas, M. D. Rand, and R. J. Lake. Notch signaling: cell fate control and signal integration in development. *Science*, 284(5415):770–776, 1999. ISSN 0036-8075. doi: 10.1126/science.284.5415.770.

[59] Francesca A. Carrieri and Jacqueline Kim Dale. Turn it down a notch. *Front. Cell Dev. Biol.*, 4:151, 2016. ISSN 2296-634X. doi: 10.3389/fcell.2016.00151.

[60] Juan J. Sanz-Ezquerro, Andrea E. Münsterberg, and Sigmar Stricker. Editorial: Signaling pathways in embryonic development. *Front. Cell Dev. Biol.*, 5:76, 2017. ISSN 2296-634X. doi: 10.3389/fcell.2017.00076. URL `https://directory.doabooks.org/handle/20.500.12854/59364`.

[61] David Sprinzak, Amit Lakhanpal, Lauren LeBon, Leah A. Santat, Michelle E. Fontes, Graham A. Anderson, Jordi Garcia-Ojalvo, and Michael B. Elowitz. Cis-interactions between notch and delta generate mutually exclusive signalling states. *Nature*, 465(7294):86–90, 2010. ISSN 0028-0836. doi: 10.1038/nature08959.

[62] Oren Shaya, Udi Binshtok, Micha Hersch, Dmitri Rivkin, Sheila Weinreb, Liat Amir-Zilberstein, Bassma Khamaisi, Olya Oppenheim, Ravi A. Desai, Richard J. Goodyear, Guy P. Richardson, Christopher S. Chen, and David Sprinzak. Cell-cell contact area affects notch signaling and notch-dependent patterning. *Dev. Cell*, 40(5):505–511.e6, 2017. ISSN 15345807. doi: 10.1016/j.devcel.2017.02.009.

[63] Tomer Yaron, Yossi Cordova, and David Sprinzak. Juxtacrine signaling is inherently noisy. *Biophys. J.*, 107(10):2417–2424, 2014. doi: 10.1016/j.bpj.2014.10.006.

[64] Eial Teomy, David A. Kessler, and Herbert Levine. Ordered hexagonal patterns via notch-delta signaling. *Phys. Biol.*, 18(6), 2021. doi: 10.1088/1478-3975/ac28a4.

[65] Ryoichiro Kageyama, Toshiyuki Ohtsuka, Hiromi Shimojo, and Itaru Imayoshi. Dynamic notch signaling in neural progenitor cells and a revised view of lateral inhibition. *Nat. Neurosci*, 11(11): 1247–1251, 2008. doi: 10.1038/nn.2208.

[66] Nagarajan Nandagopal, Leah A. Santat, Lauren LeBon, David Sprinzak, Marianne E. Bronner, and Michael B. Elowitz. Dynamic ligand discrimination in the notch signaling pathway. *Cell*, 172(4): 869–880.e19, 2018. ISSN 0092-8674. doi: 10.1016/j.cell.2018.01.002.

[67] Pulin Li and Michael B. Elowitz. Communication codes in developmental signaling pathways. *Development*, 146(12), 2019. doi: 10.1242/dev.170977.

[68] Ranjith Viswanathan, Aleksandar Necakov, Mateusz Trylinski, Rohit Krishnan Harish, Daniel Krueger, Emilia Esposito, Francois Schweisguth, Pierre Neveu, and Stefano de Renzis. Optogenetic inhibition of delta reveals digital notch signalling output during tissue differentiation. *EMBO Rep.*, 20(12):e47999, 2019. doi: 10.15252/embr.201947999.

[69] Jordi Cayuso, Aliaksandr Dzementsei, Johanna C. Fischer, Gopal Karemore, Sara Caviglia, Josefin Bartholdson, Gavin J. Wright, and Elke A. Ober. Ephrinb1/ephb3b coordinate bidirectional epithelial-mesenchymal interactions controlling liver morphogenesis and laterality. *Dev. Cell*, 39(3):316–328, 2016. ISSN 15345807. doi: 10.1016/j.devcel.2016.10.009.

[70] Jingyi Gong, Roman Körner, Louise Gaitanos, and Rüdiger Klein. Exosomes mediate cell contact-independent ephrin-eph signaling during axon guidance. *J. Cell Biol.*, 214(1):35–44, 2016. ISSN 0021-9525. doi: 10.1083/jcb.201601085.

[71] Elena B. Pasquale. Eph receptor signalling casts a wide net on cell behaviour. *Nat. Rev. Mol. Cell Biol.*, 6(6):462–475, 2005. doi: 10.1038/nrm1662.

[72] Eduard Batlle, Jeffrey T. Henderson, Harry Beghtel, Maaike M.W. van den Born, Elena Sancho, Gerwin Huls, Jan Meeldijk, Jennifer Robertson, Marc van de Wetering, Tony Pawson, and Hans Clevers. β-catenin and tcf mediate cell positioning in the intestinal epithelium by controlling the expression of ephb/ephrinb. *Cell*, 111(2):251–263, 2002. ISSN 0092-8674. doi: 10.1016/S0092-8674(02)01015-2.

[73] David Sadava, David M. Hillis, H. Craig Heller, Sally D. Hacker, and Jürgen Markl. *Purves Biologie*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2019. ISBN 978-3-662-58171-1. doi: 10.1007/978-3-662-58172-8.

[74] Uwe Irion, Hans Georg Frohnhöfer, Jana Krauss, Tuba Çolak Champollion, Hans-Martin Maischein, Silke Geiger-Rudolph, Christian Weiler, and Christiane Nüsslein-Volhard. Gap junctions composed of connexins 41.8 and 39.4 are essential for colour pattern formation in zebrafish. *eLife*, 3:e05125, 2014. doi: 10.7554/eLife.05125.

[75] Santosh Manicka and Michael Levin. Modeling somatic computation with non-neural bioelectric networks. *Sci. Rep.*, 9(1):18612, 2019. ISSN 2045-2322. doi: 10.1038/s41598-019-54859-8.

[76] Bastiaan C. Buddingh' and Jan C. M. van Hest. Artificial cells: Synthetic compartments with life-like functionality and adaptivity. *Acc. Chem. Res.*, 50(4):769–777, 2017. doi: 10.1021/acs.accounts.6b00512.

[77] Chikako Kurokawa, Kei Fujiwara, Masamune Morita, Ibuki Kawamata, Yui Kawagishi, Atsushi Sakai, Yoshihiro Murayama, Shin-Ichiro M. Nomura, Satoshi Murata, Masahiro Takinoue, and Miho Yanagisawa. Dna cytoskeleton for stabilizing artificial cells. *PNAS*, 114(28):7228–7233, 2017. doi: 10.1073/pnas.1702208114.

[78] Marian Weiss, Johannes Patrick Frohnmayer, Lucia Theresa Benk, Barbara Haller, Jan-Willi Janiesch, Thomas Heitkamp, Michael Börsch, Rafael B. Lira, Rumiana Dimova, Reinhard Lipowsky, Eberhard Bodenschatz, Jean-Christophe Baret, Tanja Vidakovic-Koch, Kai Sundmacher, Ilia Platzman, and Joachim P. Spatz. Sequential bottom-up assembly of mechanically stabilized synthetic cells by microfluidics. *Nat. Mater.*, 17(1):89–96, 2018. doi: 10.1038/NMAT5005.

[79] Kilian Vogele, Thomas Frank, Lukas Gasser, Marisa A. Goetzfried, Mathias W. Hackl, Stephan A. Sieber, Friedrich C. Simmel, and Tobias Pirzer. Towards synthetic cells using peptide-based reaction compartments. *Nat. Commun.*, 9(1):3862, 2018. doi: 10.1038/s41467-018-06379-8.

[80] Xin Huang, Mei Li, David C. Green, David S. Williams, Avinash J. Patil, and Stephen Mann. Interfacial assembly of protein-polymer nano-conjugates into stimulus-responsive biomimetic protocells. *Nat. Commun.*, 4:2239, 2013. doi: 10.1038/ncomms3239.

[81] Dennis E. Discher and Fariyal Ahmed. Polymersomes. *Annu. Rev. Biomed. Eng.*, 8:323–341, 2006. ISSN 1523-9829. doi: 10.1146/annurev.bioeng.8.061505.095838.

[82] Alexandra M. Tayar, Eyal Karzbrun, Vincent Noireaux, and Roy H. Bar-Ziv. Propagating gene expression fronts in a one-dimensional coupled system of artificial cells. *Nat. Phys*, 11(12):1037–1041, 2015. ISSN 1745-2473. doi: 10.1038/nphys3469.

[83] Edward M. de Robertis. Spemann's organizer and self-regulation in amphibian embryos. *Nat. Rev. Mol. Cell Biol.*, 7(4):296–302, 2006. ISSN 1471-0072. doi: 10.1038/nrm1855.

[84] Erika Hoyos, Kerry Kim, Josselin Milloz, Michalis Barkoulas, Jean-Baptiste Penigault, Edwin Munro, and Marie-Anne Felix. Quantitative variation in autocrine signaling and pathway crosstalk in the caenorhabditis vulval network. *Curr. Biol.*, 21(7):527–538, 2011. doi: 10.1016/j.cub.2011.02.040.

[85] Thomas Gregor, David W. Tank, Eric F. Wieschaus, and William Bialek. Probing the limits to positional information. *Cell*, 130(1):153–164, 2007. ISSN 0092-8674. doi: 10.1016/j.cell.2007.05.025.

[86] Myriam Zecca and Gary Struhl. A feed-forward circuit linking wingless, fat-dachsous signaling, and the warts-hippo pathway to drosophila wing growth. *PLoS Biol.*, 8(6):e1000386, 2010. doi: 10.1371/journal.pbio.1000386.

[87] Hyung Chul Lee, Cato Hastings, Nidia M. M. Oliveira, Rubén Pérez-Carrasco, Karen M. Page, Lewis Wolpert, and Claudio D. Stern. 'neighbourhood watch' model: embryonic epiblast cells assess positional information in relation to their neighbours. *Development*, 149(10), 2022. doi: 10.1242/dev.200295.

[88] Nicolas E. Buchler, Ulrich Gerland, and Terence Hwa. On schemes of combinatorial transcription logic. *PNAS*, 100(9):5136–5141, 2003. ISSN 0027-8424. doi: 10.1073/pnas.0930314100.

[89] Uri Alon. *An introduction to systems biology: Design principles of biological circuits*. Mathematical and computational biology. Chapman & Hall/CRC, Boca Raton Fla. and London, 2nd ed. edition, 2019. ISBN 1-439-83718-X.

[90] Shiraz Kalir and Uri Alon. Using a quantitative blueprint to reprogram the dynamics of the flagella gene network. *Cell*, 117(6):713–720, 2004. ISSN 0092-8674. doi: 10.1016/j.cell.2004.05.010.

[91] Koichiro Uriu, Bo-Kai Liao, Andrew C. Oates, and Luis G. Morelli. From local resynchronization to global pattern recovery in the zebrafish segmentation clock. *eLife*, 10, 2021. doi: 10.7554/eLife.61358.

[92] Laurent Potvin-Trottier, Nathan D. Lord, Glenn Vinnicombe, and Johan Paulsson. Synchronous long-term oscillations in a synthetic gene circuit. *Nature*, 538(7626):514–517, 2016. ISSN 0028-0836. doi: 10.1038/nature19841.

[93] Daniel B. Forger. *Biological clocks, rhythms, and oscillations: The theory of biological timekeeping*. The MIT Press, Cambridge, Massachusetts, 2017. ISBN 9780262036771. URL https://www.ncbi.nlm.nih.gov/books/NBK544607/.

[94] Maximilian Weitz, Jongmin Kim, Korbinian Kapsner, Erik Winfree, Elisa Franco, and Friedrich C. Simmel. Diversity in the dynamical behaviour of a compartmentalized programmable biochemical oscillator. *Nat. Chem.*, 6(4):295–302, 2014. ISSN 1755-4349. doi: 10.1038/nchem.1869.

[95] Alexandra M. Tayar, Eyal Karzbrun, Vincent Noireaux, and Roy H. Bar-Ziv. Synchrony and pattern formation of coupled genetic oscillators on a chip of artificial cells. *PNAS*, 114(44):11609–11614, 2017. doi: 10.1073/pnas.1710620114.

[96] W. James Nelson and Roel Nusse. Convergence of wnt, beta-catenin, and cadherin pathways. *Science*, 303(5663):1483–1487, 2004. ISSN 0036-8075. doi: 10.1126/science.1094291.

[97] Barry M. Gumbiner. Regulation of cadherin-mediated adhesion in morphogenesis. *Nat. Rev. Mol. Cell Biol.*, 6(8):622–634, 2005. doi: 10.1038/nrm1699.

[98] Anna Stopka, Marcelo Boareto, and Dagmar Iber. Cell-based simulations of notch-dependent cell differentiation on growing domains. 2019. doi: 10.1101/859363v1.

[99] Charleston Noble, Adam N. Nilsson, Camilla Freitag, Jason P. Beech, Jonas O. Tegenfeldt, and Tobias Ambjörnsson. A fast and scalable kymograph alignment algorithm for nanochannel-based optical dna mappings. *PloS one*, 10(4):e0121905, 2015. ISSN 1932-6203. doi: 10.1371/journal.pone.0121905.

[100] Stephen Wolfram. Tables of cellular automaton properties, 1986.

[101] Wolfram|Alpha. rule 165, 2021. URL https://www.wolframalpha.com/input/?i=rule+165.

[102] Susan Wopat, Jennifer Bagwell, Kaelyn D. Sumigray, Amy L. Dickson, Leonie F. A. Huitema, Kenneth D. Poss, Stefan Schulte-Merker, and Michel Bagnat. Spine patterning is guided by segmentation of the notochord sheath. *Cell Rep.*, 22(8):2026–2038, 2018. ISSN 2211-1247. doi: 10.1016/j.celrep.2018.01.084.

[103] Milan Vispoel, Aisling J. Daly, and Jan M. Baetens. Progress, gaps and obstacles in the classification of cellular automata. *Phys. D: Nonlinear Phenom.*, T9(3):133074, 2021. ISSN 01672789. doi: 10.1016/j.physd.2021.133074.

[104] Cyrille de Joussineau, Jonathan Soulé, Marianne Martin, Christelle Anguille, Philippe Montcourrier, and Daniel Alexandre. Delta-promoted filopodia mediate long-range lateral inhibition in drosophila. *Nature*, 426(6966):555–559, 2003. ISSN 0028-0836. doi: 10.1038/nature02157.

[105] Stephen Wolfram. *A new kind of science*. Wolfram Media, Champaign Ill., 1. ed., 4. print edition, 2002. ISBN 978-1-57955-008-0.

[106] Karel Culik and Sheng Yu. Undecidability of ca classification schemes. *Complex Syst.*, 2(2), 1988. doi: 10.5555/45269.45271.

[107] Chris G. Langton. Computation at the edge of chaos: Phase transitions and emergent computation. *Phys. D: Nonlinear Phenom.*, 42(1-3):12–37, 1990. ISSN 01672789. doi: 10.1016/0167-2789(90)90064-V.

[108] Navot Israeli and Nigel Goldenfeld. Coarse-graining of cellular automata, emergence, and the predictability of complex systems. *Phys. Rev. E*, 73(2 Pt 2):026203, 2006. ISSN 1539-3755. doi: 10.1103/PhysRevE.73.026203.

[109] Peter D. Grunwald and Paul M. B. Vitanyi. Algorithmic information theory, 2008. URL http://arxiv.org/pdf/0809.2754v2.

[110] Kamaludin Dingle, Chico Q. Camargo, and Ard A. Louis. Input-output maps are strongly biased towards simple outputs. *Nat. Commun.*, 9(1):761, 2018. doi: 10.1038/s41467-018-03101-6.

[111] Gina Maira Barbosa de Oliveira, Pedro P.B. de Oliveira, and Nizam Omar. Guidelines for dynamics-based parameterization of one-dimensional cellular automata rule spaces. *Complexity*, 6(2):63–71, 2000. ISSN 1076-2787. doi: 10.1002/cplx.1021.

[112] Gérard Y. Vichniac. Boolean derivatives on cellular automata. *Phys. D: Nonlinear Phenom.*, 45(1-3): 63–74, 1990. ISSN 0167-2789. doi: 10.1016/0167-2789(90)90174-N.

[113] C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3):379–423, 1948. ISSN 00058580. doi: 10.1002/j.1538-7305.1948.tb01338.x.

[114] T. M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience, Hoboken, N.J, 2nd ed. edition, 2006. ISBN 978-0-471-24195-9. URL http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10305137.

[115] Bernhard Altaner. Nonequilibrium thermodynamics and information theory: basic concepts and relaxing dynamics. *J. Phys. A: Math. Theor.*, 50(45):454001, 2017. ISSN 1751-8113. doi: 10.1088/1751-8121/aa841d.

[116] Mareike Bojer. *Cellular Automata: From Theory to Applications in Biological Pattern Formation*. Phd, Technical University of Munich, München, 2022.

[117] Rick Quax, Gregor Chliamovitch, Alexandre Dupuis, Jean-Luc Falcone, Bastien Chopard, Alfons G. Hoekstra, and Peter M. A. Sloot. Information processing features can detect behavioral regimes of dynamical systems. *Complexity*, 2018(37):1–16, 2018. ISSN 1076-2787. doi: 10.1155/2018/6047846. URL http://arxiv.org/pdf/1709.09447v1.

[118] Bernd Aulbach. *Gewöhnliche Differenzialgleichungen*. Spektrum Akad. Verl., München, 2. aufl., unveränd. nachdr edition, 2010. ISBN 978-3-8274-1492-2.

[119] Steven Strogatz. *Nonlinear dynamics and chaos: With applications to physics, biology, chemistry, and engineering*. CRC Press Taylor & Francis Group, Boca Raton, second edition edition, 2014. ISBN 978-0813349107. URL https://www.taylorfrancis.com/books/9780429680168.

[120] Andrew Wuensche and Mike Lesser. *The global dynamics of cellular automata: An atlas of basin of attraction fields of one-dimensional cellular automata*, volume 1 of *Santa Fe Institute studies in the sciences of complexity Reference volumes*. Addison-Wesley, Reading, Mass and Wokingham, 1992. ISBN 0-201-55740-1.

[121] Ilya R. Akberding, Evgeniy A. Ozonov, Victoria V. Mironova, Nadezda A. Omelyanchuk, Vitaly A. Likhoshvai, Dmytry N. Gorpinchenko, and Nikolai A. Kolchanov. A cellular automaton to model the development of primary shoot meristems of arabidopsis thaliana. *J Bioinform Comput Biol*, 05(02b): 641–650, 2007. ISSN 0219-7200. doi: 10.1142/S0219720007002862.

[122] Liana Manukyan, Sophie A. Montandon, Anamarija Fofonjka, Stanislav Smirnov, and Michel C. Milinkovitch. A living mesoscopic cellular automaton made of skin scales. *Nature*, 544(7649):173–179, 2017. ISSN 0028-0836. doi: 10.1038/nature22031.

[123] Takanori Kihara, Kosuke Kashitani, and Jun Miyake. In silico characterization of cell-cell interactions using a cellular automata model of cell culture. *BMC Res. Notes*, 10(1):283, 2017. ISSN 1756-0500. doi: 10.1186/s13104-017-2613-x.

[124] Andreas Deutsch and Sabine Dormann. *Cellular Automaton Modeling of Biological Pattern Formation: Characterization, Applications, and Analysis*. Modeling and Simulation in Science, Engineering and Technology. Birkhäuser Boston, Boston, MA, 2005. ISBN 9780817642815. doi: 10.1007/b138451. URL http://gbv.eblib.com/patron/FullRecord.aspx?p=337234.

[125] G. B. Ermentrout and L. Edelstein-Keshet. Cellular automata approaches to biological modeling. *J. Theor. Biol.*, 160(1):97–133, 1993. ISSN 00225193. doi: 10.1006/jtbi.1993.1007.

[126] H.S Silva and M.L Martins. A cellular automata model for cell differentiation. *Phys. A: Stat. Mech. Appl.*, 322:555–566, 2003. ISSN 03784371. doi: 10.1016/S0378-4371(02)01807-1.

[127] Eduardo P. Olimpio, Yiteng Dang, and Hyun Youk. Statistical dynamics of spatial-order formation by communicating cells. *iScience*, 2:27–40, 2018. ISSN 25890042. doi: 10.1016/j.isci.2018.03.013.

[128] Alyssa Adams, Hector Zenil, Paul C. W. Davies, and Sara Imari Walker. Formal definitions of unbounded evolution and innovation reveal universal mechanisms for open-ended evolution in dynamical systems. *Sci. Rep.*, 7(1):997, 2017. doi: 10.1038/s41598-017-00810-8.

[129] Alyssa Adams, Angelica Berner, Paul Davies, and Sara Walker. Physical universality, state-dependent dynamical laws and open-ended novelty. *Entropy*, 19(9):461, 2017. doi: 10.3390/e19090461.

[130] Carla Mulas, Agathe Chaigne, Austin Smith, and Kevin J. Chalut. Cell state transitions: definitions and challenges. *Development*, 148(20), 2021. doi: 10.1242/dev.199950.

[131] Jan Baedke and Tobias Schöttler. Visual metaphors in the sciences: The case of epigenetic landscape images. *J. Gen. Philos. Sci.*, 48(2):173–194, 2017. ISSN 0925-4560. doi: 10.1007/s10838-016-9353-9.

[132] Conrad Hal Waddington. *The strategy of the genes: A discussion of some aspects of theoretical biology*. Allen & Unwin, London, 1957.

[133] Cole Trapnell. Defining cell types and states with single-cell genomics. *Genome Res.*, 25(10):1491–1498, 2015. ISSN 1549-5469. doi: 10.1101/gr.190595.115.

[134] Meritxell Sáez, Robert Blassberg, Elena Camacho-Aguilar, Eric D. Siggia, David A. Rand, and James Briscoe. Statistically derived geometrical landscapes capture principles of decision-making dynamics during cell fate transitions. *Cell Syst.*, 2021. ISSN 2405-4712. doi: 10.1016/j.cels.2021.08.013.

[135] Melody K. Morris, Julio Saez-Rodriguez, Peter K. Sorger, and Douglas A. Lauffenburger. Logic-based models for the analysis of cell signaling networks. *Biochemistry*, 49(15):3216–3224, 2010. ISSN 1520-4995. doi: 10.1021/bi902202q.

[136] Roeland M.H. Merks and James A. Glazier. A cell-centered approach to developmental biology. *Phys. A: Stat. Mech. Appl.*, 352(1):113–130, 2005. ISSN 03784371. doi: 10.1016/j.physa.2004.12.028.

[137] Sagar and Dominic Grün. Deciphering cell fate decision by integrated single-cell sequencing analysis. *Annu. Rev. Biomed. Data Sci.*, 3:1–22, 2020. doi: 10.1146/annurev-biodatasci-111419-091750.

[138] Filippo Castiglione. Agent based modeling and simulation, introduction to. In Robert A. Meyers, editor, *Encyclopedia of complexity and systems science*. Springer, New York, NY, 2009. ISBN 9780387758886.

[139] Jonathan M. Raser and Erin K. O'Shea. Noise in gene expression: origins, consequences, and control. *Science*, 309(5743):2010–2013, 2005. doi: 10.1126/science.1105891.

[140] Lev S. Tsimring. Noise in biology. *Rep. Prog. Phys.*, 77(2):026601, 2014. ISSN 0034-4885. doi: 10.1088/0034-4885/77/2/026601.

[141] Christoph Zechner, Elisa Nerli, and Caren Norden. Stochasticity and determinism in cell fate decisions. *Development*, 147(14), 2020. doi: 10.1242/dev.181495.

[142] Joe Padgett and Silvia D. M. Santos. From clocks to dominoes: lessons on cell cycle remodelling from embryonic stem cells. *FEBS letters*, 2020. doi: 10.1002/1873-3468.13862.

[143] Nicolaas Godfried van Kampen. *Stochastic Processes in Physics and Chemistry*. Elsevier, 2007. ISBN 9780444529657. doi: 10.1016/B978-0-444-52965-7.X5000-4.

[144] Ramón Alonso-Sanz. Cellular automata with memory. In Robert A. Meyers, editor, *Encyclopedia of complexity and systems science*. Springer, New York, NY, 2009. ISBN 9780387758886.

[145] Bastien Chopard. Cellular automata modeling of physical systems. In Robert A. Meyers, editor, *Encyclopedia of complexity and systems science*. Springer, New York, NY, 2009. ISBN 9780387758886.

[146] Stephan Kremser, Gabriel Vercelli, and Ulrich Gerland. The french flag problem revisited: Creating robust and tunable axial patterns without global signaling. In Preparation, 2023.

[147] María Molina Antón. Report on project 'pattern formation via local cell-cell signalling with variable interaction ranges'. personal communication, 19.09.2022.

[148] Lukas Zett. *Robust, Precise, and Modular Solutions to the French Flag Problem in Two Spatial Dimensions without Global Signaling*. Master thesis, Ludwig-Maximilian-Universität, München, 06.04.2023.

[149] Rinat Arbel-Goren, Valentina Buonfiglio, Francesca Di Patti, Sergio Camargo, Anna Zhitnitsky, Ana Valladares, Enrique Flores, Antonia Herrero, Duccio Fanelli, and Joel Stavans. Robust, coherent, and synchronized circadian clock-controlled oscillations along anabaena filaments. *eLife*, 10, 2021. doi: 10.7554/eLife.64348.

[150] Larissa B. Patterson and David M. Parichy. Zebrafish pigment pattern formation: Insights into the development and evolution of adult form. *Annu. Rev. Genet.*, 53:505–530, 2019. doi: 10.1146/annurev-genet-112618-043741.

[151] Christian Dahmann, Andrew C. Oates, and Michael Brand. Boundary formation and maintenance in tissue development. *Nat. Rev. Genet.*, 12(1):43–55, 2011. ISSN 1471-0056. doi: 10.1038/nrg2902.

[152] Thimo Rohlf and Stefan Bornholdt. Self-organized pattern formation and noise-induced control based on particle computations. *J. Stat. Mech. Theory Exp.*, 2005(12):L12001–L12001, 2005. ISSN 1742-5468. doi: 10.1088/1742-5468/2005/12/L12001.

[153] Agoston E. Eiben and Jim Smith. From evolutionary computation to the evolution of things. *Nature*, 521(7553):476–482, 2015. ISSN 0028-0836. doi: 10.1038/nature14544.

[154] Timothy L. Bailey. Discovering sequence motifs. In Jonathan M. Keith, editor, *Bioinformatics Volume I*, pages 231–251. Humana Press, 2008. ISBN 978-1-60327-159-2.

[155] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *J Mach Learn Res*, 9: 2579–2605, 2008.

[156] Laurens van der Maaten. Learning a parametric embedding by preserving local structure. *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.

[157] Laurens van der Maaten. Accelerating t-sne using tree-based algorithms. *J Mach Learn Res*, 15: 1–21, 2014.

[158] Donald Ervin Knuth. *Sorting and searching*, volume volume 3 of *The art of computer programming*. Addison-Wesley, Boston, second edition, forty-second printing edition, 1998. ISBN 978-0-201-89685-5.

[159] Wikipedia contributors. French ensigns, 2022. URL https://en.wikipedia.org/w/index.php?title=French_ensigns&oldid=1076569056.

[160] Thimo Rohlf and Stefan Bornholdt. Morphogenesis by coupled regulatory networks: Reliable control of positional information and proportion regulation. *J. Theor. Biol.*, 261(2):176–193, 2009. ISSN 00225193. doi: 10.1016/j.jtbi.2009.07.023.

[161] Bertie Ancona, Ayesha Bajwa, Nancy Lynch, and Mallmann-Trenn Frederik. How to color a french flag: Biologically inspired algorithms for scale-invariant patterning. In Keren Censor-Hillel and Michele Flammini, editors, *Structural Information and Communication Complexity*, pages 327–331. Springer International Publishing, Cham, 2019. ISBN 978-3-030-24921-2. doi: 10.1007/978-3-030-24922-9{\textunderscore}22.

[162] Tiago Ramalho. *Information processing in biology: A study on signaling and emergent computation*. Dissertation, Ludwig-Maximilian-Universität, München, 2015. URL http://d-nb.info/107885209X/34.

[163] Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites, 2015. URL http://arxiv.org/pdf/1504.04909v1.

[164] Justin K. Pugh, Lisa B. Soros, and Kenneth O. Stanley. Quality diversity: A new frontier for evolutionary computation. *Front. Robot. AI*, 3:3, 2016. doi: 10.3389/frobt.2016.00040.

[165] Antoine Cully and Yiannis Demiris. Quality and diversity optimization: A unifying modular framework. *IEEE Trans. Evol. Comput.*, 22(2):245–259, 2018. ISSN 1089-778X. doi: 10.1109/TEVC.2017.2704781.

[166] Hao Wu. *Control of One Dimensional Cellular Automata*. Master thesis, Ludwig-Maximilian-Universität, München, 2013.

[167] Stephan Kremser. *Controllability of Pattern Formation in Discrete Systems*. Master thesis, Technical University of Munich, München, 2017.

[168] Tiago Ramalho, Stephan Kremser, Hao Wu, and Ulrich Gerland. Programmable pattern formation in cellular systems with local signaling. *Commun. Phys.*, 4(1):8605, 2021. doi: 10.1038/s42005-021-00639-8.

[169] Benjamin Prud'homme, Nicolas Gompel, and Sean B. Carroll. Emerging principles of regulatory evolution. *PNAS*, 104 Suppl 1:8605–8612, 2007. ISSN 0027-8424. doi: 10.1073/pnas.0700488104.

[170] Creative Commons. Creative commons attribution 4.0 international (cc by 4.0), 2021. URL http://creativecommons.org/licenses/by/4.0/.

[171] Mark E. J. Newman. *Networks: An introduction*. Oxford Univ. Press, Oxford, 2010. ISBN 9780199206650.

[172] Gábor Csárdi and Tamás Nepusz. The igraph software package for complex network research. *Int. j. complex syst.*, 1695(5):1–9, 2006.

[173] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gäel Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15. Passadena, 2008.

[174] Tilo Arens, Frank Hettlich, Christian Karpfinger, Ulrich Kockelkorn, Klaus Lichtenegger, and Hellmuth Stachel. *Mathematik*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015. ISBN 978-3-642-44918-5. doi: 10.1007/978-3-642-44919-2.

[175] Lukas Zett. Patterning the french flag with 2d ca: verbal, 05.10.2022.

[176] Jose Negrete and Andrew C. Oates. Towards a physical understanding of developmental patterning. *Nat. Rev. Genet.*, 22(8):518–531, 2021. ISSN 1471-0056. doi: 10.1038/s41576-021-00355-7.

[177] C. J. Twining and P.-M. Binder. Enumeration of limit cycles in noncylindrical cellular automata. *J. Stat. Phys.*, 66(1-2):385–401, 1992. ISSN 0022-4715. doi: 10.1007/BF01060073.

[178] I. Farkas, D. Helbing, and T. Vicsek. Mexican waves in an excitable medium. *Nature*, 419(6903): 131–132, 2002. ISSN 0028-0836. doi: 10.1038/419131a.

[179] Michael Schulz, editor. *Control Theory in Physics and other Fields of Science: Concepts, Tools, and Applications*, volume 215 of *Springer Tracts in Modern Physics*. Springer-Verlag, Berlin Heidelberg, 2006. ISBN 9783540295143. doi: 10.1007/11374343. URL `http://dx.doi.org/10.1007/11374343`.

[180] Yang-Yu Liu, Jean-Jacques Slotine, and Albert-Laszlo Barabasi. Controllability of complex networks. *Nature*, 473(7346):167–173, 2011. ISSN 0028-0836. doi: 10.1038/nature10011.

[181] Yang-Yu Liu and Albert-László Barabási. Control principles of complex systems. *Rev. Mod. Phys.*, 88(3), 2016. ISSN 0034-6861. doi: 10.1103/RevModPhys.88.035006.

[182] Franco Bagnoli, Samira El Yacoubi, and Raul Rechtman. Regional control of boolean cellular automata. In Samira El Yacoubi, Jaroslaw Wąs, and Stefania Bandini, editors, *Cellular Automata*, volume 9863, pages 101–112. Springer International Publishing, Cham, 2016. ISBN 978-3-319-44364-5. doi: 10.1007/978-3-319-44365-2{\textunderscore}10.

[183] Franco Bagnoli, Samira El Yacoubi, and Raúl Rechtman. Toward a boundary regional control problem for boolean cellular automata. *Nat. Comput.*, 17(3):479–486, 2018. ISSN 1567-7818. doi: 10.1007/s11047-017-9626-1.

[184] S. Dridi, S. El Yacoubi, F. Bagnoli, and A. Fontaine. A graph theory approach for regional controllability of boolean cellular automata. *Int J Parallel Emergent Distrib Syst*, 35(5):499–513, 2020. doi: 10.1080/17445760.2019.1608442.

[185] Franco Bagnoli, Sara Dridi, Samira El Yacoubi, and Raúl Rechtman. Regional control of probabilistic cellular automata. In Giancarlo Mauri, Samira El Yacoubi, Alberto Dennunzio, Katsuhiro Nishinari, and Luca Manzoni, editors, *Cellular Automata*, volume 11088, pages 243–254. Springer International Publishing, Cham, 2018. ISBN 978-3-319-99812-1. doi: 10.1007/978-3-319-99813-8_22.

[186] Franco Bagnoli, Sara Dridi, Samira El Yacoubi, and Raúl Rechtman. Optimal and suboptimal regional control of probabilistic cellular automata. *Nat. Comput.*, 3(3):307, 2019. ISSN 1567-7818. doi: 10.1007/s11047-019-09763-5.

[187] Susanna E. Brantley and Stefano Di Talia. Cell cycle control during early embryogenesis. *Development*, 148(13), 2021. doi: 10.1242/dev.193128.

[188] Michael Rubenstein, Alejandro Cornejo, and Radhika Nagpal. Robotics. programmable self-assembly in a thousand-robot swarm. *Science*, 345(6198):795–799, 2014. doi: 10.1126/science.1254295.

[189] I. Slavkov, D. Carrillo-Zapata, N. Carranza, X. Diego, F. Jansson, J. Kaandorp, S. Hauert, and J. Sharpe. Morphogenesis in robot swarms. *Sci. Robot.*, 3(25), 2018. doi: 10.1126/scirobotics.aau9178.

[190] Peng Yin, Sudheer Sahu, Andrew J. Turberfield, and John H. Reif. Design of autonomous dna cellular automata. In Alessandra Carbone and Niles A. Pierce, editors, *DNA computing: DNA11, London, ON, Canada, June 6 - 9, 2005 ; revised selected papers*, volume 3892 of *Lecture notes in computer science*, pages 399–416. Springer, Berlin, 2006. ISBN 978-3-540-34161-1.

[191] Stephen Wolfram. Cellular automata as models of complexity. *Nature*, 311(5985):419–424, 1984. ISSN 0028-0836. doi: 10.1038/311419a0.

[192] Nazim Fatès. A guided tour of asynchronous cellular. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Jarkko Kari, Martin Kutrib, and Andreas Malcher, editors, *Cellular Automata and Discrete Complex Systems*, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-40866-3.

[193] Stefania Bandini, Andrea Bonomi, and Giuseppe Vizzari. An analysis of different types and effects of asynchronicity in cellular automata update schemes. *Nat. Comput.*, 11(2):277–287, 2012. ISSN 1567-7818. doi: 10.1007/s11047-012-9310-4.

[194] Olivier Bouré, Nazim Fatès, and Vincent Chevrier. Probing robustness of cellular automata through variations of asynchronous updating. *Nat. Comput.*, 11(4):553–564, 2012. ISSN 1567-7818. doi: 10.1007/s11047-012-9340-y.

[195] Renato E. Mirollo and Steven H. Strogatz. Synchronization of pulse-coupled biological oscillators. *SIAM J Appl Math*, 50(6):1645–1662, 1990. ISSN 0036-1399. doi: 10.1137/0150098.

[196] Juan A. Acebrón, L. L. Bonilla, Conrad J. Pérez Vicente, Félix Ritort, and Renato Spigler. The kuramoto model: A simple paradigm for synchronization phenomena. *Phys. Rev. E*, 77(1):137–185, 2005. ISSN 1063-651X. doi: 10.1103/RevModPhys.77.137.

[197] G. Korniss, M. A. Novotny, H. Guclu, Z. Toroczkai, and P. A. Rikvold. Suppressing roughness of virtual times in parallel discrete-event simulations. *Science*, 299(5607):677–679, 2003. doi: 10.1126/science.1079382.

[198] Kazumasa A. Takeuchi. An appetizer to modern developments on the kardar–parisi–zhang universality class. *Phys. A: Stat. Mech. Appl.*, 504:77–105, 2018. ISSN 03784371. doi: 10.1016/j.physa.2018.03.009.

[199] M. Kardar, G. Parisi, and Y. C. Zhang. Dynamic scaling of growing interfaces. *Phys. Rev. Lett.*, 56 (9):889–892, 1986. ISSN 0031-9007. doi: 10.1103/PhysRevLett.56.889.

[200] Konstantin Matetski, Jeremy Quastel, and Daniel Remenik. The kpz fixed point, 2017. URL `https://arxiv.org/abs/1701.00018`.

[201] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nat. Methods*, 17(3):261–272, 2020. doi: 10.1038/s41592-019-0686-2.

[202] Korniss, Toroczkai, Novotny, and Rikvold. From massively parallel algorithms and fluctuating time horizons to nonequilibrium surface growth. *Phys. Rev. Lett.*, 84(6):1351–1354, 2000. ISSN 0031-9007. doi: 10.1103/PhysRevLett.84.1351.

[203] Bruce Alberts, Alexander Johnson, Julian Lewis, David Morgan, Martin Raff, Keith Roberts, Peter Walter, John Wilson, and Tim Hunt. *Molecular biology of the cell*. Garland Science Taylor and Francis Group, New York, NY, sixth edition edition, 2015. ISBN 978-0-8153-4432-2.

[204] Ludwig Burger. *An Ising-type model for pattern formation in compartmentalized systems*. Bachelor's thesis, Technical University of Munich, München, 23.09.2019.

[205] Mareike Bojer, Stephan Kremser, and Ulrich Gerland. Robust boundary formation in a morphogen gradient via cell-cell signaling. *Phys. Rev. E*, 105(6), 2022. ISSN 1063-651X. doi: 10.1103/PhysRevE.105.064405.

[206] Tobias Bollenbach, Periklis Pantazis, Anna Kicheva, Christian Bökel, Marcos González-Gaitán, and Frank Jülicher. Precision of the dpp gradient. *Development*, 135(6):1137–1146, 2008. doi: 10.1242/dev.012062.

[207] Wolfgang Driever and Christiane Nüsslein-Volhard. The bicoid protein determines position in the drosophila embryo in a concentration-dependent manner. *Cell*, 54(1):95–104, 1988. ISSN 0092-8674. doi: 10.1016/0092-8674(88)90183-3.

[208] Patrick Hillenbrand, Ulrich Gerland, and Gasper Tkacik. Beyond the french flag model: Exploiting spatial and gene regulatory interactions for positional information. *PloS one*, 11(9):e0163628, 2016. ISSN 1932-6203. doi: 10.1371/journal.pone.0163628.

[209] Tobias Bollenbach. *Formation of morphogen gradients*. Dissertation, Technischen Universitĺat Dresden, Dresden, 2005.

[210] Szabolcs Zakany, Stanislav Smirnov, and Michel C. Milinkovitch. Lizard skin patterns and the ising model. *Phys. Rev. Lett.*, 128(4):613, 2022. ISSN 0031-9007. doi: 10.1103/PhysRevLett.128.048102.

[211] Kurt Binder and Dieter W. Heermann. *Monte Carlo Simulation in Statistical Physics*. Springer International Publishing, Cham, 2019. ISBN 978-3-030-10757-4. doi: 10.1007/978-3-030-10758-1.

[212] Stephen Eltinge. Numerical ising model simulations on exactly solvable and randomized lattices, 2015.

[213] Avraham E. Mayo, Yaakov Setty, Seagull Shavit, Alon Zaslaver, and Uri Alon. Plasticity of the cis-regulatory input function of a gene. *PLoS Biol.*, 4(4):e45, 2006. doi: 10.1371/journal.pbio.0040045.

[214] Naama Barkai and Ben-Zion Shilo. Reconstituting tissue patterning. *Science*, 370(6514):292–293, 2020. doi: 10.1126/science.abe4217.

[215] Aurore Dupin, Lukas Aufinger, Igor Styazhkin, Florian Rothfischer, Benedikt Kaufmann, Sascha Schwarz, Nikolas Galensowske, Hauke Clausen-Schaumann, and Friedrich C. Simmel. Synthetic cell-based materials extract positional information from morphogen gradients, 2021.

[216] Içvara Barbier, Rubén Perez-Carrasco, and Yolanda Schaerli. Controlling spatiotemporal pattern formation in a concentration gradient with a synthetic toggle switch. *Mol. Syst. Biol.*, 16(6):e9361, 2020. doi: 10.15252/msb.20199361.

[217] Jordi van Gestel, Tasneem Bareia, Bar Tenennbaum, Alma Dal Co, Polina Guler, Nitzan Aframian, Shani Puyesky, Ilana Grinberg, Glen G. D'Souza, Zohar Erez, Martin Ackermann, and Avigdor Eldar. Short-range quorum sensing controls horizontal gene transfer at micron scale in bacterial communities. *Nat. Commun.*, 12(1):2324, 2021. doi: 10.1038/s41467-021-22649-4.

[218] Greg D. Field and Fred Rieke. Nonlinear signal transfer from mouse rods to bipolar cells and implications for visual sensitivity. *Neuron*, 34(5):773–785, 2002. doi: 10.1016/S0896-6273(02)00700-6.

[219] Sabina Orazov. *Interplay of Mechanics and Information Processing in Cell Tissues*. Master thesis, Ludwig-Maximilian-Universität, München, 2022.

[220] Reza Farhadifar, Jens-Christian Röper, Benoit Aigouy, Suzanne Eaton, and Frank Jülicher. The influence of cell mechanics, cell-cell interactions, and proliferation on epithelial packing. *Curr. Biol.*, 17(24):2095–2104, 2007. ISSN 09609822. doi: 10.1016/j.cub.2007.11.049.

[221] Kaoru Sugimura and Shuji Ishihara. The mechanical anisotropy in a tissue promotes ordering in hexagonal cell packing. *Development*, 140(19):4091–4101, 2013. ISSN 0950-1991. doi: 10.1242/dev.094060.

[222] Silvanus Alt, Poulami Ganguly, and Guillaume Salbreux. Vertex models: from cell mechanics to tissue morphogenesis. *Philos. Trans. R. Soc. Lond., B, Biol. Sci.*, 372(1720), 2017. doi: 10.1098/rstb.2015.0520.

[223] Zena Hadjivasiliou, Ginger L. Hunter, and Buzz Baum. A new mechanism for spatial pattern formation via lateral and protrusion-mediated lateral signalling. *J. R. Soc. Interface*, 13(124), 2016. ISSN 1742-5662. doi: 10.1098/rsif.2016.0484.

[224] Dino Di Carlo. A mechanical biomarker of cell state in medicine. *J. Lab. Autom.*, 17(1):32–42, 2012. doi: 10.1177/2211068211431630.

[225] Ginger L. Hunter, Zena Hadjivasiliou, Hope Bonin, Li He, Norbert Perrimon, Guillaume Charras, and Buzz Baum. Coordinated control of notch/delta signalling and cell cycle progression drives lateral inhibition-mediated tissue patterning. *Development*, 143(13):2305–2310, 2016. doi: 10.1242/dev.134213.

[226] Max Sina Knicker. *A Minimal Model for Tissue Homeostasis in the Small Intestine Based on Probabilistic Cellular Automata*. Master thesis, Technical University of Munich, München, 30.04.2021.

[227] Alana M. Chin, David R. Hill, Megan Aurora, and Jason R. Spence. Morphogenesis and maturation of the embryonic and postnatal intestine. *Semin. Cell Dev. Biol.*, 2017.

[228] Joana Costa and Arti Ahluwalia. Advances and current challenges in intestinal in vitro model engineering: A digest. *Front. Bioeng. Biotechnol.*, 7:144, 2019. ISSN 2296-4185. doi: 10.3389/fbioe.2019.00144.

[229] Elise S. Demitrack and Linda C. Samuelson. Notch regulation of gastrointestinal stem cells. *J. Physiol.*, 594(17):4791–4803, 2016. doi: 10.1113/JP271667.

[230] Rocio Sancho, Catherine A. Cremona, and Axel Behrens. Stem cell and progenitor fate in the mammalian intestine: Notch and lateral inhibition in homeostasis and disease. *EMBO Rep.*, 16(5):571–581, 2015. doi: 10.15252/embr.201540188.

[231] Toshio Takahashi and Akira Shiraishi. Stem cell signaling pathways in the small intestine. *Int. J. Mol. Sci.*, 21(6), 2020. doi: 10.3390/ijms21062032.

[232] Peter Buske, Jörg Galle, Nick Barker, Gabriela Aust, Hans Clevers, and Markus Loeffler. A comprehensive model of the spatio-temporal stem cell and tissue organisation in the intestinal crypt. *PLoS Comput. Biol.*, 7(1):e1001045, 2011. ISSN 1553-734X. doi: 10.1371/journal.pcbi.1001045.

[233] Carmen Pin, Alastair J. M. Watson, and Simon R. Carding. Modelling the spatio-temporal cell dynamics reveals novel insights on cell differentiation and proliferation in the small intestinal crypt. *PloS one*, 7(5):e37115, 2012. ISSN 1932-6203. doi: 10.1371/journal.pone.0037115.

[234] Qihang Hou, Lulu Ye, Lulu Huang, and Qinghua Yu. The research progress on intestinal stem cells and its relationship with intestinal microbiota. *Front. Immunol.*, 8:599, 2017. ISSN 1664-3224. doi: 10.3389/fimmu.2017.00599.

[235] Amanda T. Mah, Kelley S. Yan, and Calvin J. Kuo. Wnt pathway regulation of intestinal stem cells. *J. Physiol.*, 594(17):4837–4847, 2016. doi: 10.1113/JP271754.

[236] S. Chwalinski and C. S. Potten. Crypt base columnar cells in ileum of bdf1 male mice–their numbers and some features of their proliferation. *Am. J. Anat.*, 186(4):397–406, 1989. ISSN 0002-9106. doi: 10.1002/aja.1001860409.

[237] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, pages 1942–1948. IEEE, 27 Nov.-1 Dec. 1995. ISBN 0-7803-2768-3. doi: 10.1109/ICNN.1995.488968.

[238] K. E. Parsopoulos and M. N. Vrahatis. Recent approaches to global optimization problems through particle swarm optimization. *Nat. Comput.*, 1(2/3):235–306, 2002. ISSN 1567-7818. doi: 10.1023/A:1016568309421.

[239] G. N. Lance and W. T. Williams. Computer programs for hierarchical polythetic classification ("similarity analyses"). *Comput. J*, 9(1):60–64, 1966. ISSN 0010-4620. doi: 10.1093/comjnl/9.1.60.

[240] J. M. Qiu, S. A. Roberts, and C. S. Potten. Cell migration in the small and large bowel shows a strong circadian rhythm. *Epithelial Cell Biol.*, 3(4):137–148, 1994. ISSN 0940-9912.

[241] U. Paulus, M. Loeffler, J. Zeidler, G. Owen, and C. S. Potten. The differentiation and lineage development of goblet cells in the murine small intestinal crypt: experimental and modelling studies. *J. Cell Sci.*, 106(2):473–483, 1993. ISSN 0021-9533. doi: 10.1242/jcs.106.2.473.

[242] David Basanta, Mark Miodownik, and Buzz Baum. The evolution of robust development and homeostasis in artificial organisms. *PLoS Comput. Biol.*, 4(3):e1000030, 2008. ISSN 1553-734X. doi: 10.1371/journal.pcbi.1000030.

[243] Marc Barthélemy. *Morphogenesis of spatial networks*. Lecture Notes in Morphogenesis. Springer, Cham, 2018. ISBN 9783319205649. doi: 10.1007/978-3-319-20565-6. URL `http://dx.doi.org/10.1007/978-3-319-20565-6`.

[244] Matthew D. B. Jackson, Salva Duran-Nebreda, and George W. Bassel. Network-based approaches to quantify multicellular development. *J. R. Soc. Interface*, 14(135), 2017. ISSN 1742-5662. doi: 10.1098/rsif.2017.0484.

[245] Manlio de Domenico, Albert Solé-Ribalta, Emanuele Cozzo, Mikko Kivelä, Yamir Moreno, Mason A. Porter, Sergio Gómez, and Alex Arenas. Mathematical formulation of multilayer networks. *Phys. Rev. X.*, 3(4):1082, 2013. ISSN 0002-9890. doi: 10.1103/PhysRevX.3.041022.

[246] Hyobin Kim and Hiroki Sayama. Robustness and evolvability of multilayer gene regulatory networks, 2018.

[247] Paul François, Vincent Hakim, and Eric D. Siggia. Deriving structure from evolution: Metazoan segmentation. *Mol. Syst. Biol.*, 3:154, 2007. doi: 10.1038/msb4100192.

[248] Chloé S. Baron and Alexander van Oudenaarden. Unravelling cellular relationships during development and regeneration using genetic lineage tracing. *Nat. Rev. Mol. Cell Biol.*, 20(12):753–765, 2019. doi: 10.1038/s41580-019-0186-3.

[249] Michaela Asp, Joseph Bergenstråhle, and Joakim Lundeberg. Spatially resolved transcriptomes-next generation tools for tissue exploration. *BioEssays*, 42(10):e1900221, 2020. doi: 10.1002/bies.201900221.

[250] Erick Armingol, Adam Officer, Olivier Harismendy, and Nathan E. Lewis. Deciphering cell-cell interactions and communication from gene expression. *Nat. Rev. Genet.*, 22(2):71–88, 2021. ISSN 1471-0056. doi: 10.1038/s41576-020-00292-x.

[251] Ravi U. Sheth and Harris H. Wang. Dna-based memory devices for recording cellular events. *Nat. Rev. Genet.*, 19(11):718–732, 2018. ISSN 1471-0056. doi: 10.1038/s41576-018-0052-8.

[252] Adam L. Haber, Moshe Biton, Noga Rogel, Rebecca H. Herbst, Karthik Shekhar, Christopher Smillie, Grace Burgin, Toni M. Delorey, Michael R. Howitt, Yarden Katz, Itay Tirosh, Semir Beyaz, Danielle Dionne, Mei Zhang, Raktima Raychowdhury, Wendy S. Garrett, Orit Rozenblatt-Rosen, Hai Ning Shi, Omer Yilmaz, Ramnik J. Xavier, and Aviv Regev. A single-cell survey of the small intestinal epithelium. *Nature*, 551(7680):333–339, 2017. ISSN 0028-0836. doi: 10.1038/nature24489.

[253] Nikos Karaiskos, Philipp Wahle, Jonathan Alles, Anastasiya Boltengagen, Salah Ayoub, Claudia Kipar, Christine Kocks, Nikolaus Rajewsky, and Robert P. Zinzen. The drosophila embryo at single-cell transcriptome resolution. *Science*, 358(6360):194–199, 2017. ISSN 0036-8075. doi: 10.1126/science.aan3235.

[254] Mireya Plass, Jordi Solana, F. Alexander Wolf, Salah Ayoub, Aristotelis Misios, Petar Glažar, Benedikt Obermayer, Fabian J. Theis, Christine Kocks, and Nikolaus Rajewsky. Cell type atlas and lineage tree of a whole complex animal by single-cell transcriptomics. *Science*, 360(6391), 2018. doi: 10.1126/science.aaq1723.

[255] Paul Villoutreix. What machine learning can do for developmental biology. *Development*, 148(1), 2021. doi: 10.1242/dev.188474.

[256] Ryan N. Gutenkunst, Joshua J. Waterfall, Fergal P. Casey, Kevin S. Brown, Christopher R. Myers, and James P. Sethna. Universally sloppy parameter sensitivities in systems biology models. *PLoS Comput. Biol.*, 3(10):1871–1878, 2007. ISSN 1553-734X. doi: 10.1371/journal.pcbi.0030189.

[257] Xiaojie Qiu, Yan Zhang, Jorge D. Martin-Rufino, Chen Weng, Shayan Hosseinzadeh, Dian Yang, Angela N. Pogson, Marco Y. Hein, Kyung Hoi Joseph Min, Li Wang, Emanuelle I. Grody, Matthew J. Shurtleff, Ruoshi Yuan, Song Xu, Yian Ma, Joseph M. Replogle, Eric S. Lander, Spyros Darmanis, Ivet Bahar, Vijay G. Sankaran, Jianhua Xing, and Jonathan S. Weissman. Mapping transcriptomic vector fields of single cells. *Cell*, 185(4):690–711.e45, 2022. ISSN 0092-8674. doi: 10.1016/j.cell.2021.12.045.

[258] Mayukh Deb, Ujjwal Singh, Mainak Deb, and Bradly Alicea. Devolearn: Machine learning models and education that enable computational developmental biology, 2022.

[259] Wolfram|Alpha. Wolfram|alpha, 2022. URL `https://www.wolframalpha.com/`.