NTNU – Trondheim
Norwegian University of
Science and Technology

A new Method of Output from Cellular Automata: The Togglecount Transform

Terje Schjelderup

Master of Science in Computer Science
Submission date: March 2015
Supervisor: Gunnar Tufte, IDI
Co-supervisor: Stefano Nichele, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Project description

Investigate how the emergent behaviour of a cellular automata like machine can be transformed to multiple variables.

A preliminary investigation of using local frequencies as a representation of output variables.

Supervisor: Gunnar Tufte, IDI
Co-supervisor: Stefano Nichele, IDI
Assignment given: 2014-10-15

Abstract

Cellular automata (CAs) are lattices of simple cells, whose states change according to a set of local rules. Applications range from simulating real world systems to a general platform for computation. Within the field of computer science, current research is mainly concerned with developing methods for programming CAs to solve particular tasks, as well as the pursuit of CAs with signs of complex behaviour. When it comes to I/O little has been done.

In this project the togglecount transform is introduced and investigated. It is a method for obtaining multiple outputs from a simple transformation over the temporal evolution of a CA, based on the number of state changes for individual cells during the time window of the transform. Investigation is done on elementary CAs (ECAs), and is mainly concerned with the diversity of the output, with respect to CA rule, CA size and the length of the time frame for the transform. The output variation is quantified by counting the number of unique achievable togglecount spectra, coined the "spectral diversity" of the CA.

Research is done both in a qualitative and quantitative manner. A combination of spacetime plots and spectrograms is introduced as a tool for inspecting the togglecount specter over the temporal evolution of the ECA. The specter density plot is introduced as a way of representing the full range of togglecount spectra achievable for a given ECA. Simulations are employed to find the spectral diversity for various sizes of the ECAs, for various transform window sizes.

Bounds for the achievable spectra are found to be intrinsic to the rule, but many ECAs show a relatively wide range of spectra over the set of initial configurations. Spectral diversity generally increases with increasing CA size s, and with transformation window size w when $w < s$. The ECAs are divided in three classes based on the behaviour of the spectral diversity with respect to window size, some of which has an oscillating behaviour, and a connection is found to an existing classification scheme based on the fourier transform. The results show that the togglecount transform leads to output consisting of multiple variables, and that those variables take a range of values depending on initial configuration.

Sammendrag

Cellulære automater (CA-er) er gittere av enkle celler, med et lokalt regelsett som avgjør hvilken tilstand de til en hver tid skal holde og skifte til. De brukes til alt fra simuleringer til generelle beregninger. Nåværende datateknologisk forskning på CA-er forsøker først og fremst å utvikle metoder for å programmere CAer til å løse gitte oppgaver. Det er også et innslag av generell søken etter CA-er som viser tegn til kompleks oppførsel. Når det gjelder metoder for innputt og utputt er lite blitt gjort.

Dette prosjektet lanserer og undersøker "togglecount"-transformasjonen, som er en metode beregnet på å la CA-en gi ut flere utputt-variabler. Transformasjonen beregnes over CA-ens utvikling over tid, og baserer seg på antallet tilstandsendringer for hver enkelt celle i løpet av et tidsvindu. Undersøkelsene er gjennomført på elementære CA-er (ECA-er), og omhandler fortrinnsvis variasjon i utputt med hensyn på regel, CA-størrelse og lengde av tidsvinduet for transformasjonen. Utputtvariasjonen er kvantifisert ved å telle antallet unike spekter fra transformasjonen, og dette blir her kalt "spektermangfoldet" til CA-en.

Det blir videre gjennomført både kvalitative og kvantitative undersøkelser. En kombinasjon av såkalte tidromplott og spektrogrammer legges fram som et verktøy for å undersøke hvordan togglecount-spekteret fra en CA varierer over tid. Spektertetthetsdiagrammer introduseres som en ny framstilling for å representere samtlige mulige spekter for en gitt CA. Spektermangfoldet for ECA-er av en rekke størrelser og med en rekke vindusstørrelser blir funnet ved hjelp av simuleringer.

Begrensninger for hva slags spekter ECA-ene kan produsere viser seg å være gitt av den enkelte CA-regelen, men mange ECA-er viser et stort mangfold av spekter over hele settet av globale initialtilstander. Generelt øker spektermangfoldet med økende CA-størrelse s, og med økende vindusstørrelse w for $w < s$. ECA-ene blir plassert i tre klasser basert på spektermangfold med hensyn på vindusstørrelse, noen av dem viser oscillerende oppførsel, og en kobling blir funnet til en eksisterende klasseinndeling basert på fourier-transformasjoner. Resultatene viser at togglecount-transformasjonen gir utputt som består av flere variabler, og at disse variablene tar verdier som avhenger av ECA-ens initialtilstand.

Preface

This thesis is written as the final part of a Master of Technology in Computer Science at the Norwegian University of Science and Technology. The work was conducted at the Department of Computer and Information Science.

I owe a huge thanks to Gunnar Tufte for introducing me to the world of cellular automata and for giving me an interesting project within the field. His question during preliminary talks of whether a simpler transformation than the fourier transform could be employed for cellular automata sparked a curiosity that lasted for the duration of the project. I would also like to thank Stefano Nichele, who has given me valuable technical feedback on the report, and Odd Rune Strømmen Lykkebø for lending me some of his time during the absence of Tufte.

Stephen Wolfram's 2002 book "A new kind of science" deserves an honourable mention. It has been an important inspiration for the work in this project, especially when it comes to a playful and experimental approach to investigating cellular automata through various visualisations.

Asgeir Bjørgan has provided valuable support, as well as tips on organising references and several iterations of proofreading. He has my sincere gratitude. Thanks also to Jens Abraham, for directing me to the tikz and pgfplots latex packages for making figures and graphs.

Contents

Acronyms

CA cellular automaton 3, 13–23, 25–35, 38–41, 43–47, 49, 51–54

DFT discrete fourier transform 21

ECA elementary CA 3, 14–18, 20, 21, 26, 28, 29, 32–34, 36–38, 40–46, 51–53

GA genetic algorithm 13, 20

Part I

Introduction

Invented by Ulam and von Neumann [6] almost seventy years ago, for the purpose of simulating self-replicating machines, cellular automata (CAs) have reached many fields ranging from physics to the humanities. As opposed to modern microprocessors, they are vastly parallel systems of simple locally connected cells, whose state shifting behaviour is only dependent on a local set of neighbours. Nevertheless, CAs has shown great capabilities at performing computation [47]. Some CAs are known to hold the property of universal computation [4,8,9,41], which is the ability to emulate any other general computing machine.

While implementations modelling real world systems are abundant, CAs are not yet mainstream within the field of computer science. Perhaps the lack of direct methods for programming cellular systems is to blame. Current efforts center around the use of genetic algorithms (GAs) as a tool in the search for CAs with wanted properties [2,27,32]. Examples range from specific tasks such as finding CAs duplicating a pattern given in the initial state [5] to searching for CAs showing signs of complex behaviour [31].

The study of CAs and their computational capabilities has led to multiple classification schemes and parameters for estimating complexity, of which Wolfram's classes [45] and Langton's lambda parameter [18] are well known examples. In efforts of classification, transforms of both emerging behaviour and global state have been employed. This thesis revolves around using transforms not to classify the CA but as a way to provide output.

Any useful computation requires input to be calculated on, and a way to provide the results of the computation as output. Traditionally, output from CAs mostly involve the global state of the CA at a single time step. While this is an easy way to handle output, it is rather limited. Especially for CAs with many cells rapidly changing states, choosing one time step to treat as the output may prove challenging. Further, it would be interesting to be able to extract several outputs. When using the global state as output, this raises a problem of which part of the global state to use for what part of the output. As the evolved spatial configuration of a CA is often subject to statistical properties intrinsic to the rule itself [19], spatial division of the output – and even usage of global state in the first place – may be problematic, because of the homogeneity and other limitations these boundaries give to the possible output configurations.

This thesis proposes the togglecount method as a solution to both the issue of getting multiple output variables and the issue of choosing the correct time step for the output. It is a simple transform over the temporal evolution of the CA, effectively omitting the dependence on one time step by relying on several consecutive time steps. The result is based on the temporal behaviour of the cells, rather than on one state at a more or less arbitrary time step. The output of the transform is an array of several output values, in the form of a specter.

The method is investigated in order to find out if homogeneity in the temporal domain raises an issue similar to that of spatial homogeneity. The question of whether the transform is a result of initial state or of the CA rule itself is also considered. Furthermore, the output range given by the togglecount transform is investigated with respect to the window size for the transform and to the size of the CA. As a reality check the results are compared to Wolfram's classes and Langtons λ parameter. This results in the following research questions, which form the basis for the investigations done during this project:

1. To what extent do the elementary CA (ECA) rules have intrinsic bounds for the spectra produced, and what is the nature of these bounds for varying CA size, varying window size and over the temporal evolution of the CA?
2. What determines the number of unique spectra from a CA rule, and how does the number of spectra change with changing CA size and window size for the transformation?
3. How do the number of unique spectra relate to Wolfram's classes and Langtons λ parameter?

The project description, given at the very beginning of this document, allows for a wide range of investigations. Here it has been narrowed down and interpreted as a study on the particular method of togglecount transform, a transformation suggested by this thesis. The togglecount transform is one example of "how the emergent behaviour of a cellular automata like machine can be transformed to multiple variables" for output, and it uses a form of "local frequencies as a representation of output variables." The above questions are chosen for the preliminary investigation of this transform, and the study is limited to ECAs.

The research questions are investigated by simulating ECAs, registering the count of unique resulting spectra for analysis. A qualitative study is performed by introducing two visualisations of the spectra. The first, "waterfall plots", consists of a spectrogram plotted along with the temporal evolution of the CA itself, allowing for investigations of the temporal evolution of the specter. The second, "specter density plots", is a collection of superimposed spectra, displaying the occurring values for each variable of the specter.

The rest of the thesis is organised as follows: In Part II, CAs are introduced in Section 1 before computation in the context of CAs is handled in Section 2. Prior research involving transformations are summarised in Section 3, and some mathematical and statistical concepts are briefly presented in Section 4.

Part III introduces the togglecount transform in Section 5, as well as the use of waterfall plots and specter density plots in Section 6 and Section 7. An overview of the experiment approach is given in Section 8, and software written for the purpose of CA simulation, plotting and data handling is described in Section 9.

Part IV starts with a brief discussion of theoretical limits in Section 10. A qualitative study of ECA spectra using waterfall plots and specter density plots is presented in Section 11. Spectral diversity data is investigated in Section 12, with respect to both the window size for the transform and the size of the CA.

In Part V, findings are summarised in Section 13. An outline for further investigation is given along with some comments on preliminary data in Section 14.

Part II

Background

1 Cellular automata

1.1 Definition

CAs are $\mathfrak{n}$-dimensional integer lattices of cells, where each cell at any given time t is in one of a set of k possible states. Cells are updated in discrete time steps using an update function $\mathbf{F}$ that computes the next state for a cell given the current states of the neighbouring cells. The calculation synchronised, i.e. all cells change from one time step to the next at the same time.

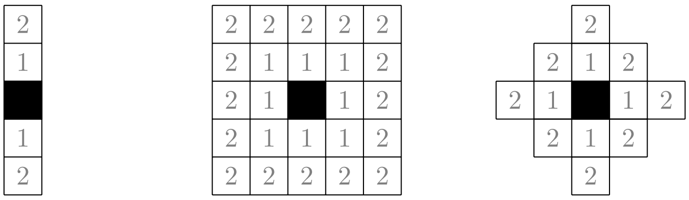
For one-dimensional CAs, the neighbourhood consists of the cell itself and a range of r cells in each direction. This results in a neighbourhood size of $2r+1$. Formally, the behaviour of a one-dimensional CA can be expressed as

$$a_i^{(t)} = \mathbf{F}[a_{i-r}^{(t-1)}, a_{i-r+1}^{(t-1)}, \ldots, a_i^{(t-1)}, \ldots, a_{i+r}^{(t-1)}], \tag{1}$$

where $a_i^{(t)}$ is the value of cell i at time step t for an automaton ruled by the function $\mathbf{F}$, with a neighbourhood range r. [45] The update function is also referred to as the rule of the CA. [47] The set of one-dimensional CAs with rules of range $r = 1$ and 2 possible states are called elementary CAs (ECAs). [44].

1.2 Neighbourhoods

For two dimensions the Moore [29] and the von Neumann neighbourhoods are commonly used. Both employs a combination of the range r and a distance metric on the lattice. The former uses the Chebyshev distance as the distance metric, the latter uses the Manhattan distance. Both of these metrics can be used for any number of dimensions. Examples for $r = 2$ in one and two dimensions are shown in Figure 1.



(a) One dimension (b) Moore (2D) (c) von Neumann (2D)

Figure 1: Some examples of neighbourhoods. Each subfigure shows the neighbourhood of range $r = 2$ for the black cell. The distance to the black cell is indicated for the other cells in the neighbourhood. For one dimension the Moore and von Neumann neighbourhoods are equivalent.

1.3 Visualisation

The evolution of one-dimensional CAs can be plotted in two dimensions using spacetime plots, as shown in Figure 2. Here the cells are represented as squares, whose colour indicates state. Time flows downward on the plot, and the rows show the spatial configuration of the CA for consecutive time steps. One column in the plot shows the temporal evolution for one cell.

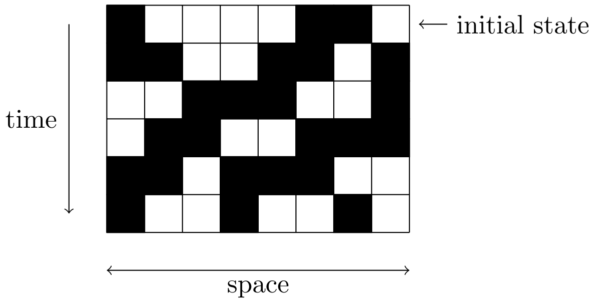


Figure 2: Spacetime plot. Excerpt from larger plot of ECA rule 30. Each row is a snapshot of the CA for one time step. Each column is the temporal evolution of one cell.

CAs of higher dimensionality are harder to represent in the plane, due to the obvious lack of dimensions. One-dimensional series of cells from higher-dimensional CAs can be plotted using the same spacetime plots as for one-dimensional CAs [35], providing some insights into the general temporal behaviour. Other methods include state transition diagrams [34] and 3D renderings of spacetime plots for two spatial and one temporal dimension. [47] Another option is to use animation to show the evolution of the CA over time, as has been done for numerous implementations of Conway's Game of Life.

1.4 Representation

The rule of a CA can be represented in different ways, such as using boolean logic [44] or implementation as program code [5]. In this thesis, a rule is represented in tabular form [20] and numbered according to the numbering scheme presented by Wolfram. [44]

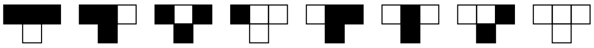
For table representation, each combination of neighbour states is treated as an integer value whose base is the number of CA states, and sorted numerically. The next state of the center cell is assigned to each combination, resulting in a table of next states indexed by the local neighbourhood. An example table for ECA rule 110 is shown in Table 1.

Using Wolfram's naming scheme [44], the list of next states is converted to an integer similarly to the conversion from neighbourhood states to table indexes. The ordering of digits is such that the next state of the first neighbourhood index is the least significant digit in the resulting rule number in base k. For the example in Table 1, the table gives binary rule number 0b01101110, which is 110 in decimal. An alternative way to show the same naming scheme is

Neighbourhood	Next state
0 0 0	0
0 0 1	1
0 1 0	1
0 1 1	1
1 0 0	0
1 0 1	1
1 1 0	1
1 1 1	0

Table 1: Table representation of ECA rule 110 = 0b01101110. The neighbourhood is used as an index to look up the next state of the center cell.

provided in Figure 3. The possible state combinations for the neighbourhood and the resulting next state for the center cell are represented using squares as in spacetime plots. Below are the numerical representations of the next center cell state, forming the binary representation of the rule number.



0 b 0 1 1 0 1 1 1 0 = 110

Figure 3: Rule numbering. ECA rule 110 (binary number 0b01101110). First row indicates neighbourhood states, second row indicates next center cell state. Binary representation of rule number at the bottom. Based on similar figures from Wolfram [44, 47].

1.5 Attractors, transients and Garden of Eden states

The k^s configurations of the cells of a k-coloured CA of size s constitutes the set of possible global states for that CA. As this set is finite and the evolution of the CA is deterministic, at some point in the evolution of the CA it will enter a previously visited global state. The case of a state whose next state is the same is called a point attractor, while a series of states in a loop is called a cyclic attractor.

States that are part of an attractor are called attractor states. States that can only appear as initial states, i.e. states with no preceding states, are called Garden of Eden states. [29] States that are neither attractor states nor Garden of Eden states are called transient states.

The part of the CA's evolution which is either in Garden of Eden or in transient states is called the transient part, and the transient time is the number of time steps needed for the CA to reach an attractor. The attractor length is the number of distinct states in an attractor.

1.6 Classification

1.6.1 Overview

Different CAs, even of the same rule space, show highly different behaviour. [44] Several classification schemes for CAs have been proposed [13,15,17,20,48], two of which [18,45] will be presented in Sections 1.6.2 and 1.6.3. A fourier transform based classification [33] is summarised in Section 3.4.

1.6.2 Wolfram classes

Wolfram classifies CAs in four classes: [45]

Class I reaches a uniform point attractor after only a few time steps. The attractor is the same for almost all initial states.

Class II quickly reaches a state of repeating localised structures, i.e. individual portions of the CA do not interact after a short transient, and each structure has a short period.

Class III shows lasting random-like behaviour, although some small structures are always seen as part of the randomness.

Class IV shows localised moving structures with intricate interactions. While the structures themselves are fairly simple, their interaction is complex and random-like. It is postulated that class IV automata hold the property of universal computation [47].

A listing of the classification of the ECA rules is provided in [23].

1.6.3 Langton's lambda parameter

The λ parameter, introduced by Langton [18], is the proportion of neighbourhood configurations leading to a chosen quiescent state, according to the update function of a CA. The formal definition from Langton is

$$\lambda = \frac{K^N - n}{K^N}, \tag{2}$$

where K is the number of colurs, N is the size of neighbourhood and n is the number of transitions to the quiescent state. $\lambda = 0.0$ when all neighbourhood configurations leads to the quiescent state, and $\lambda = 1.0$ when the quiescent state is absent from the transition table. While λ is a good parameter for larger CA rule spaces, only a rough correlation between λ and CA dynamics is found for ECAs. [18]

2 Computation in CAs

2.1 The notion of computation

Mitchell et al. [28] describes three different definitions of “computation” in the context of CAs. The first definition is that the CA performs a “useful” computational task. Input is taken in the form of an initial configuration, output is the arrival at some “goal” state or the state after a given number of time steps. The second definition is that the CA is capable of universal computation and can emulate a programmable computer given the right initial condition. The third definition is intrinsic computation, measured as the amount of computational behavior in the CA independently of its “usefulness”.

According to Langton, [18] the CAs can be viewed either as a machine or as a physical layer on which to build a machine. In the first case, the machine is the CA itself while the initial configuration is pure input. In the second case, the CA provides a simulated physics environment in which an embedded computer can be built. The initial configuration then consists of both machine and data. Those two views correspond roughly to the first two meanings of computation as described by Mitchell.

Universal computation has been proven for several CAs, by von Neumann [41], Conway et al. [4], Codd [8] and others [1, 9, 10, 21, 40].

In this thesis, the CA is primarily regarded a function mapping the set of initial configurations to a set of outputs. The form of computation that is considered interesting is the work done by the CA to perform this mapping. As in the first view from Langton, initial configuration is treated as pure input. The limitations of the output set are extended beyond those of the first definition from Mitchell, so that it covers any type of data that can be extracted from the CA during its evolution. This includes any transformation of the spatial configuration or the temporal behavior of the CA, allowing for exploration of new types of output.

2.2 Frequently used problems

2.2.1 Overview

Among the most frequently used problems within the field of CAs as computing machines are pattern generation, pattern replication, density estimation and the firing squad synchronisation problem. Pattern generation is the problem of finding a CA that produces a given pattern from a given, simple, initial configuration. The problem of pattern replication is that of finding a CA reproducing multiple instances of a small pattern given as an initial configuration. Both the density estimation problem and the firing squad synchronisation problem are somewhat more complicated, and are described in the following subsections.

2.2.2 Density estimation

The problem of density estimation is the decision problem of deciding whether p_0, the ratio of zeroes in an initial configuration, is above or below a given threshold p_c. The output is a point attractor after M time steps, consisting of only 0s if $p_0 < p_c$ and only 1s if $p_0 > p_c$. The case of $p_0 = p_c$ is undefined. [27] One such problem is the $p_c = 1/2$ task for binary CAs, which has been used in studies of CA and GAs. [28, 36]

2.2.3 Firing squad synchornisation

The firing squad synchronisation problem is the task of constructing a one-dimensional CA of range $r = 1$ and arbitrary size s, where all cells are to enter a "firing state" F simultaneously and for the first time. With the exception of one cell at the far end of the automata, all cells start out in the quiescient state L. The far end cell start in the "general" state G. [22, 25] Since its introduction, several solutions have been found. [3, 12, 24, 26, 42, 49]

3 Prior research using frequencies

3.1 Overview

Various researchers have investigated fourier spectra of ECAs, both on evolved spatial configurations [19, 39] and for temporal behaviour [30, 33]. In the classification cases the specter is used as a fingerprint for classifying the CA under the assumption that properties of the specter are dependent not on the initial state but on the CA rule itself. In this section, three studies of ECAs and their fourier spectra are briefly presented.

3.2 Power spectra and regular languages

Dependence on rule for statistical properties of the attractor states is explicitly asserted by Li [19], although the only sources for this statement are "W. Li (unpublished) and H. Gutowitz (private communication)." There does not seem to be any published studies on spectral diversity with respect to initial state.

The article features fourier transforms of the evolved spatial configuration of all the ECAs, save for the eight Wolfram class I rules[*].

The article relates CA attractors and regular languages, and provides a method to calculate the power spectra of regular languages. The generation of characteristic power spectra for a couple of ECA rules are provided as examples.

[*] Wolfram class I rules revert quickly to a uniform state, of which there is no use of spatial power spectra. They would all be zero everywhere but for frequency 0.

3.3 Spectral equivalence classes

In the paper by Ruivo and Oliveira [39], discrete fourier transforms of evolved spatial configurations of ECAs are used to relate the rules based on similarities in the observed spectra. Both periodic and non-periodic boundary conditions are investigated, periodic first. The spectra are based on the evolved global state from 1000 initial configurations of size 1024, after 200 time steps. This choice is validated by running additional simulations for some rules, using larger CAs and more initial states, with the resulting spectra remaining approximately the same.

Using a euclidean distance metric for the difference between the various spectra, 59 spectral classes are detected among the ECAs. Of these, some classes have resembling spectra not deemed equal by the first application of the metric. Using the distance metric further, 19 groups of connected classes – as well as connections among these groups – are found. The final result is a connected graph of the groups, and connected graphs of the spectral classes for each group.

The same method employed for non-periodic boundary conditions results in different spectral equivalence classes than for the case of periodic boundary conditions.

3.4 Classification and $1/f$ noise

Ninagawa [33] investigates the discrete fourier transform (DFT) of the temporal evolution of ECAs, in particular that of rule 110. All 88 equivalence classes are investigated and sorted in categories based on their power spectra given by the DFT.

Category 1 show very low power density, with a possible peak at the highest frequency. The subcategory with no peak is the set of Wolfram class I rules and a subset of the class II rules which have point attractors. The one with the peak is a subset of the class II rules whose localised structures toggles for every time step.

Category 2 have spectra with broad-banded noise. It is divided into two subclasses, where 2-A show greater variation in the specter than 2-B. The latter is close to white noise (i.e. a flat specter, but of much larger strength than those of category 1.) The 2-B rules are the same as the Wolfram class III rules[†].

Category 3, coined the "Power law spectrum", consists of spectra where the power density is inversely proportional to the frequency. This is also called "1/f noise". [16] The ECA rules 54, 62 and 110 are classified in this category. These rules are investigated for long time intervals, and rule 110 is shown to result in $1/f$ noise even for the longest runs. For rules 54 and 62, the $1/f$ noise is only observed for limited observation lengths. It is proposed that $1/f$ noise is related to universal computation.

[†] Although 2-B is presented as coinciding with class III, the list also contains class IV rule 106.

Two rules do not fit in any of the three categories, and are called “exceptional”. These are rules 73 and 204. The former has a spacetime plot of stable walls dividing areas of chaotic behaviour, resulting in myriad local periods each leading to a peak in the power spectrum. The latter is the identity rule, whose specter is zero everywhere but at frequency 0.

4 Mathematical and statistical foundations

4.1 Functions

The mathematical notion of a function is important to the project, and some formal terminology is quickly described here. The most important terms for this thesis are shown in Figure 4.

The function $f : A \to B$ is a mapping from the non-empty set A to the non-empty set B. Every element a in A is mapped to an element b in B, and the notation $f(a) = b$ expresses this mapping for a particular a. The element b is the image of a, and a is a preimage of b. While b is the sole image of a, b may have multiple preimages. The set A is called the domain of f, while the set B is called the codomain of f. The range of f is the set of all images of elements of A, in other words the subset of B where every element is mapped to by at least one element of A. [38]

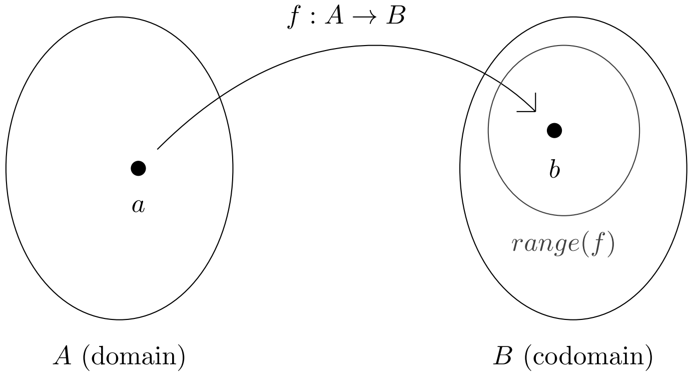


Figure 4: Functions. Based on similar figures from Rosen [38].

To avoid confusion with the range r of a CA rule, the range of a function will in this thesis be called “function range”. The size of the function range when applied to a specter generating CA is coined the “spectral diversity” of the CA.

4.2 Necklaces

When using periodic boundary conditions for a one-dimensional CA, the topology of the CA becomes a circle. For some purposes, all rotated versions of a global state now can be regarded the same.[‡] This is the concept of equivalence classes of k-ary strings over rotational shift, or k-ary necklaces. [7]

The number of binary necklaces is given by the equation

$$Z_s = \frac{1}{s} \sum_{d|s} \varphi(d) 2^{s/d} \tag{3}$$

where s is the length of the string, $d|s$ denotes the divisors d of s and φ is Euler's totient function [11]. Values for the first few string sizes s beginning at $s = 1$ are 2, 3, 4, 6, 8, 14, 20, 36. In contrast the series 2^s of binary global states not counting equivalence over rotation is 2, 4, 8, 16, 32, 64, 128, 256 for corresponding values of s. It is clear that rotational equivalence noticeably limits the number of global states where applicable.

An efficient algorithm for sequentially generating the set of k-ary necklaces of size s is provided by Fredricksen and Kessler. [11]

4.3 Multichoose

The number of possible spectra when counting the number of toggles for each cell of a size s CA over a window of size w is limited by both s and w. It can be viewed as a combinatorial problem of picking one of w possible frequency counts s times. The number of ways to do this, also known as "w multichoose s", can be calculated using the equation:

$$\left(\!\!\binom{w}{s}\!\!\right) = \binom{w+s-1}{s} = \frac{(w+s-1)!}{s!(w-1)!} \tag{4}$$

4.4 Correlation

Linear association between two statistic variables X and Y can be estimated using the Pearson product-moment correlation coefficient r:

$$r(X,Y) = \frac{S_{xy}}{\sqrt{S_{xx}S_{yy}}}, \tag{5}$$

where

$$S_{xy} = \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}). \tag{6}$$

Here, x_i denotes the i-th value of the variable X, and $\bar{x}$ denotes the arithmetic mean of the values in X.

r takes a value in the range $[-1, 1]$, where 1 indicates a perfect linear relationship between X and Y, 0 indicates no linear relationship and -1 indicates perfect

[‡] As will become clear in Section 10, this holds for initial states but not for evolved global states.

inverse linear relationship. Further, r^2 describes the proportion of the total variation of the values in Y that can be attributed a linear relationship between X and Y. [43]

Part III

Methodology

5 Togglecount

Introduced in this thesis is the togglecount method for transforming the temporal evolution of a CA to output in the form of a specter. The transform is fairly simple, as shown in Figure 5. Top left is the spacetime plot of a cellular automata. Each state change of a cell – marked in the figure using red arrows – is referred to as a "toggle". For each of the s cells of the automata, the number of toggles within a time window of w time steps are counted, as shown bottom left in Figure 5. An array of the number of occurrences for each toggle count constitutes the final specter, shown in Figure 5 as a bar chart (top right) and corresponding array (bottom right).

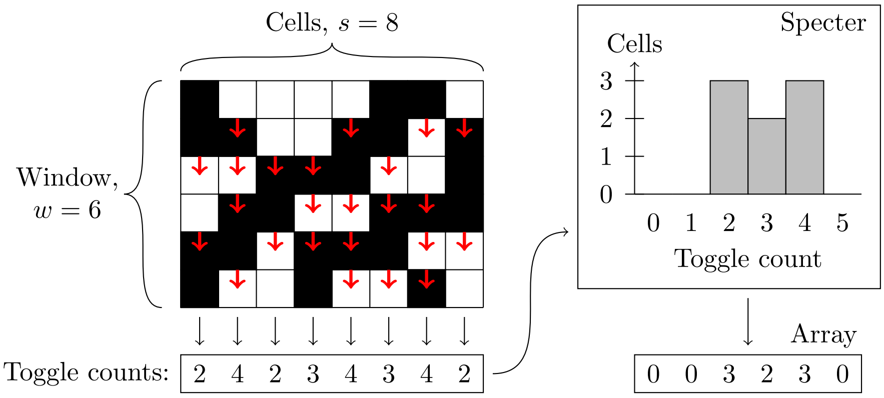


Figure 5: Togglecount. The number of state changes during a time window of size w is counted for each of the s cells (left), before the counts are accumulated to form a specter (right). The specter is visualised as a bar chart (top right) or listed as an array (bottom right). The specter consists of w integer values whose sum is s.

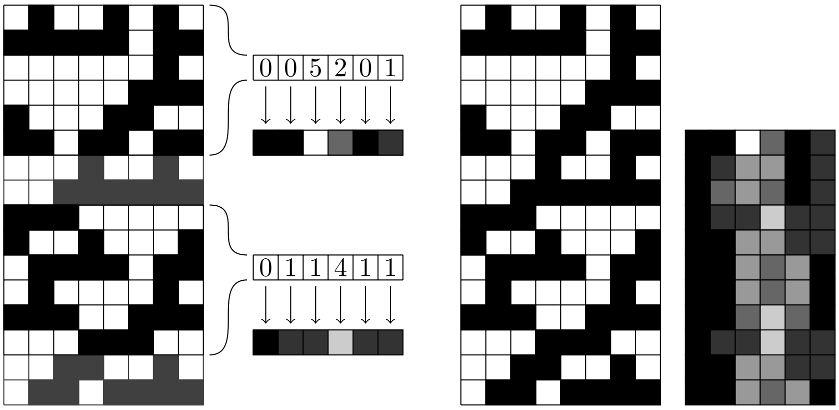
Observe that the specter consists of w non-negative integers whose sum is s. Each integer in the array is a cell count, i.e. the number of cells in the CA with a given number of toggles within the window of the transformation. The first integer is for zero toggles, the last integer is for $w - 1$ toggles. The number of cells is referred to as being in the "cell count domain", while the toggle count with which it is associated is referred to as being in the "frequency domain". When plotted, each toggle count in the frequency domain is associated with a number of cells in the cell count domain.

The togglecount transform addresses the issue of obtaining multiple outputs from a CA by providing an array of integer values. As it is computationally simple it allows for easy implementation in both software and hardware, which constitutes an advantage compared to the fourier transform.

As togglecount is a transform over the temporal behaviour of the CA, for a sufficiently large window size the entire cyclic attractor will contribute to the result. In contrast, choosing one time step from a cyclic attractor yields output depending on the phase of the attractor. It is thus less of a problem of choosing the one "correct" time step from the CA. It also allows for using the transient phase as a basis for output. However, as all point attractors show the same static behaviour of no toggles, any point attractor will yield the same transform.

6 Waterfall plots

Inspired by spectrograms, used for instance by amateur radio operators and in sound editing software, a new and combined approach is used in this thesis for visualising the togglecount spectra of an evolving CA over time. Spectrograms are plots showing power density along one axis and time along the other. The idea of the waterfall plots of this thesis is to plot the spacetime plot of a CA aligned with a spectrogram generated by the togglecount transform over the same CA using a sliding window on the same temporal evolution. The process of creating the plot is shown in Figure 6.



(a)

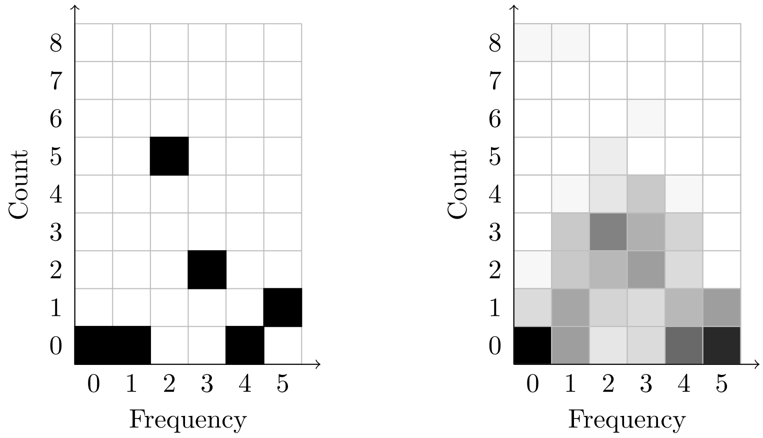
(b)

Figure 6: Waterfall plot, $w = 6$, ECA rule 30. (a) Left, spacetime plot; right, togglecount transforms of two time frames, represented as numbers and as shades of gray. (b) Waterfall plot: Graphical representations of the togglecount transform for each window position, aligned with the spacetime plot.

In Figure 6a, two togglecount transforms are generated from the spacetime diagram to the left, and represented as shades of gray. The series of one-dimensional specter plots forms a continuous flow of specter data alongside the spacetime plot, as shown for a range of time steps for the same spacetime plot in Figure 6b. The plots are arranged so that each line of specter is besides the row in the spacetime plot which is the last row of the window for that specter. Using this representation, it is possible to get an intuition for how the togglecount specter changes with the temporal behaviour of the CA.

7 Specter density plots

To visualise the spread in resulting spectra from the togglecount transform, specter density plots are generated by superimposing multiple pixelated two dimensional specter plots. A pixelated specter plot for the array $[0, 0, 5, 2, 0, 1]$ is shown in Figure 7a. This is the same as the topmost specter from Figure 6a in Section 6. For each frequency, i.e. toggle count, a black square marks the number of cells having that number of toggles.



(a)

(b)

Figure 7: Specter density plot, rule 30, CA size $s = 8$, window size $w = 6$. (a) One pixelated specter. (b) All spectra, pixelated, faded and superimposed.

The final specter density plot in Figure 7b is generated by superimposing the set of pixelated specter plots from all initial configurations. Each occurrence of a black cell in a pixelated specter plot increments the "blackness" of the corresponding cell in the specter density plot. The cell of the highest blackness is painted black, and the rest are painted in a shade linear with respect to blackness. This means the distribution of cell counts for each frequency can be read from the plot.

For large CA sizes and window sizes, cells coloured by very few spectra become very faint to the point of invisibility. Because of this, contribution is capped at a certain maximum level, so that the contributions from most spectra are still visible in the density plot. For large w and s the result is still some fading indicating areas of fewer spectra, but information on distribution above the threshold is lost. The plot still shows what counts are achievable, which is the main objective.

8 Experimental approach

The purpose of the project is to find out whether the togglecount transform is a feasible method for getting multiple variables from a CA. The most alarming concern is whether there is enough diversion among the resulting spectra to make use of the multiple outputs generated. As stated in research question 1 (see Part I), intrinsic bounds on the spectra produced is to be investigated both with respect to CA size s and transform window size w. Specter density plots are created for several combinations of s and w, and organised in tabular form, in order to map the bounds and compare them.

Also of importance is the limitations on and changes in spectral diversity as window size and CA size changes, as stated in research question 2. Theoretical limits for spectral diversity as a function of CA size and window size are calculated based on the number of possible initial configurations and the number of possible toggle count spectra from a given CA size and window size. Simulations are run to gather data on the spectral diversity for all ECAs over a range of CA sizes and window sizes. The collected data is plotted and compared to theoretical limits. Different behaviour of the spectral diversity with respect to CA size and window size is investigated, with the aid of specter density plots where needed.

Relation to existing classification schemes are considered during the investigations of the first two questions, and also employed in the efforts of explaining observed behaviour.

9 Programs

9.1 Program overview

Various programs have been created for this thesis, to generate plots and data through simulation and to analyse generated data. Most simulation is done with the program “codomaincounter”, which is a multithreaded CA simulation tool. Its purpose is to find the spectral diversity of a set of CAs, and to output this number labeled with what particular CA it is associated with. The data is fed to an sqlite database from where data can either be browsed directly using SQL or be extracted by other programs for analysis.

The program “dbextract” is a small utility program written to extract data points from the sqlite file, and arrange them space separated in lines of plaintext.

Programs working on the data from dbextract uses standard in and standard out for data handling, so that steps in the process of analysis and presentation can be performed either by piping data directly from program to program or by writing to and reading from files holding intermediate representations and results. These programs are the following:

“graphmaker”: Generates scatter plots or line plots.

“statistics”: Calculates statistical properties of the data.

In addition, two programs are used for generating images. The first program is “caprint”, a predecessor to codomaincounter. Originally written to print space-time plots of ECAs, it ended up being changed repeatedly to perform several different tasks, before ending up writing low resolution image files of superimposed spectra. Although intended to be what codomaincounter became, the combination of architectural limitations and the usability of the superimposed spectra plots led to the decision to freeze program development, letting the program do what it does the best. Cpaprint is now a specter density plot generator for ECAs. At an earlier stage of development, it produced waterfall plots, but this capability is regretfully lost during development.

The second image generating program is “spacetime”, which generates spacetime plots of ECAs.

9.2 Main simulation tool

9.2.1 Objective

The CA simulator named codomaincounter is core to the project. Its main objective is to find the spectral diversity for a given combination of parameters for the CA and for the transform. It is configured to run a batch of simulations for a set number of randomly chosen CAs of a given number of colours and neighbourhood range. Each CA is simulated over a range of CA sizes and for a set of window sizes for the transformation. The results are written to an sql file, for import to an sql database.

9.2.2 Main thread

The main thread of the program generates a queue of assignment objects, each containing the information needed for a simulation run: Number of colours, neighbourhood range, rule number, CA size and a list of window sizes. Simulation is outsourced to worker threads whose task is to find the spectral diversities for the given CA for all listed window sizes. The results are saved in the assignment object, before the worker thread exits.

After having initiated a configurable number of worker threads, the main thread enters a state of periodically checking for finished assignments. Data points from a finished assignment are written to file, and a new worker thread is started with a new assignment provided the queue is not empty. The program exits when the assignment queue is depleted and all data points have been handled.

9.2.3 Worker threads

The worker threads simulate a CA according to the specification given in an assignment object. A CA object is initialised with size, rule number, range and number of colours. A rule table is calculated and represented as an array inside the CA object. Member functions provides access to setting the global state, getting a pointer to the current state and increment the CA to next time step. Boundary conditions are handled by using a larger size for the global state internally than visible outside the class, so that updating to the next time step is identical for all cells. Outside cells are then updated according to cyclic boundary conditions. This internal representation of current state simplifies the logic for calculating the next time step, and allows for easier future implementations of additional choices for boundary conditions. The CA class does not hold any temporal backlog.

A necklace generator using the algorithm from Fredricksen and Kessler [11], mentioned in Section 4.2, provides the initial states. It is implemented as a class, with member functions for accessing the current necklace, to increment to the next necklace and to reset. When incrementing from the last necklace, the object is reset to the first necklace. A member function returning whether the current state is the reset state allows for looping through the full set of necklaces with this as a looping condition. The necklace class handles arbitrary sizes and number of colours.

A two dimensional array holds the spacetime plot of the CA. This is reused for each run from initial state. It is arranged so that the temporal domain is contiguous in memory. This allows for a function performing a transform on the temporal behaviour of a single cell while being agnostic to the overall CA size. While this may also be beneficial for speed, optimisation was not a main reason for this choice. Implementing a togglecount function that only needs a single pass over each cell while calculating spectra for multiple window sizes would probably lead to better speed improvements if optimisation was the main objective.

The spectra, once aggregated from the toggle counts of individual cells, are saved as vectors in map structures for each window size. When simulation from all initial states are finished, the sizes of these maps constitutes the spectral diversities of the CA for the given window sizes. Results are written to the assignment object and a “finished” flag is set before the thread exits, leaving the task of registering the results to the main thread.

Part IV

Results and Discussion

10 Theoretical expectations

10.1 Theoretical limits for spectral diversity

Before analysing the actual spectral diversities, some theoretical limits can be calculated.

In order to allow for a consistent temporal transform, each subsequent state has to be spatially located according to the one preceding it. This means all k^s possible global states are eligible during the temporal evolution of the CA. However, in the final transform only the behaviour of a cell is of importance, not its spatial location.

This means the whole spacetime plot from an initial configuration can be regarded as belonging to an equivalence group over rotation. All rotated versions of an initial configuration will always result in the same specter, as long as all other states during the temporal evolution of the CA are treated unequal over rotation. Equality over rotation is the concept of necklaces, introduced in Section 4.2.

Using Equation 3 from Section 4.2, the number of distinct initial configurations for the range of CA sizes have been calculated and are presented in Figure 8. Also presented in Figure 8 are limits for the number of distinct spectra for three window sizes. The number of possible spectra is the number of ways to distribute s cells over w toggle counts, for CA size s and window size w. These are calculated using Equation 4 from Section 4.3.

The size of a function range can not exceed neither the size of the domain nor the size of the codomain. In the case of the CAs concerned in this thesis, this means the spectral diversity can not exceed the number of distinct initial configurations nor the number of conceivable togglecount spectra.

When the lower bound on spectral diversity is given by the number of distinct initial configurations, the CA will be denoted as input bound. When the lower bound on spectral diversity is given by the number of possible spectra, the CA will be denoted as output bound. As can be seen from Figure 8, the CA is input bound for small s and output bound for large s. The least required s for a CA to be output bound increases with window size.

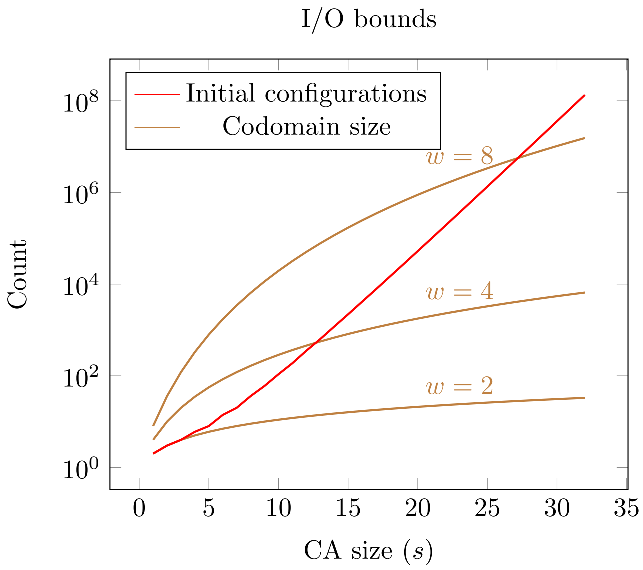


Figure 8: Number of unique initial states and number of possible spectra for window sizes $w \in \{2, 4, 8\}$, for CA sizes in the range $[1, 32]$. (Note the logarithmic scale on the y axis.)

11 Qualitative study of ECA spectra

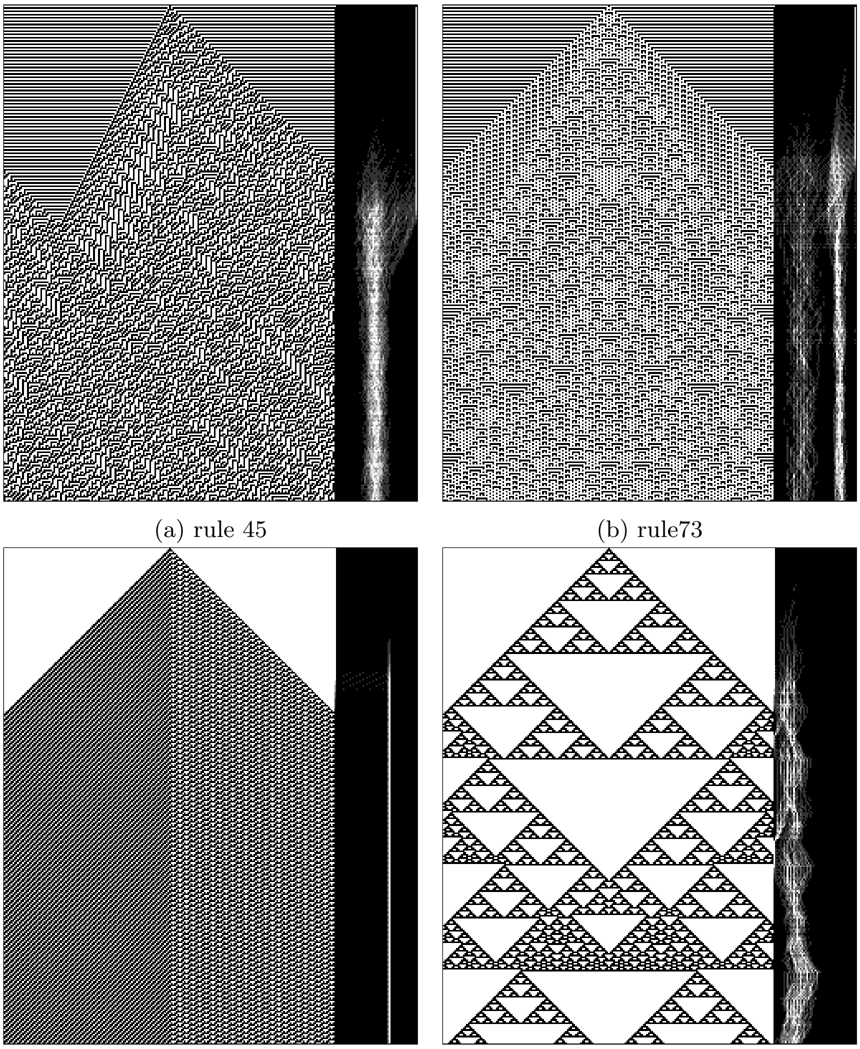
11.1 Waterfall plots

As a preliminary investigation, waterfall plots of all ECAs are considered. The parameters for the experiment are CA size $s = 200$, a sliding window of $w = 30$ and runs for 300 time steps from an initial condition of one black cell. The plots show variation between different rules, as shown in Figure 9.

While many rules reach either a point attractor or a short cyclic attractor leading to constant or near-constant spectra, others show more interesting behaviour. Rule 62 in Figure 9c is an example of the former, although untypical. Most rules falling in this category end up having spectra consisting of almost all zeroes, with the exception of 0 or 29 toggles (i.e. either point attractor or all cells toggling for every time step.)

Wolfram class III rules 45 and rule 73, in Figure 9a and Figure 9b respectively, show stable spectra of one and two peaks, with some minor variation. This is also seen for other Wolfram class III and IV rules. While the spectra may possibly be very different for other initial states, the randomness of class III automata would suggest the waterfalls should show more variation over time if this was the case. Rather, the plots seem to confirm that class III rules have random qualities, resulting in uniform spectra resembling normal distributions.

Some rules, like rule 126 in Figure 9d, show more variation in the specter over time. This may indicate a potential for greater spectral diversity. However, rapid fluctuations between spectra also leads to sensitivity to the placement of



(a) rule 45

(b) rule73

(c) rule 62

(d) rule 126

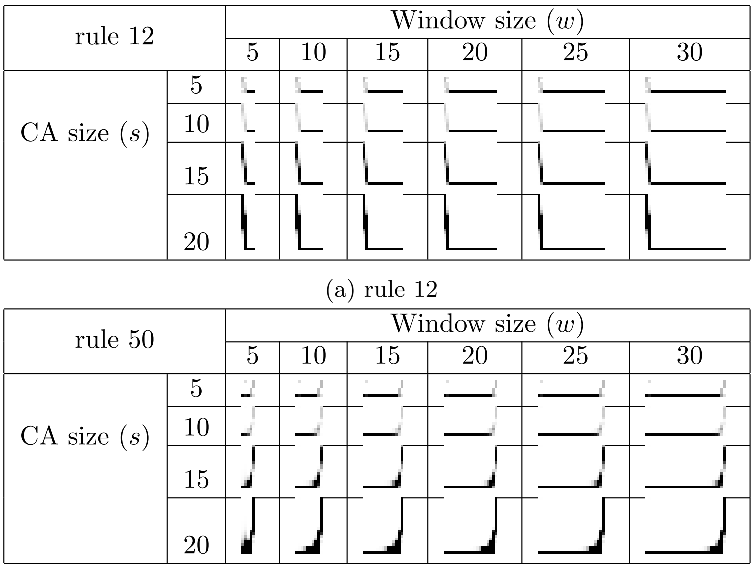
Figure 9: Four ECAs (left) and their waterfall plots (right). The waterfall is calculated using the togglecount transform and a sliding window of size $w = 30$. The CAs are of size $s = 200$ and of periodic boundary conditions. (a) Wide specter with apparent little variation. (b) Two peaks of unequal width and strength. (c) Strong and narrow stationary peak. (d) Variation in the frequency domain over time. Generated by software written for this thesis.

the transform window in the temporal domain. This sensitivity is a problem togglecount was thought to avoid. It is only occurring for a small number of rules, which indicates it being a minor issue. Looking up spectral diversities in the data sets from full simulations reveals that the rules in question are ranking fairly high in spectral diversity, although several rules have considerable higher values.

11.2 Specter density plots

11.2.1 The general case

When investigating specter density plots, the most striking feature is the similarity between different plots of the same rule. When varying the window size w and CA size s, the plot is stretched in the frequency domain and cell count domain respectively. This becomes evident when arranging specter density plots in tabular form, as is done for two rules in Figure 10. Shown in this figure are rules 12 and 50, which have very low spectral diversities. The plots show thin lines for all w and s, showing that a very limited set of spectra is present. Stretching behaviour is typical for almost all ECAs, while about a third of the rules are constrained to a specter variation of similar magnitude.



(a) rule 12

(b) rule 50

Figure 10: Typical specter density plots of ECAs. Small variation between spectra results in a fairly thin line, and the overall shape of the plot is stretched to fit within the range of frequencies and number of cells. Almost all ECAs show similar stretching behaviour, and about 1/3 of them have similarly restrained ranges of the underlying spectra. Generated by software written for this thesis.

The fact that 1/3 of the rules have a very limited specter space, as proved by the specter density plots appearing as thin lines, suggests that intrinsic bounds for the spectra produced is indeed a severely limiting factor. For this portion of the rules, the initial configuration have little influence over the resulting specter. As the spectra of different rules show different shapes, a mapping from rule to specter is evident. At least for a third of the rule space of ECAs the specter show much stronger dependence on rule than on initial configuration.

11.2.2 More relaxed bounds on spectra

When investigating the same rules as in the waterfall plots of Section 11.1, it becomes clear that the spectra shown in the waterfall are typical for these rules. Specter density plots for these rules over a range of s and w are arranged in Figure 11. The bounds provided by these rules are less strict than the narrow bounds given by the rules of Figure 10.

Rule 45 (Figure 11a) shows all plots resembling normal distribution. Rule 73 (Figure 11b) shows two peaks, but also a general floor of activity over the full range of frequencies. The peak of rule 62 (Figure 11c) is pronounced, and as w is increased the density plot holds the general shape while shifting somewhat towards higher toggle counts. Rule 126 (Figure 11d) has somewhat complex plots with small detailed "islands" of high cell counts at near zero frequencies. The scaling over both w and s show the characteristic stretching, apart from the entire line of $s = 15$. The shape is skewed towards lower frequencies, and seems to "lock" for $w \geq 20$ leaving a tail of no toggle counts above a certain threshold.

The deviations from linear scaling of the plots of rules 62 and 126 may seem puzzling at first. For rule 62, a rule of Wolfram class II, the typical attractor consists of stationary localised structures, most of which have period three, with a backdrop also of period three. This explains the location of the peak at 2/3 along the frequency domain. As for the broader base, this may be a trace from the transient period, with intermediate structures whose cells toggles with other frequencies.

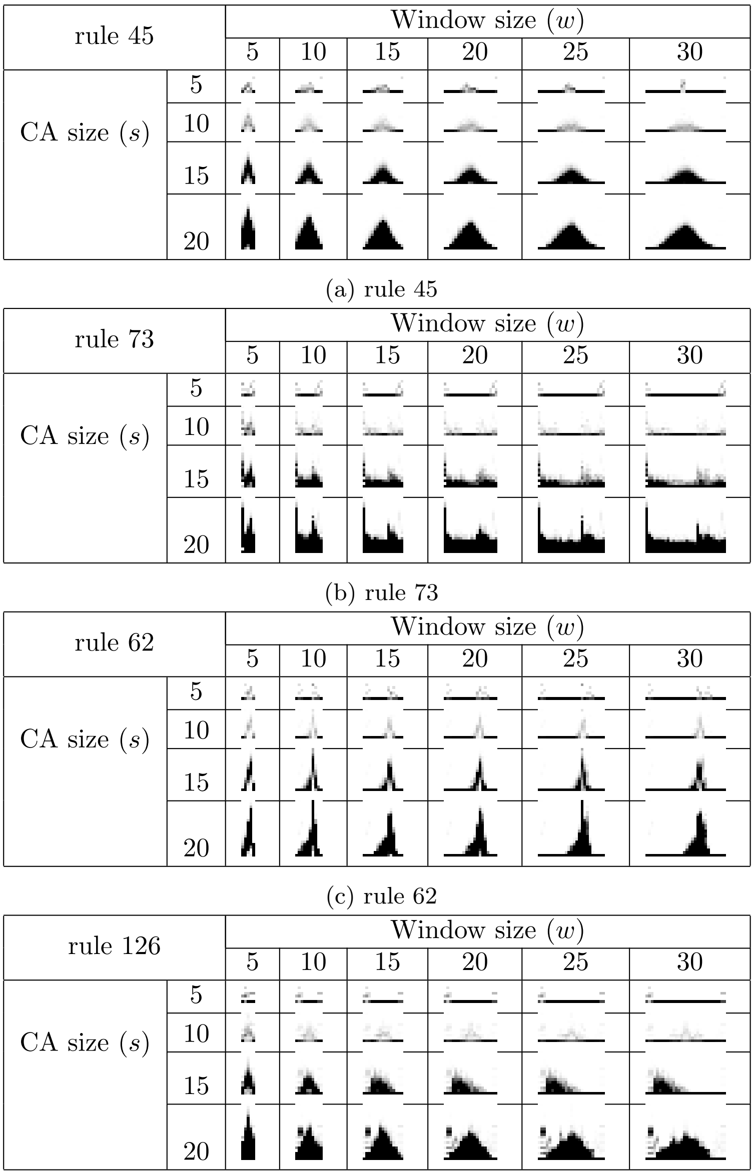
The lack of scaling of the $s = 15$ spectra of rule 126 is explained by the rule entering point attractors around $w = 20$ for this particular CA size, thus fixing the cell counts at low frequencies. Another example of a CA with similar behaviour is rule 60 for $s = 20$. As is the case for rule 126, the specter density plot is skewed and a tail of cell count zero emerges above a certain frequency threshold.

Similar tail phenomena as for rules 62 and 126 are observed for several rules of Wolfram classes I and II. As class I quickly leads to a quiescent state, the tail appears towards higher frequencies just as for rule 126. Wolfram class II rules whose attractor is typically a point attractor show the same. Class II rules whose attractor is a uniform frequency for all cells show specter density plots shifting towards higher frequencies, while leaving an increasingly longer zero tail at lower frequencies. Because of the short transients and the same typical behaviour over all CA sizes, the effects are usually visible on the full range of plots.

11.2.3 "Focused" specter density plots

Some rules show specter density plots that varies greatly both in the w and s domain, two of which are plotted in Figure 12.

Rule 105 (Figure 12a) seems to hold a set of peaks of constant width whose positions slide along with increasing window size. Increasing CA size seems to



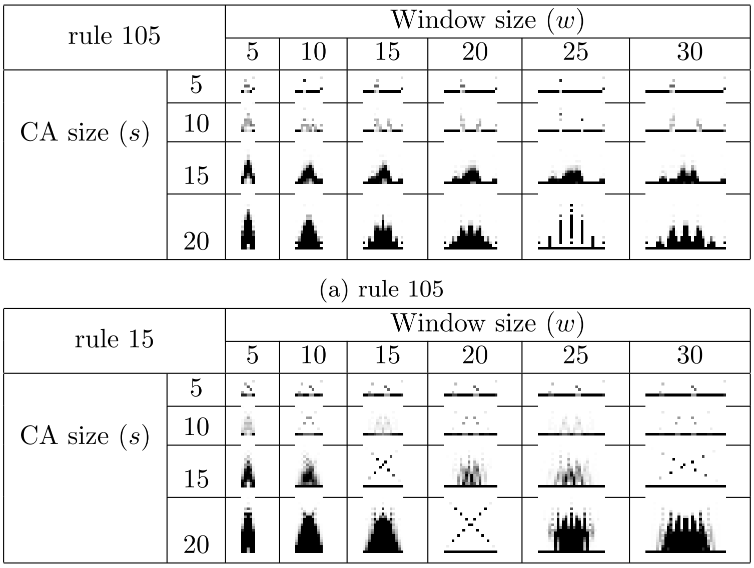
(a) rule 45

(b) rule 73

(c) rule 62

(d) rule 126

Figure 11: Specter density plots of the same ECAs as in the waterfall plots of Figure 9. As for waterfalls, density plots show variation between the four cases. (a) Resembling normal distribution. (b) Activity over all frequencies, with two peaks. (c) One tall peak whose base seems to be of near constant width, while the placement of the peak in the frequency domain is shifted due to scaling. (d) The plots of $s = 15$ seems to "lock" in a fixed pattern for $w \geq 20$, while the other plots show the usual scaling behaviour. Generated by software written for this thesis.



(a) rule 105

(b) rule 15

Figure 12: Specter density plots of two ECAs whose plots for some combinations (s, w) are more “focused”. (a) For $s = 20, w = 25$, the spectra have non-zero values only on seven out of 25 frequencies. (b) A tendency to form a dotted cross. Generated by software written for this thesis.

allow for more peaks. The most remarkable plot is the one for $s = 20, w = 25$. Here, the peaks are replaced by concentrated columns of activity, while intermediate frequencies are not present. Further, the columns leaves gaps in the low cell counts for the middle frequencies.

Rule 15 (Figure 12b) show a similar spread in specter density plots, with a tendency to form a dotted cross. Just as for rule 105, most plots show a set of overlapping peaks, while in this case several plots are severely limited with respect to spectral diversity. For the fully focused dotted crosses, each frequency is limited to hold one of two number of cells.

The issue of focused specter density plots will be revisited in Section 12.1.4.

11.2.4 Concluding remarks on specter density plots

While most rules show distinct shapes of specter density plots, many show large ranges of cell counts for some frequency ranges. The proposition that the shape of spectra are mostly dictated by the rule seems to hold true, but there is still room for spectral diversity within the borders intrinsic to the rule. The nature and extent of this spectral diversity is investigated in later sections.

12 Spectral diversity of ECAs

12.1 Varying window size

12.1.1 General results

In Figures 13, 14, 15 and 20, the spectral diversity of each of the 88 equivalence classes of ECAs are plotted over a range of window sizes $1 \leq w \leq 32$. The CA sizes were 8, 12, 16 and 24.

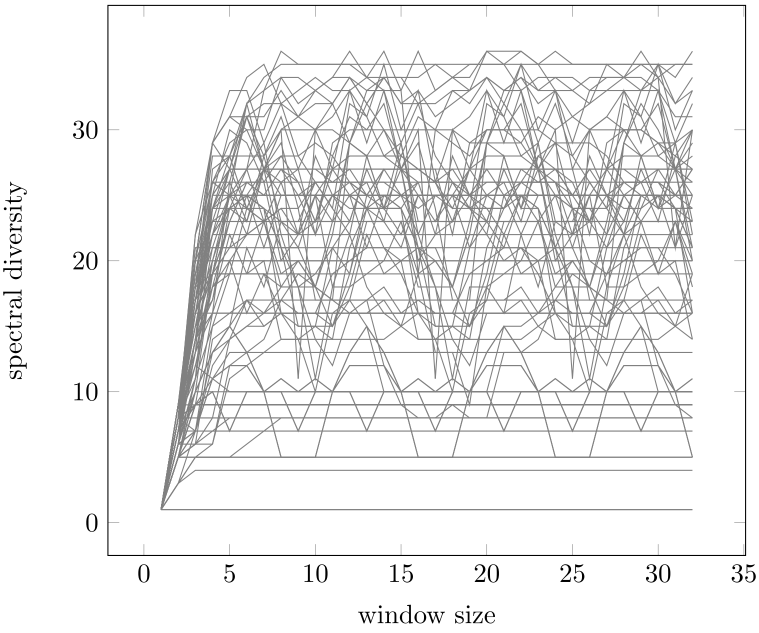


Figure 13: Spectral variation, CA size $s = 8$. All ECAs, superimposed. Periodicity is evident.

A striking feature of the data set is that while some rules seem to approach a limit spectral diversity, others show oscillating behaviour. Perhaps more striking, most oscillation seems to be synchronized with little to no phase shift and a period equal to the CA size s. Four cases of oscillation show a period of $2s$, two others a period of $\frac{1}{2}s$. For the purpose of further investigation, spectral diversity with respect to window size is classified using the following set of definitions:

- Class W1 rules have constant spectral diversity for window sizes w larger than the CA size s. For all $w > s$, all spectral diversities are equal.
- Class W2 rules are those rules that are not of class W1, and where for two window sizes w_a and w_b, with spectral diversities S_a and S_b, if $w_a < w_b$ then $S_a \leq 2S_b$. That is, the function range for a larger window size is never less than half of the function range for a smaller window size.
- Class W3 rules are those rules that are not of class W1 or W2, and whose behaviour with respect to window size is periodic. An optional subscript

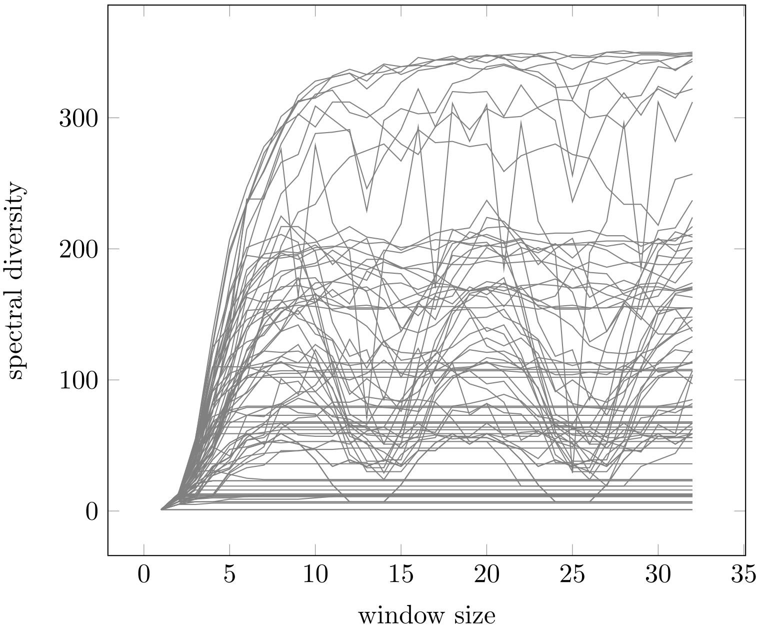


Figure 14: Spectral variation, CA size $s = 12$. All ECAs, superimposed.

denotes the period of the oscillation, which is denoted in terms of the CA size s.

- Class W4 rules are those rules that are not of any other class, should such rules exist.

This scheme is proposed in this thesis, for the purpose of classifying rules based on their togglecount transform. Note that this set of definitions may classify the same rule as belonging to different classes depending on the choice of s. E.g. for $s = 1$ all rules are of class W1, since the emergent behaviour of the one cell will inevitably be either fixed, toggling once or constantly toggling. Thus, spectral diversity is constant after a transient of at most one time step. Further discussion will assume $s = 16$ unless otherwise stated.

The naming scheme uses a "W" prefix (for "window space") to distinguish this class scheme from other CA classifications. In Figure 15 the ($s = 16$) plots are colour coded to highlight the classes. Figure 16 shows one example from each of W1, W2 and the three identified periods of oscillation in W3. The number of rules of each class belonging to each of the Wolfram classes and Ninagawa categories are provided in Table 2.

[§] In the article by Ninagawa [33] rule 26 is listed as being both category 1 and 2-A, while rule 36 is not listed at all. The assumption here is that the rule 26 listed as category 1 is in fact the missing rule 36, while the listing as category 2-A is correct. This fits with the descriptions of what behaviour to be found in each of the categories. Rule 36 is in W1, rule 26 in W2.

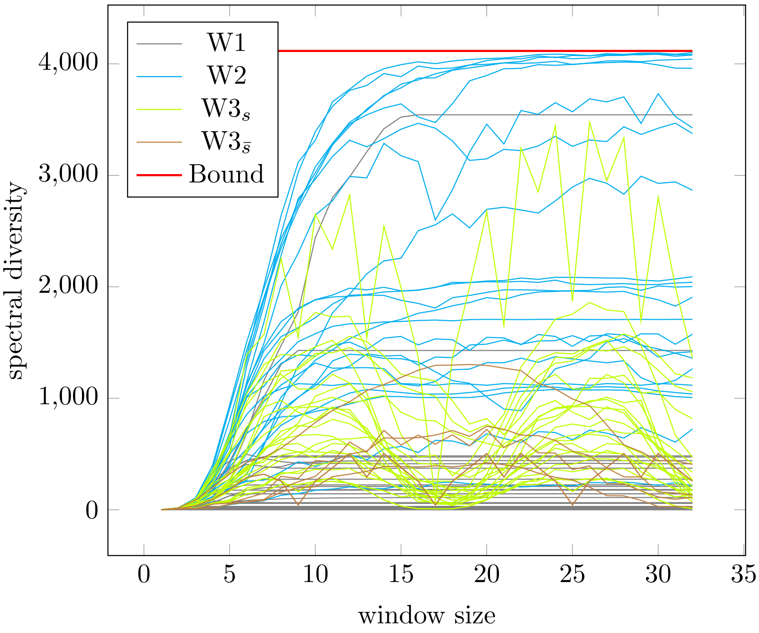


Figure 15: Spectral diversity, CA size $s = 16$. Colour coded for identified classes of behaviour with respect to window size. W1: Converges rapidly to a constant. W2: Converges but does not settle. W3$_s$: Periodic, $T = s$. W3$_{\bar{s}}$: Periodic, $T \neq s$.

	Wolfram class				Ninagawa category					
	I	II	III	IV	1	2-A	2-B	3	E	Total
W1	7	28	2	0	$34^{\S}$	0	2	0	1	37
W2	1	10	7	4	4	6	8	3	1	22
W3	0	27	2	0	0	27	2	0	0	29
Total	8	65	11	4	38	33	12	3	2	88

Table 2: The 88 ECA equivalence classes. Number of ECAs of each Wolfram class and of each Ninagawa category, grouped by window space classes W1, W2 and W3.

12.1.2 W1 rules

There are 37 rules in the W1 class for CA size 16. Spacetime plots of these rules for $s = 16$ and $w = 32$ are shown in Figure 17, and one example of the behaviour with respect to window size is shown in Figure 16. Out of the eight ECAs of Wolfram's class I, all rules except for rule 168 belong to W1. The remaining W1 rules are all from Wolfram's class II, with two exceptions: Rules 60 and 90, both of Wolfram's class III.

The fact that most class I rules also are in W1 seems reasonable. As the class I rules reaches a point attractor, once the longest transient is reached no changes in the togglecount transform are encountered. There are no additional toggles beyond this point. When rule 168 fails to reach constant spectral diversity at

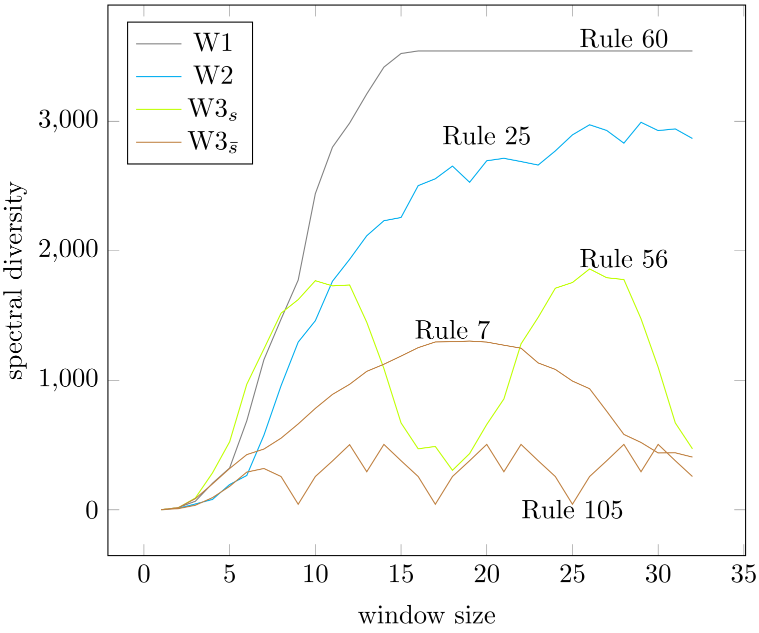


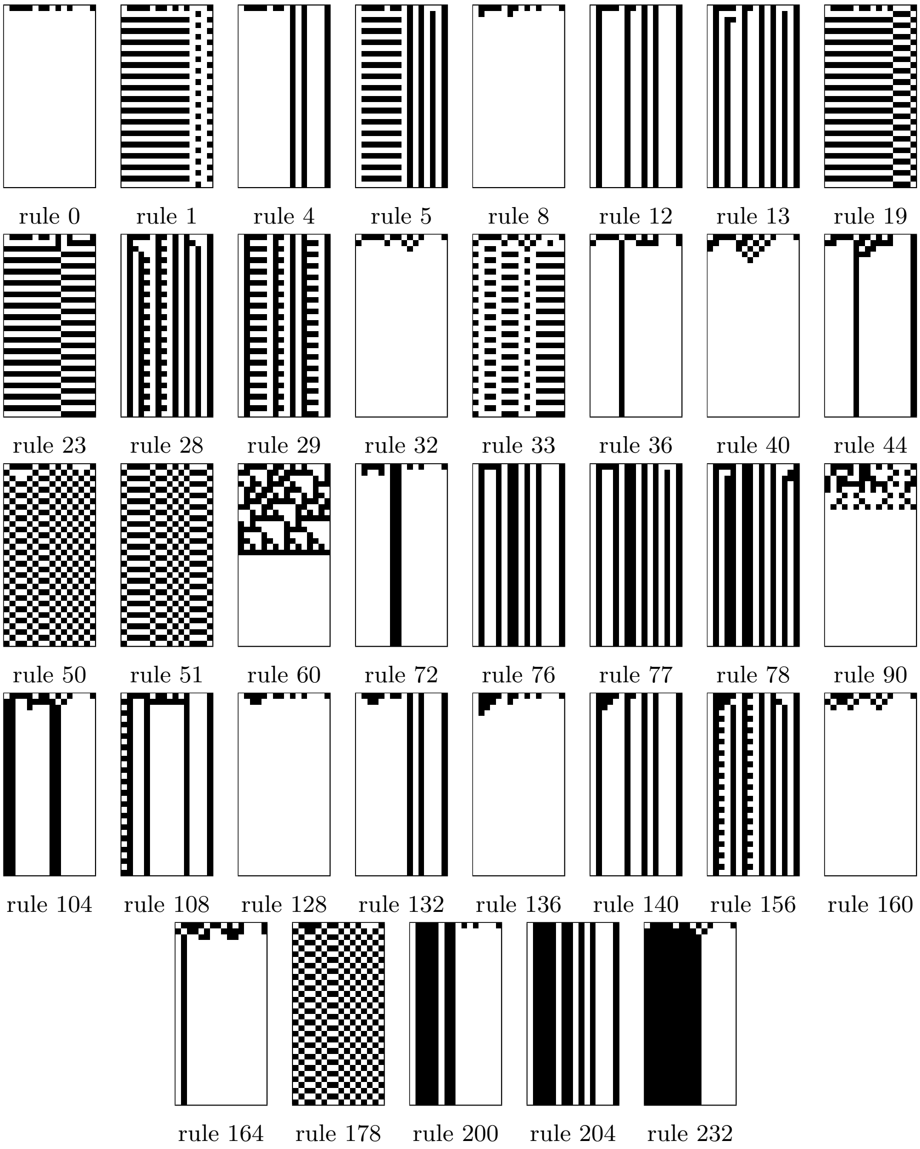
Figure 16: Spectral diversity, CA size $s = 16$. One rule from each of the identified behaviours for ECAs with respect to window size: Classes W1, W2, $W3_s$, $W3_{2s}$ and $W3_{0.5s}$. Note that rule 60 show a much higher spectral diversity than what is typical for W1 rules, although the general shape of the graph is typical.

$w = s$, it is because for some inputs it shows class II behaviour with (shifting) localised structures. This adds a component of oscillation, as will be shown for class W3 in section 12.1.4. The spectral diversity of rule 168 is varying over the range $[1038, 1104]$ for $16 \leq w \leq 32$.

Similarly, that the remaining W1 rules are mostly class II is explained by the definition of class II. Class II behaviour is repeating, localised structures. As these structures have a limited width, they are also limited with respect to the togglecount transform. For a structure of width a, the longest "local" cycle is of length a^2. This means the number of possible frequencies is also limited to a^2 for the cells of that structure. For 28 of the 65 class II rules, this results in the spectral diversity being fixed prior to $w = s$. They appear in Figure 17 as the CAs that still has black cells at the bottom of the plot. Here, they have either reached a point attractor, or a portion of the cells has entered a state of toggling for every time step while the remaining cells are fixed.

All class II rules identified as class W1 are also identified by Ninagawa as category 1. Only three class II rules (37, 94 and 172) are category 1 but not in W1. This shows a connection between W1 and category 1 behaviour, in filtering out class II rules whose temporal evolution leads to either a point attractor, or an attractor of length two.

Rules 60 and 90 are somewhat surprising to find in W1, as class III rules show random behaviour. One would think that the randomness should result in



rule 0 rule 1 rule 4 rule 5 rule 8 rule 12 rule 13 rule 19

rule 23 rule 28 rule 29 rule 32 rule 33 rule 36 rule 40 rule 44

rule 50 rule 51 rule 60 rule 72 rule 76 rule 77 rule 78 rule 90

rule 104 rule 108 rule 128 rule 132 rule 136 rule 140 rule 156 rule 160

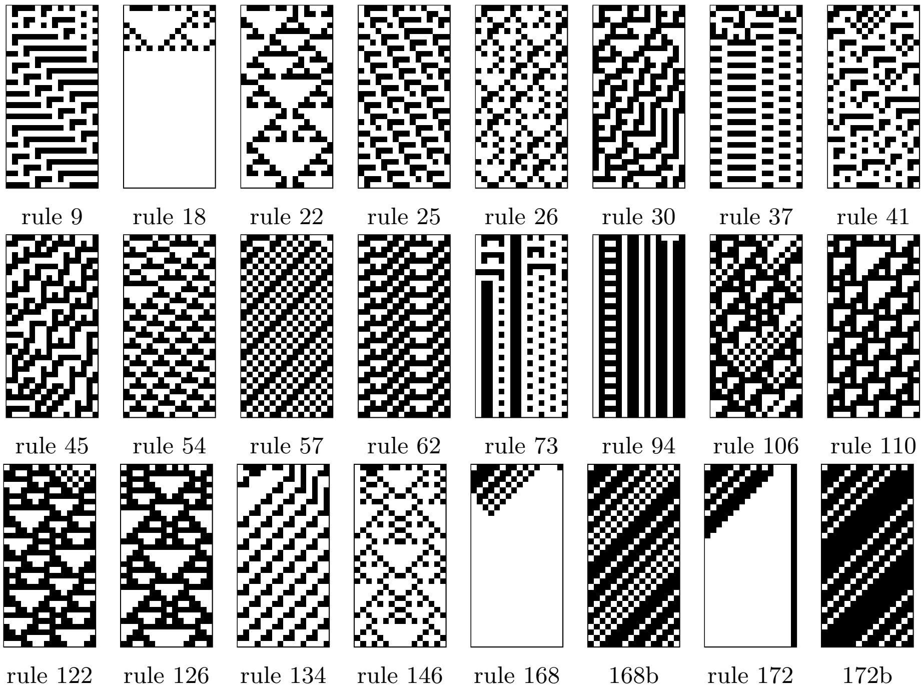
rule 164 rule 178 rule 200 rule 204 rule 232

Figure 17: Class W1 ECAs. Spacetime plots from the same random initial configuration. Generated using software written for this thesis.

varying spectral diversity over varying window size. However, both of these rules turn out to have short enough transients and small enough attractor lengths to fall within the W1 class. Indeed, for the random initial configuration used in Figure 17 they both reach the null configuration as a point attractor. Also, rules 60 and 90 by far give the largest spectral diversities for W1 rules, of 3543 and 1429 respectively. The other W1 rules are in the range $[1, 483]$. This difference is easily explained by the apparent randomness of class III rules.

12.1.3 W2 rules

22 ECA rules belong to the W2 class for $s = 16$. Spacetime plots are shown in Figure 18, and one example of the spectral diversity with respect to window size is shown in Figure 16.



rule 9 rule 18 rule 22 rule 25 rule 26 rule 30 rule 37 rule 41

rule 45 rule 54 rule 57 rule 62 rule 73 rule 94 rule 106 rule 110

rule 122 rule 126 rule 134 rule 146 rule 168 168b rule 172 172b

Figure 18: Class W2 ECAs. Spacetime plots from the same random initial configuration. 168b and 172b show emergent behaviour for an alternative initial configuration with no consecutive white cells. Generated using software written for this thesis.

The aforementioned class I rule 168 belong to this class, as explained in section 12.1.2. Rule 172, a Wolfram class II rule, show similar behaviour to that of rule 168: For some rules it shows class I behaviour, for others class II. Rule 18, while quickly entering the null point attractor in Figure 18, is a Wolfram class III rule whose typical behaviour is random-like and lasting. All four of the class IV rules are in W2. These are rules 41, 54, 106 and 110.

As for W1, the occurrence of class II rules can be explained by the localised structures leading to short (local) attractor lengths. The difference with respect to the W1 cases is that for class II rules in W2 transients are longer and show more variation in length, or the diversity of cycle lengths of repeating localised structures is larger.

The fact that most class III rules are in W2 may have to do with the relative uniformity of the randomness generated by those rules. As noted in section 11, class III rules tend to result in spectra resembling normal distributions regardless of window or CA size. Class III rules are also in Ninagawa category 2-B, which are characterised by a power spectrum resembling white noise. Again a link between togglecount and fourier transform is experienced through the window space classes and Ninagawa's categories.

As for the class IV rules, their long transients and long attractors certainly prevent them from being in W1. Rule 110, which is proven to hold the property of universal computation [9], is one of the five rules with the highest spectral diversities throughout the space of window sizes, along with class IV rules 41 and 106. W1 is also the only class to have category 3 rules, as all category 3 rules are to be found in class IV.

The plots of the rules with the largest spectral diversity show little deviation to a smooth increasing curve. It seems the potential for variation in these CAs is so high that it is pushing close to the theoretical limit imposed by the number of unique initial states or the number of possible spectra, thus never falling far beyond any level achieved by a smaller window size.

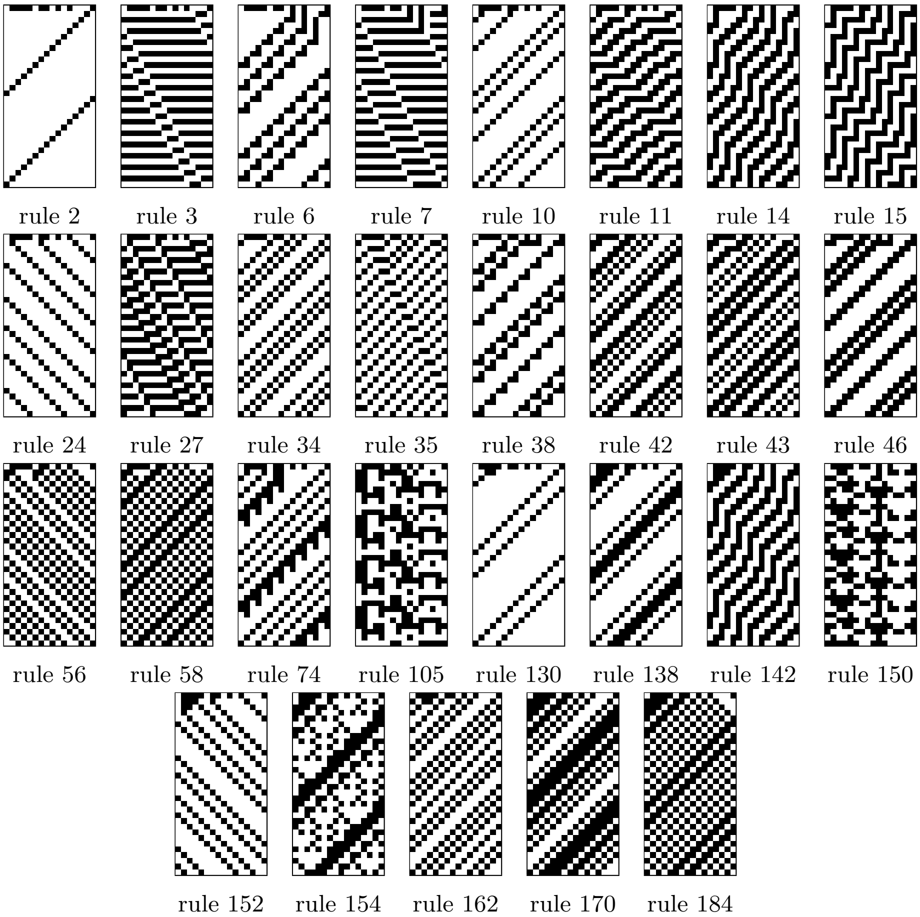
12.1.4 W3 rules

The 29 remaining rules all show oscillating behaviour for $s = 16$. 27 of them are of class II, and all of these rules yield repeating shifting patterns on spacetime plots from random initial conditions. The 23 rules showing a period of $T = s$ are the rules whose emergent patterns shifts by one cell for every time step. The four remaining class II rules show a period of $T = 2s$, and those four generate localised patterns shifting one cell for every two time steps.

Finally, without any apparent shift or symmetry in the typical emergent patterns, class III rules 105 and 150 show the exact same spectral diversity and a period of $T = 8 = \frac{1}{2}s$. Both of these rules are symmetrical, have $\lambda = \frac{1}{2}$ and are the logical opposites of each other. Their boolean forms are $p \oplus q \oplus r$ and $\neg(p \oplus q \oplus r)$ respectively. As seen for one random initial condition in Figure 19, their typical behaviour for $s = 16$ is to immediately enter a cyclic attractor of length 8. Rules 105 and 150 are among the ECA showing "focused" specter density plots (Section 11.2.3) for certain CA sizes and window sizes.

The reason for shifting patterns resulting in oscillating spectral diversity with respect to window size is that all cells are essentially toggling in the same pattern. Let the number of toggles during one full such cycle be T, and the period p. A transform over a window size of $p + 1$ will yield T as the toggle count for all cells. If this period is preceded by a transient of one time step, the specter from window sizes 2 and $p + 2$ will be the same, except the latter will be shifted T toggle counts in the frequency domain, as all cells have T additional toggles.

At the window sizes aligning with an integer number of loops through the cycle period p after the initial transient, the same spectral diversity as for the transient itself will be experienced. For intermediate values, toggle counts may differ between cells, and more spectral diversity is to be found. The farther away from the aligned window sizes, the more diversity may occur. For rules with short transients this results in a repeating oscillating sequence of spectral diversities with respect to window size.



rule 2 rule 3 rule 6 rule 7 rule 10 rule 11 rule 14 rule 15

rule 24 rule 27 rule 34 rule 35 rule 38 rule 42 rule 43 rule 46

rule 56 rule 58 rule 74 rule 105 rule 130 rule 138 rule 142 rule 150

rule 152 rule 154 rule 162 rule 170 rule 184

Figure 19: Class W3 ECAs. Spacetime plots from the same random initial configuration. Generated using software written for this thesis.

Rules with longer transients may experience some of the oscillating effect, but differences in transient lengths will lead to a differing time step for the oscillating behaviour to begin. As the oscillations are out of phase, their sum will not lead to overall oscillating behaviour. This is what happens for the rules with localised shifting patterns that are in W2.

The oscillating behaviour of 105 and 150 can be given a similar explanation as for the CAs with short transients and shifting patterns. As they immediately enter a short cyclic attractor of period p, any cell whose toggle count is T during that period will have a toggle count kT after k periods. This means the specter at all window sizes $kp + 1$ are mapped, with each specter for one k directly mapping to the specter of another k. The result is that the spectral diversity at any window size $kp + 1$ will be the same. As for the pattern shifting rules, larger spectral diversity will be found for other window sizes.

Most Ninagawa category 2-A rules are in W3. This is related to the shifting temporal evolution, which leads to a peak in the power spectrum for large windows for the fourier transform. For rule 6, whose pattern shifts one cell every time step, Ninagawa observed a peak at frequency $f = w/s$, for window size w and CA size s. [33]. This relates to the behaviour of the togglecount transform with respect to window size, with a period of w.

12.1.5 W4 rules

Of the 88 ECAs, no rules are of class W4. Decreasing spectral diversity for increasing window size seems to have a connection to oscillation, either due to shifting patterns or because of immediate short cycle lengths. When spectral diversity is introduced for a smaller window size, it seems unlikely to find an effect permanently undoing this diversity for all larger window sizes.

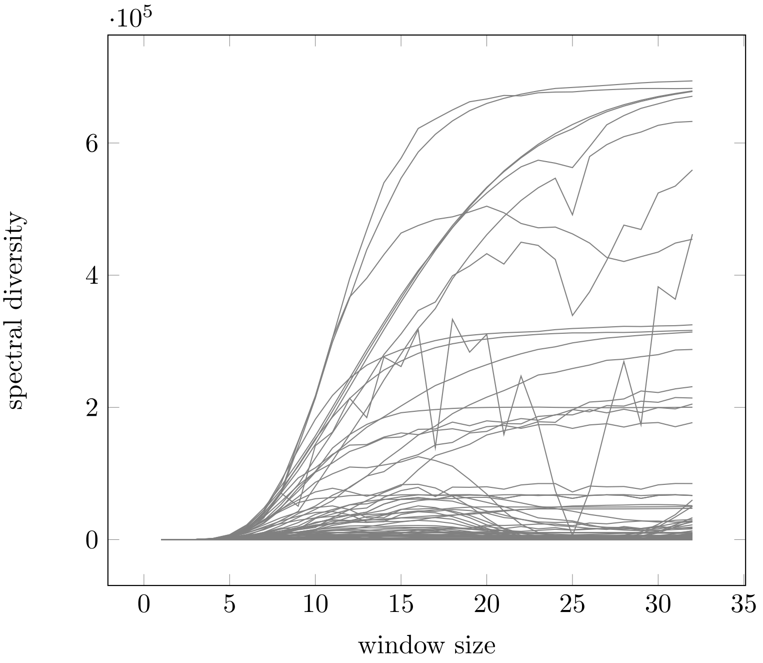


Figure 20: Spectral diversity, CA size $s = 24$. All ECAs, superimposed. The relative difference in spectral diversity is larger than for smaller CA sizes.

12.2 Varying CA size

12.2.1 General results

Spectral diversity for ECA rules with respect to CA size s are plotted in Figure 22 and Figure 21, colour coded for which Wolfram class they belong to. For all window sizes, Wolfram class III and IV rules show the highest spectral diversities, class II rules are distributed over most of the range of diversities and class I are mainly centered at the bottom. When $s \leq w$, for most combinations of s and window size w all class III and class IV rules have spectral diversities of more than 1/10 of the teoretical bound.

When investigating the λ parameter, a connection to spectral diversity is found, as is expected. For this investigation, the rule of the lowest λ value was chosen from each equivalence class, so that the λ value is on the interval $[0, 0.5]$. While some rules of all values of λ show little spectral diversity, only one rule with $\lambda = 2/8$, and some rules of $\lambda = 3/8$ and $\lambda = 4/8$ show a spectral diversity of more than 1/3 of the theoretical bound for $w = 16$ and $s = 24$. This shows a weak correlation between λ and spectral diversity.

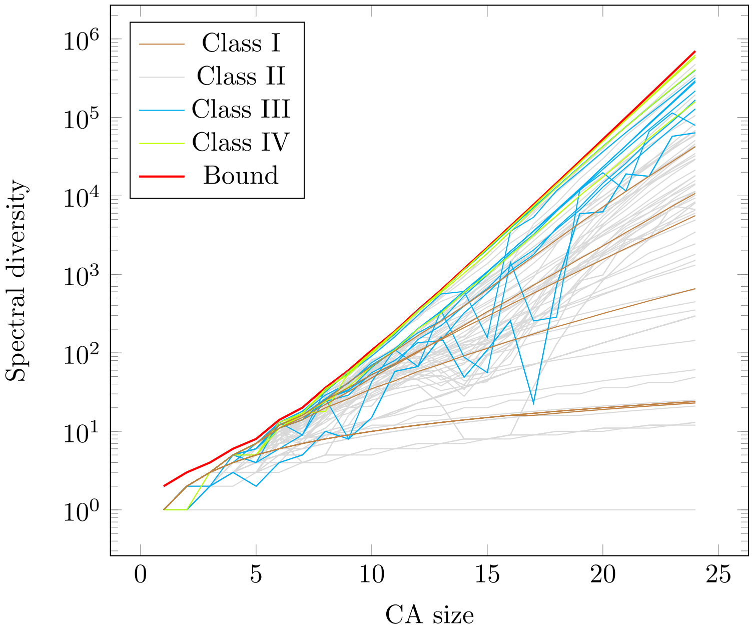
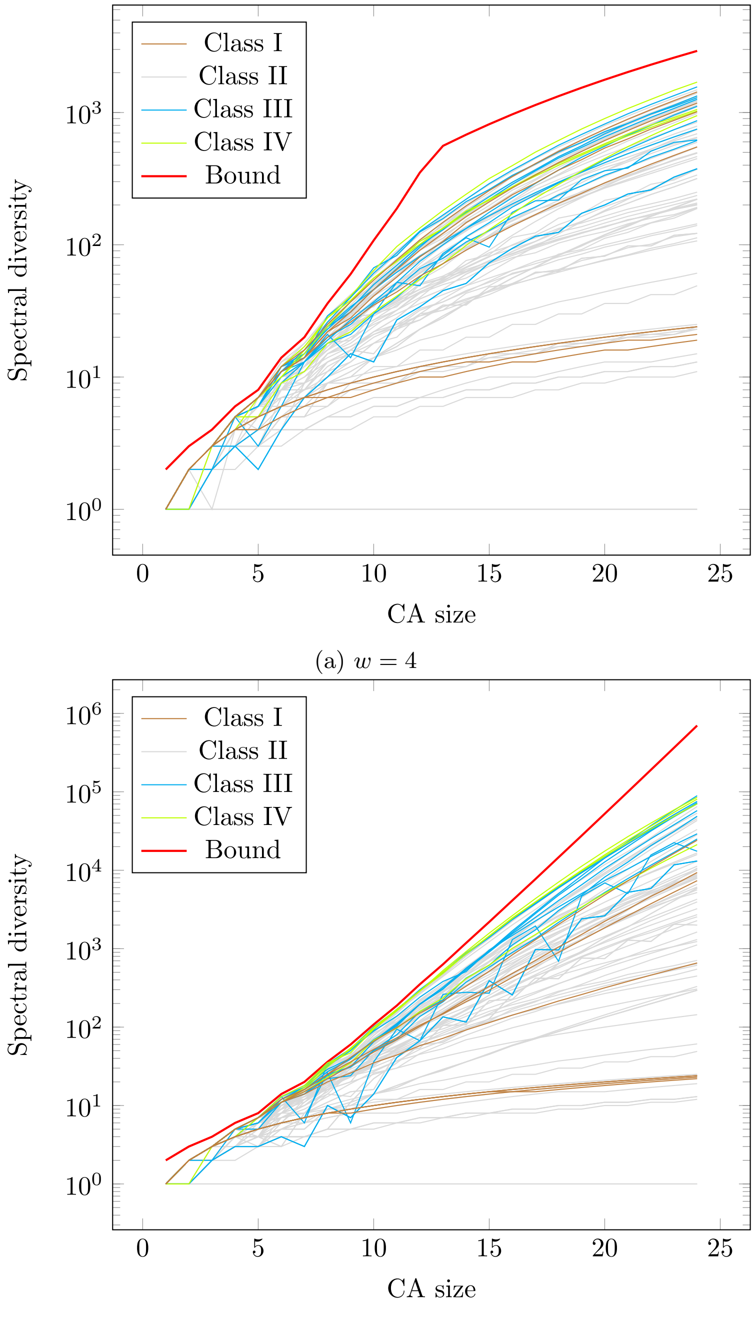


Figure 21: Spectral diversities along with the bounds given by the domain. Window size $w = 16$. Color coded according to Wolfram class. Most rules of class III and IV are above 1/10 of the bound for most CA sizes $1 \leq s \leq 24$. Most class I and II rules show a spectral diversity below 1/10 of the bound, and are distributed over the full range.

12.2.2 Small CA sizes

For CA sizes s less than almost two times the window size w, the rules with the highest spectral diversities are virtually at the theoretical limit. This is most noticeable in Figure 21, where the graphs of some class IV rules overlap the graph of the bound for most of the graphed interval. Also, the spread of spectral diversities for rules of all Wolfram classes, as well as that for all window space classes, is near uniform for the smallest CA sizes.

The spread of spectral diversities with respect to CA size is best illustrated by comparing the window space graphs of Figure 13 and Figure 20 from Section 12.1. In the former, $s = 8$, and spectral diversities of all types of rules are



(a) $w = 4$

(b) $w = 8$

Figure 22: Spectral diversities along with the bounds given by the domain and codomain. Window sizes $w = 4$ and $w = 8$. Color coded according to Wolfram class. (a) $w = 4$, domain bound for $1 \leq s \leq 12$, codomain bound for $s \geq 13$. (b) $windowsize = 8$, domain bound.

spread over the full range of theoretically possible values. In the latter, $s = 24$, and most rules have relatively small values of spectral diversity. Of the rules with a spectral diversity of more than 1/4 of the bound, few show oscillating behaviour and none reach a constant value for $s = 24$. In contrast, for $s = 8$ most rules of all classes show a diversity of more than 1/4 of the theoretical maximum.

12.2.3 Large CA sizes

As the CA size grows larger, spectral diversity decreases relative to the theoretical bound, for all rules. As seen in Figure 22a, the largest gap occurs at the point where the CA changes from being input bound to being output bound. The graph is for window size $w = 4$, and the CA is output bound from CA size $s = 13$. For even larger CA sizes, for the rules with highest spectral diversities the relative gap slowly decreases. The emergence of a growing gap is also seen for other window sizes, for example for $w = 8$ graphed in Figure 22b.

12.2.4 The rule intrinsic bound on spectral diversity

When the spectral diversity is close to the domain bound, it means almost every unique initial configuration gives a unique specter. For small CA sizes, where the number of initial configurations is by far the limiting factor, this is the case for many rules. It relates to the observation of stretching in the s domain of the specter density plots of Section 11.2.1: Even for small s, the set of initial configurations results in a full repertoire of differing specter shapes.

For larger CA sizes, even if the theoretical bound on the codomain size allows for each initial configuration to map to a unique specter, the rule specific specter limitations blocks part of the codomain from usage. A new bound is imposed by limitations intrinsic to the rule, encompassing only a fraction of the spectra. This lowers the codomain boundary to a fraction of its theoretical value, with the fraction being different for different rules.

Part V

Conclusion

13 Results

Togglecount, a transformation for the purpose of getting multiple variables as output from CAs, have been suggested in this thesis. It is a simple transform over the temporal evolution of the CA. Also introduced is the concept of spectral diversity, which is the number of unique spectra given by the togglecount transforms of evolutions from all initial configurations.

Some ECAs show more variation of spectra over temporal evolution than others. For ECAs whose specter heavily depends on the choice of timing for the transform window, this may lead to similar problems as when choosing the global state on a particular time step for output. A simple investigation of the specter evolution over time, with a sliding window of constant size for the transform, suggests most ECAs have little temporal fluctuation in their spectra.

All rules have intrinsic boundaries for the spectra from the togglecount transform over the temporal evolution of the CA. While some rules exhibit strict boundaries, others allow for more spectral diversity. The general shape of these boundaries scale with window size and CA size. Some rules show different boundaries for certain CA sizes, window sizes or combinations thereof.

Spectral diversity over increasing window size can be classified in three classes, which are proposed in this thesis:

- W1: Approaches a constant spectral diversity prior to $w = s$, for window size w and CA size s.
- W2: Approaches a limit spectral diversity, but fails to stabilise. Does not fall below 1/2 of highest spectral diversity of any smaller window size.
- W3: Spectral diversity is oscillating with respect to w. Related to short transient and shifting patterns in the attractor.

In addition, class W4 is reserved for rules that do not fit in any of the other classes although the suggestion is that no such rule exists. A rule may belong to different classes for different CA sizes. The classes are related to the five categories of Ninagawa, in that most category 1 rules are W1, most category 2-A rules are W3, and most categories 2-B and 3 rules are W2. As the basis for the Ninagawa categories is fourier transforms over the temporal evolution of the CA, this indicates a relation between the fourier transform and the togglecount transform.

While boundaries for resulting spectra are intrinsic to rules, most rules give room for spectral diversity. Wolfram class III and IV rules show spectral diversity of more than 1/10 of the theoretical maximum, for most combinations of CA size and window size. The theoretical boundaries are given by the number of unique initial configurations (domain) and the number of possible spectra (codomain). The latter seems to impose more of a limit than the former. Several rules show spectral diversity close to the boundary when the domain is the smaller, while no rules are close to the boundary when the codomain is the smaller. This is because of the boundaries for the spectra intrinsic to the rule itself.

Spectral diversity increases with CA size, and the increase is near linear with domain size for small CA sizes relative to window size. For large CA sizes relative to window size, spectral diversity is increasing near linearly with codomain size.

Spectral diversity increases with window size, but the increase stops around the point where the window size reaches the size of the CA. Some rules show lasting oscillating spectral diversity with respect to window size, most of which has a period equal to the CA size.

Togglecount allows for multiple variable output from a CA with the constraint of a sum of these variables equal to the size of the CA, a number of variables equal to the window size chosen for the transform, and boundaries imposed by the particular CA rule. For some ECA rules pectral diversity leads to varying output values for multiple variables, and the togglecount transform seems to be a feasible way to get multiple variables from a CA. The subject should be investigated further.

14 Further work

14.1 Changing the boundary condition

By changing the boundary condition, the number of distinct initial configurations will change. Using the null boundary condition will lift the equivalence on rotation, while introducing the null boundary on one side and the one state as boundary on the other side will remove equivalence on reflection for asymmetric rules. As the bound on initial configurations is increased, higher spectral diversity is to be expected for CA sizes that are now input bound.

14.2 Number of colours and neighbourhood range

Further investigation could be done by increasing the number of colours or the size of the neighbourhood. The program “codomaincounter” is capable of simulating CAs of more colours and larger ranges. As a preliminary step the spectral diversities of 100 random four-colour rules, 1000 random three-colour rules and 200 random range two binary rules have been found for various CA sizes and window sizes. This data is yet to be analysed. Further scaling of colour count and range requires a change in the way codomaincounter handles rule numbers, as the rule space exceeds what can be represented using 64 bit integers.

14.3 Dimensionality

A comparison of CAs of different dimensionality using different neighbourhoods could investigate how these parameters relate. A 2D von Neumann neighbourhood of range 1 and a one dimensional neighbourhood of range 2 both give a neighbourhood of 5 cells, but the interconnectedness will differ. It could be interesting to find out how they relate with respect to spectral diversity of the togglecount transform.

14.4 Larger CAs and windows

While ECAs has been investigated for CA sizes up to 24 and window sizes up to 32, interesting phenomena may appear on larger CAs or larger window sizes. Based on knowledge from this thesis, research on larger CA sizes and window sizes can possibly be more focused, allowing for some shortcuts. For instance, W1 and W3 rules can be detected and avoided when running tests on large window sizes.

Specter density plots of large CA sizes and window sizes could prove useful in determining the shape of the intrinsic boundaries on spectra for a given rule. Scaling to large CA sizes and window sizes could lead to new insights into the dynamics of rules showing complex boundaries, but would require either better simulators, more powerful equipment or the introduction of statistics (i.e. running simulations only for a limited number of initial configurations).

14.5 Do window space class W4 rules exist?

Four classes of variation in spectral diversity over increasing window size were introduced in this thesis, in Section 12.1.1. While no ECA seems to be in class W4, it is not known whether such CAs are possible, either for certain CA sizes or for other rule spaces. The only observed effects leading to decreasing spectral diversity for increasing window size are oscillations due to either shifting or short immediate cyclic attractors. Other effects may still exist, that settles on low spectral diversity for large window sizes but show large spectral diversity for intermediate window sizes.

14.6 Walsh-Hadamard transform

Proposed by Wolfram [46] is to use the discrete Walsh-Hadamard transform [37] for statistical analysis of the emergent configuration of CAs. The fast Walsh-Hadamard transform is a space efficient version of the transform, which in contrast to the fast fourier transform uses only addition and subtraction of real numbers. [14] This makes the fast Walsh-Hadamard transform a candidate for similar studies as has been done on togglecount in this thesis.

14.7 The distribution of spectral diversities

The distribution of spectral diversities can also be investigated. As has been shown in Section 12.2.2, for small CA sizes rules are evenly distributed over the full range of spectral diversities, when for large CA sizes most rules have little spectral diversity and few rules are close to the bounds. How this distribution changes with respect to CA size, window size, window space class and other parameters could be an interesting topic for further research. “Density estimation” is a large field of statistics, for the purpose of converting a set of values to a “probability density function” describing the distribution. The method of “variable kernel density estimation” could be a starting point for plotting distributions of spectral diversity.

References

[1] J. Albert and K. Culik II. A simple universal cellular automaton and its one-way and totalistic version. *Complex Systems*, 1:1–16, 1987.

[2] D. Ashlock and S. McNicholas. Fitness landscapes of evolved apoptotic cellular automata. *Evolutionary Computation, IEEE Transactions on*, 17(2):198–212, 2013.

[3] R. Balzer. An 8-state minimal time solution to the firing squad synchronization problem. *Information and Control*, 10(1):22–42, 1967.

[4] E.R. Berlekamp, J.H. Conway, and R.K. Guy. Winning ways for your mathematicautomata, volume 2. *Academic Press*, 1982.

[5] M. Bidlo and Z. Vasicek. Evolution of cellular automata using instruction-based approach. pages 1–8. IEEE, 2012.

[6] A.W. Burks. *On Cellular Automata.* University of Illinois Press, 1970.

[7] K. Cattell, F. Ruskey, J. Sawada, M. Serra, and C. R. Miers. Fast algorithms to generate necklaces, unlabeled necklaces, and irreducible polynomials over gf (2). *Journal of Algorithms*, 37(2):267–282, 2000.

[8] E.F. Codd. *Cellular automata.* Academic Press, 1968.

[9] M. Cook. Universality in elementary cellular automata. *Complex Systems*, 15(1):1–40, 2004.

[10] E. Fredkin and T. Toffoli. Conservative logic. *International Journal of Theoretical Physics*, 21(3/4), 1982.

[11] H. Fredricksen and I. J. Kessler. An algorithm for generating necklaces of beads in two colors. *Discrete mathematics*, 61(2):181–188, 1986.

[12] E. Goto. A minimal time solution of the firing squad problem. *Dittoed course notes for Applied Mathematics*, 208:52–59, 1962.

[13] H. A. Gutowitz. A hierarchical classification of cellular automata. *Physica D: Nonlinear Phenomena*, 45(1-3):136–156, 1990.

[14] A.A.C.A. Jayathilake, A.A.I. Perera, and M.A.P. Chamikara. Discrete walsh-hadamard transform in signal processing. *International Journal of Research in Information Technology*, 1(1):48–57, 2013.

[15] Y. Kayama, M. Tabuse, H. Nishimura, and T. Horiguchi. Characteristic parameters and classification of one-dimensional cellular automata. *Chaos, Solitions & Fractals*, 3(6):651–665, 1993.

[16] M.S. Keshner. 1/f noise. *Proceedings of the IEEE*, (70):212–218, 1982.

[17] G. A. Kohring. Towards the classification of all boolean cellular automata. *Physica A: Statistical Mechanics and its Applications*, 182(3):320–324, 1992.

[18] C. G. Langton. Computation at the edge of chaos: phase transition and emergent computation. *Physica D: Nonlinear Phenomena*, 42(1):12–37, 1990.

[19] W. Li. Power spectra of regular languages and cellular automata. *Complex Systems*, 1(1):107–130, 1987.

[20] W. Li and N. Packard. The structure of the elementary cellular automata rule space. *Complex Systems*, 4(3):281–297, 1990.

[21] K. Lindgren and M.G. Nordahl. Universal computation in simple one-dimensional cellular automata. *Complex Systems*, 4(3):299–318, 1990.

[22] B. Martin. Inherent generation of fractals by cellular automata. *Complex Systems*, 8(5):347–366, 1994.

[23] G.J. Martinez. A note on elementary cellular automata classification. *Journal of Cellular Automata*, 8(3–4):233–259, 2013.

[24] J. Mazoyer. A six-state minimal time solution to the firing squad synchronization problem. *Theoretical Computer Science*, 50(2):183–238, 1987.

[25] J. Mazoyer. An overview of the firing squad synchronization problem. In *Automata Networks*. Springer, 1988.

[26] M.L. Minsky. *Computation: finite and infinite machines*. Prentice-Hall, Inc., 1967.

[27] M. Mitchell, J. P. Crutchfield, and P. T. Hraber. Evolving cellular automata to perform computations: Mechanisms and impediments. *Physica D: Nonlinear Phenomena*, 75(11):361–391, 1994.

[28] M. Mitchell, P.T. Hraber, and J.P. Crutchfield. Revisiting the edge of chaos: Evolving cellular automata to perform computations. *Complex*, pages 89–130, 1993.

[29] E.F. Moore. Machine models of self-reproduction. In *Proceedings of a Symposium of the Applied Mathematical Society, Providence, RI*, volume 14, pages 17–33, 1962.

[30] J. Nagler and J. C. Claussen. 1/ f α spectra in elementary cellular automata and fractal signals. *Physical Review E*, 71, 2005.

[31] S. Nichele and G. Tufte. Evolution of incremental complex behavior on cellular machines. In *Advances in Artificial Life, ECAL*, volume 12, pages 63–70, 2013.

[32] S. Nichele and G. Tufte. Evolutionary growth of genomes for the development and replication of multicellular organisms with indirect encoding. In *Evolvable Systems (ICES), 2014 IEEE International Conference on*, pages 141–148. IEEE, 2014.

[33] S. Ninagawa. Power spectral analysis of elementary cellular automata. *Complex Systems*, 17(4):399, 2008.

[34] M. Olivier, A.M. Odlyzko, and S. Wolfram. Algebraic properties of cellular automata. *Communications in mathematical physics*, 93(2):219–258, 1984.

[35] N.H. Packard. Two-dimensional cellular automata. *Journal of Statistical Physics*, 38(5-6):901–946, 1985.

[36] N.H. Packard. *Adaptation toward the edge of chaos.* University of Illinois at Urbana-Champaign, Center for Complex Systems Research, 1988.

[37] K.R. Rao and N. Ahmed. Orthogonal transforms for digital signal processing. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'76.*, volume 1, pages 136–140. IEEE, 1976.

[38] K.H. Rosen. *Discrete Mathematics and Its Application.* McGraw-Hill, 6 edition, 2007.

[39] E. L. P. Ruivo and P. P. B. de Oliveira. A spectral portrait of the elementary cellular automata rule space. In *Irreducibility and Computational Equivalence*, chapter 16, pages 211–235. Springer, 2013.

[40] A.R. Smith III. Simple computation-universal cellular spaces. *Journal of the ACM*, 18(3):339–353, 1971.

[41] J. Von Neumann and A.W. Burks. Theory of self-reproducing automata. *IEEE Transactions on Neural Networks*, 5(1):3–14, 1966.

[42] A. Waksman. An optimum solution to the firing squad synchronization problem. *Information and Control*, 9(1):66–78, 1966.

[43] R.E. Walpole, R.H. Myers, S.L. Myers, and K. Ye. *Probability and statistics for engineers and scientists.* Pearson Prentice Hall, 8 edition, 2007.

[44] S. Wolfram. Statistical mechanics of cellular automata. *Reviews of modern physics*, 55(3):601, 1983.

[45] S. Wolfram. Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1):1–35, 1984.

[46] S. Wolfram. Twenty problems in the theory of cellular automata. *Physica Scripta*, 1985(T9):170, 1985.

[47] S. Wolfram. *A New Kind of Science.* Wolfram Media, 2002.

[48] A. Wuenche. Classifying cellular automata automatically. Santa Fe Institute Santa Fe, NM, 1998.

[49] J.B. Yunès. Seven-state solutions to the firing squad synchronization problem. *Theoretical Computer Science*, 127(2):313–332, 1994.