Global properties of cellular automata

Edward Jack Powley

Submitted for the qualification of PhD

University of York

Department of Computer Science

October 2009

Abstract

A *cellular automaton* (CA) is a discrete dynamical system, composed of a large number of simple, identical, uniformly interconnected components. CAs were introduced by John von Neumann in the 1950s, and have since been studied extensively both as models of real-world systems and in their own right as abstract mathematical and computational systems.

CAs can exhibit emergent behaviour of varying types, including universal computation. As is often the case with emergent behaviour, predicting the behaviour from the specification of the system is a nontrivial task. This thesis explores some properties of CAs, and studies the correlations between these properties and the qualitative behaviour of the CA.

The properties studied in this thesis are properties of the global state space of the CA as a dynamical system. These include degree of symmetry, numbers of preimages (convergence of trajectories), and distances between successive states on trajectories. While we do not obtain a complete classification of CAs according to their qualitative behaviour, we argue that these types of global properties are a better indicator than other, more local, properties.

Contents

Author's declaration

The material in Sections 7.1 to 7.4 was presented at the *Automata 2007* conference, and in *Journal of Cellular Automata* [PS09a].

The material in Chapter 8 was presented at the *Automata 2008* conference [PS08], and in *Journal of Cellular Automata* [PS09b].

The material in Chapter 9 and Appendix F is to appear in *Journal of Cellular Automata* [PS09c].

The material in Sections 10.1 and 10.2 has been submitted to *Complex Systems* [PS09d].

CHAPTER 1

Introduction

A *cellular automaton (CA)* is a system composed of a large number of simple components (*cells*) arranged on a lattice, with each cell connected to several other spatially local cells in a uniform way. Each cell has a *state*, and the states of the cells are updated synchronously on discrete time steps, with each cell's new state computed as a function of its current state and the current states of the cells to which it is connected. A *configuration* of the CA is an assignment of states to cells. We choose an initial configuration, and observe the sequence of configurations that follow on successive time steps.

CAs were introduced in the 1950s by John von Neumann, on the suggestion of Stanislaw Ulam, in von Neumann's work on self-reproducing machines [vNB66, Bur70]. CAs have since been used to model a wide variety of physical and biological systems [FHP86, TM87, Lam98]. They are also a useful formalism for massively parallel computation, mapping particularly well to hardware architectures such as field programmable gate arrays [STCS02] and programmable graphics processing units [GDH07].

CAs are examples of systems with *emergent* properties: the global behaviour of a CA is not designed into its components, but arises from the complex interactions between these components. Systems with emergent properties are ubiquitous in physics, biology, and the social sciences [Hol98, Joh01].

Wolfram [Wol84] asserts that the long-term qualitative behaviour of a CA falls into one of four classes: homogeneous, periodic, chaotic, or complex. These classes are described in more detail in Section 2.5; for now, the important point is that CAs as a class of systems exhibit a wide range of behaviour, although a particular CA tends to exhibit only one class for the vast majority of initial configurations. Furthermore, it is surprisingly difficult to formalise this classification.

This thesis explores several properties of CAs, and investigates how these properties relate to the CA's class of behaviour. The properties investigated are *global*, in that they relate to the entire lattice, and often the entire configuration space (as opposed to *local* properties, which relate to individual cells). The CA's Wolfram class is clearly a global property, so it seems reasonable that global properties of the CA would be a better indicator of

qualitative behaviour than local properties. The aim is not to derive a complete formal classification of CAs, and indeed this thesis does not present such a classification; rather, the aim is to investigate what conditions on the CA might be necessary and/or sufficient for certain classes of behaviour to occur.

The configuration space of a CA grows exponentially with its number of cells. A recurring theme in this thesis is the problem of how to study global properties of CAs when the numbers involved become so large so quickly. Simply throwing more computational resources at the problem is not an effective solution: it is the nature of exponential problems that orders-of-magnitude increases in resources yield only incremental increases in the size of problem that can be tackled. However, in some cases mathematical "tricks" can be used to reduce certain instances of the problem to non-exponential problems.

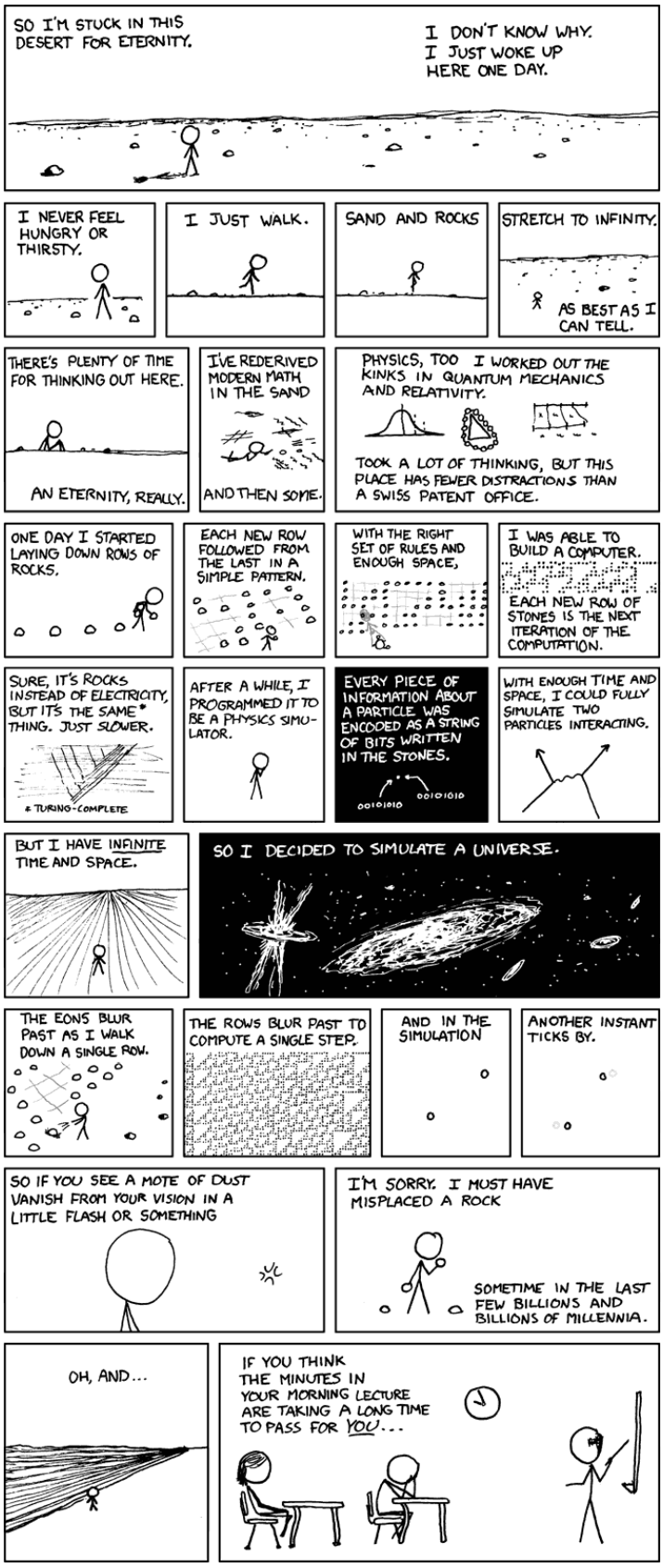
The structure of this thesis is as follows. Part 1 is primarily concerned with reviewing the existing literature, although some new results do appear (and are indicated as such in the text). The majority of the original work appears in Part 2.

In Part 1, Chapter 2 defines cellular automata, and discusses some important elementary concepts. Chapter 3 introduces *linear CAs*, which, like linear systems in general, can be studied in different ways to their nonlinear counterparts. Chapter 4 discusses the problem of running a CA "in reverse": that is, given the configuration at time t, determining what configurations may have applied at time $t - 1$. Chapter 5 reviews some miscellaneous properties of CAs, including parameters on the cell's update rule, entropy measures, and ideas from the theory of dynamical systems.

In Part 2, Chapter 6 defines *transition graphs* of CAs, and gives some algorithms for working with them. A transition graph is a way of visualising the entire configuration space of a CA, including the transitions between configurations. Chapter 7 presents results regarding numbers of symmetries for transition graphs for CAs in general, and Chapter 8 extends these results for linear CAs in particular. Chapter 9 revisits the notion from Chapter 4 of running a CA "in reverse", but here for the special case of homogeneous configurations (configurations in which every cell has the same state). Chapter 10 investigates the Hamming distances, and distances with respect to other metrics, covered by transitions between configurations. Finally, Chapter 11 gives some concluding remarks.

We assume the reader is familiar with elementary concepts from mathematics and theoretical computer science, such as set theory, functions, modular arithmetic, graph theory, formal languages, Turing machines, and computational complexity. More advanced mathematical prerequisites are given in Appendix A, and referenced from the main text when they are first used.

Figure on next page: "A Bunch of Rocks", from `http://xkcd.com/505/`.


SO I'M STUCK IN THIS DESERT FOR ETERNITY.
I DON'T KNOW WHY. I JUST WOKE UP HERE ONE DAY.
I NEVER FEEL HUNGRY OR THIRSTY.
I JUST WALK.
SAND AND ROCKS
STRETCH TO INFINITY.
AS BEST AS I CAN TELL.
THERE'S PLENTY OF TIME FOR THINKING OUT HERE.
AN ETERNITY, REALLY.
I'VE REDERIVED MODERN MATH IN THE SAND
AND THEN SOME.
PHYSICS, TOO I WORKED OUT THE KINKS IN QUANTUM MECHANICS AND RELATIVITY.
TOOK A LOT OF THINKING, BUT THIS PLACE HAS FEWER DISTRACTIONS THAN A SWISS PATENT OFFICE.
ONE DAY I STARTED LAYING DOWN ROWS OF ROCKS.
EACH NEW ROW FOLLOWED FROM THE LAST IN A SIMPLE PATTERN.
WITH THE RIGHT SET OF RULES AND ENOUGH SPACE,
I WAS ABLE TO BUILD A COMPUTER.
EACH NEW ROW OF STONES IS THE NEXT ITERATION OF THE COMPUTATION.
SURE, IT'S ROCKS INSTEAD OF ELECTRICITY, BUT IT'S THE SAME* THING. JUST SLOWER.
* TURING-COMPLETE
AFTER A WHILE, I PROGRAMMED IT TO BE A PHYSICS SIMULATOR.
EVERY PIECE OF INFORMATION ABOUT A PARTICLE WAS ENCODED AS A STRING OF BITS WRITTEN IN THE STONES.
00101010
00101010
WITH ENOUGH TIME AND SPACE, I COULD FULLY SIMULATE TWO PARTICLES INTERACTING.
BUT I HAVE INFINITE TIME AND SPACE.
SO I DECIDED TO SIMULATE A UNIVERSE.
THE EONS BLUR PAST AS I WALK DOWN A SINGLE ROW.
THE ROWS BLUR PAST TO COMPUTE A SINGLE STEP.
AND IN THE SIMULATION
ANOTHER INSTANT TICKS BY.
SO IF YOU SEE A MOTE OF DUST VANISH FROM YOUR VISION IN A LITTLE FLASH OR SOMETHING
I'M SORRY. I MUST HAVE MISPLACED A ROCK
SOMETIME IN THE LAST FEW BILLIONS AND BILLIONS OF MILLENNIA.
OH, AND...
IF YOU THINK THE MINUTES IN YOUR MORNING LECTURE ARE TAKING A LONG TIME TO PASS FOR YOU...

Part 1

Literature review

CHAPTER 2

Cellular automata

The structure of this chapter is as follows. Section 2.1 gives a formal definition of CAs and some related concepts. Section 2.2 describes *Conway's Game of Life*, a particularly well-known example of a CA. Sections 2.3 and 2.4 introduce an important class of CAs, the *elementary CAs*, which are studied extensively in the remainder of this thesis. Section 2.5 describes how CAs can be (informally) classified according to their qualitative behaviour. Section 2.6 discusses the maximum speed at which information can propagate in a CA, and Section 2.7 describes the structures by which information propagation takes place. Finally, Section 2.8 gives some examples of Turing complete CAs.

2.1. Definition and dynamics

A *cellular automaton (CA)* is a tuple $\langle S, (\mathbb{L}, +), T, f \rangle$ consisting of four components: a set S of *states*, a *lattice* $(\mathbb{L}, +)$, a *neighbourhood template* T, and a *local rule* f.

(1) The set of *states*, denoted S, can be any finite set with at least two elements. Often the state set is $S = \mathbb{Z}_s = \{0, \ldots, s-1\}$, the set of integers modulo s.

(2) The *lattice* $(\mathbb{L}, +)$ is an abelian group. Abelian groups are formally defined in Definitions A.7 and A.9; for current purposes, an abelian group is a set $(\mathbb{L})$ on which is defined a binary operation ($+$, or "addition") which satisfies the kinds of properties we expect of addition (including the existence of a "zero" element). The elements of the lattice are the *cells* of the CA. The following two families of lattice are the most common:
   (a) the infinite D-dimensional lattice $\mathbb{Z}^D$ with the operation of D-dimensional integer addition;
   (b) the finite D-dimensional lattice with periodic boundary condition, $\mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_D}$, with the operation of D-dimensional modular addition.

   When referring to a group, it is common to omit the binary operation if it can be inferred from context. We henceforth denote the lattice by $\mathbb{L}$ instead of $(\mathbb{L}, +)$.

(3) The *neighbourhood template* $T = \langle \eta_1, \ldots, \eta_m \rangle$ is a sequence over $\mathbb{L}$.
(4) The *local rule* is a function $f : S^{|T|} \to S$, mapping neighbourhood states to cell states. It is sometimes useful to think in terms of the CA's *rule table*, which is simply a table giving the value of f for each possible neighbourhood state.

A *configuration* of the CA is a function $c : \mathbb{L} \to S$. In other words, the configuration assigns a state to each cell. The set of all configurations is denoted $S^{\mathbb{L}}$.

The state of cell i in configuration c is denoted $c[i]$. The square brackets here are a matter of style: $c[i]$ has exactly the same meaning as $c(i)$, but the former notation is clearer in the frequent case when it is nested within other function applications. Also, it is common to think of configurations of 1-dimensional CAs as strings over the state set (see Section 2.3), in which case the $c[i]$ notation is intentionally reminiscent of the syntax for array indices in many programming languages.

Let $i \in \mathbb{L}$ be a cell. The *neighbourhood* of i is the sequence of cells $\langle i + \eta_1, \ldots, i + \eta_m \rangle$ obtained by adding i to each element of the neighbourhood template (recall that $+$ is the binary operation of the abelian group $\mathbb{L}$, whatever that operation may be).

The dynamics of the CA proceeds in discrete time steps $t = 0, 1, 2, \ldots$, with the "current" configuration being updated on each time step. Let c_t denote the configuration at time t. The state of cell i at time $t+1$ is obtained by applying the local rule to the states of i's neighbourhood at time t:

$$c_{t+1}[i] = f(c_t[i + \eta_1], \ldots, c_t[i + \eta_m]) \,. \tag{2.1}$$

This process yields a function from configurations to configurations, denoted $F : S^{\mathbb{L}} \to S^{\mathbb{L}}$, called the *global map* of the CA.

CAs are examples of *discrete dynamical systems*. Indeed, CAs are discrete in three different senses: they operate in discrete time steps, on a discrete space (the lattice), with a discrete state at each point in space (cell).

A configuration of the CA is a *state* of the dynamical system. (Note the clash of notation: a state of the CA as defined above is *not* the same as the state in dynamical systems terms. In this thesis, the word "state" generally has the former meaning.) A sequence of configurations visited by the CA is a *trajectory*, which may reach a *fixed point* (a repeating configuration) or an *attractor cycle* (a repeating sequence of configurations). The portion of the trajectory preceding the fixed point or attractor cycle is the *transient*.

Some other concepts from the theory of dynamical systems, such as sensitive dependence on initial conditions (Section 5.5) and basins of attraction (Chapter 6), can usefully be applied to CAs.

2.2. Example: Conway's Game of Life

Conway's Game of Life was first described by Martin Gardner [Gar70], having been suggested by John Conway in a letter to Gardner, and was later described by Conway himself [BCG82]. Partly because both of these early accounts are in the context of recreational mathematics and written for a general audience, and partly because of the fascinating and aesthetically pleasing patterns it produces when its evolution is animated on a computer screen, Life is by far the most famous example of a cellular automaton.

Life is a CA with state set $S = \{0, 1\}$, lattice $\mathbb{L} = \mathbb{Z}^2$ (in the infinite case) or $\mathbb{L} = \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2}$ (in the finite case), and neighbourhood template

$$T = \left\langle \begin{matrix} (-1,-1), & (0,-1), & (1,-1), \\ (-1,0), & (0,0), & (1,0), \\ (-1,1), & (0,1), & (1,1) \end{matrix} \right\rangle . \tag{2.2}$$

The neighbourhood of a cell consists of that cell and the eight cells immediately adjacent on the horizontal, vertical and diagonals.

The local rule $f : S^9 \to S$ is defined by

$$f \begin{pmatrix} x_{-1,-1}, & x_{0,-1}, & x_{1,-1}, \\ x_{-1,0}, & x_{0,0}, & x_{1,0}, \\ x_{-1,1}, & x_{0,1}, & x_{1,1} \end{pmatrix} = \begin{cases} 1 & \text{if } x_{0,0} = 1 \text{ and } \sigma \in \{2, 3\}\,; \\ 1 & \text{if } x_{0,0} = 0 \text{ and } \sigma = 3; \\ 0 & \text{otherwise,} \end{cases} \tag{2.3}$$

where

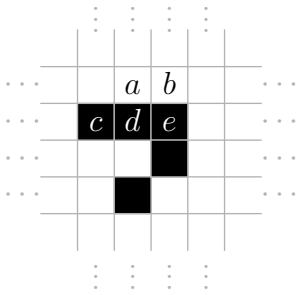
$$\sigma = \sum \begin{Bmatrix} x_{-1,-1}, & x_{0,-1}, & x_{1,-1}, \\ x_{-1,0}, & & x_{1,0}, \\ x_{-1,1}, & x_{0,1}, & x_{1,1} \end{Bmatrix} \tag{2.4}$$

is the sum of the states of the cell's neighbourhood, excluding the cell itself.

The rule can be described in words as follows:

- If a cell is in state 1 and has two or three neighbours in state 1, it remains in state 1.
- If a cell is in state 0 and has exactly three neighbours in state 1, it enters state 1.
- In all other cases, the cell enters or remains in state 0.

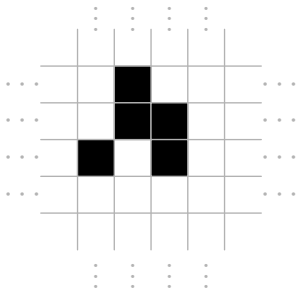
Consider the following initial configuration, where cells in states 0 and 1 are drawn as white and black squares respectively:



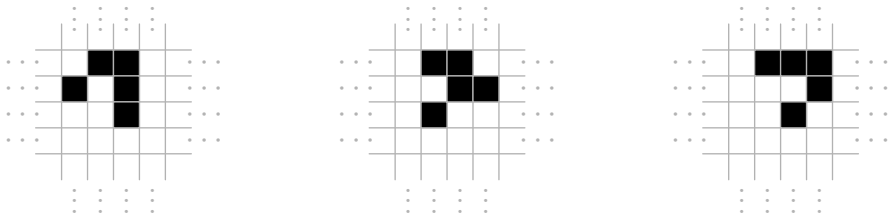
We now apply the local rule (Equation 2.3) across all cells. For example:

- Cell a is in state 0 and has three neighbours in state 1, so it enters state 1;
- Cell b is in state 0 but has only two neighbours in state 1, so it remains in state 0;
- Cell c is in state 1 but has only one neighbour in state 1, so it enters state 0;
- Cell d is in state 1 and has three neighbours in state 1, so it remains in state 1;
- Cell e is in state 1 and has two neighbours in state 1, so it remains in state 1.

Continuing in this way, the next configuration is:



and the next three configurations are:



Notice that the fifth configuration is identical to the first, but shifted one cell up and to the right. Due to the homogeneity of the CA's lattice, this pattern of cells in state 1 will continue to propagate diagonally in this way, moving up and to the right by one cell every four generations. This particular pattern is called the *glider*.

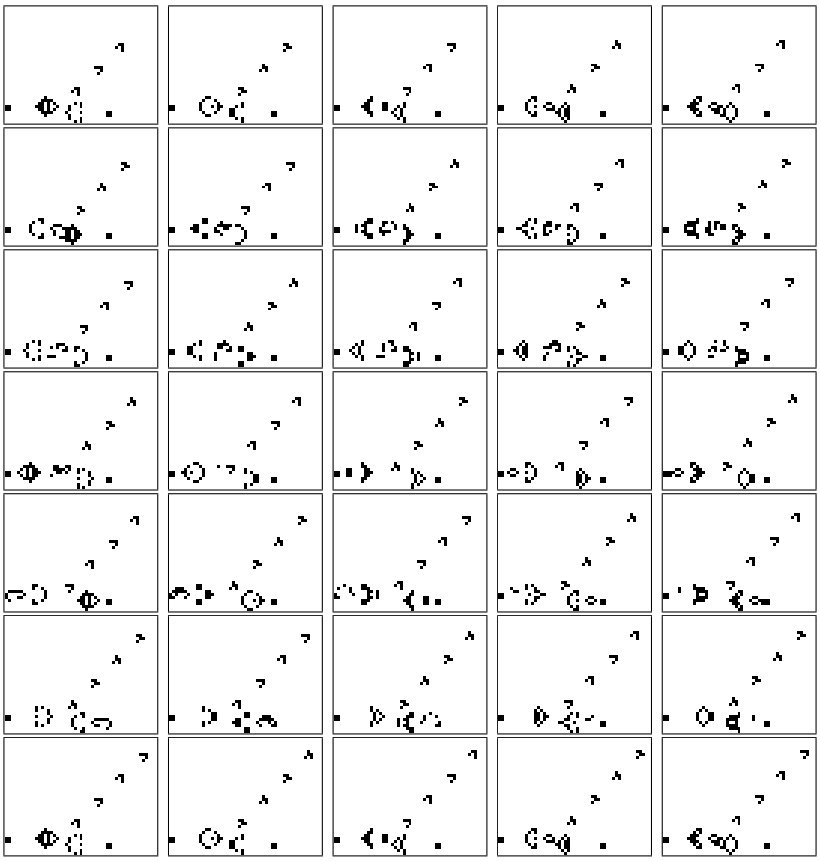


FIGURE 2.1. Successive generations of a glider gun in Conway's Game of Life. Note that the configuration at time $t+30$ is identical to that at time t with the addition of one extra glider.

Conway's Game of Life exhibits much more complex patterns than the glider. For example, the configuration shown in Figure 2.1 is known as a *glider gun*. Every 30 generations, a new glider is produced. This pattern demonstrates that, assuming an infinite lattice, the number of cells in state 1 can increase without limit as the CA evolves (in this case, that number increases by 5 every 30 generations).

The "Life Lexicon" [Sil06] lists several hundred patterns in Conway's Game of Life, of varying size and complexity.

Several other CAs have patterns similar to the gliders in Conway's Game of Life. Section 2.7 generalises the notion of gliders to other CAs.

2.3. 1-dimensional CAs and elementary CAs

A 1-dimensional CA has lattice $\mathbb{Z}_N$ in the finite case, or $\mathbb{Z}$ in the infinite case. (From this point, we frequently use N to denote the lattice size for finite 1-D CAs.) A configuration of a finite 1-D CA is a function $c : \mathbb{Z}_N \to S$.

It is convenient to write such a configuration as a string of length N over S, effectively changing the set of all configurations from $S^{\mathbb{Z}_N}$ to S^N. Clearly this notation is not quite so convenient in the infinite case, as the corresponding strings over S are infinite.

It is common to specify the neighbourhood template of a finite 1-D CA in terms of its *radius*. A CA with radius r has neighbourhood template $\langle -r, -r+1, \ldots, r-1, r\rangle$, so that each neighbourhood consists of $2r+1$ cells.

An *elementary CA (ECA)* is a 1-D CA with state set $\mathbb{Z}_2 = \{0, 1\}$ and radius $r = 1$. The local rule is thus a function $f : \mathbb{Z}_2^{2^3} \to \mathbb{Z}_2$; there are 256 such functions. Each local rule is assigned a number between 0 and 255 inclusive, by interpreting the string

$$f(1,1,1)f(1,1,0)\ldots f(0,0,1)f(0,0,0) \tag{2.5}$$

as an 8-bit binary number and converting to decimal.

Example 2.1. The decimal number 30 has binary representation 00011110, and so ECA rule 30 has the local rule defined by

xyz	111	110	101	100	011	010	001	000
$f(x,y,z)$	0	0	0	1	1	1	1	0

(2.6)

Take the initial configuration c to consist of a single cell in state 1, with all other cells in state 0:

$$c[i] = \begin{cases} 1 & \text{if } i = 0 \\ 0 & \text{if } i \neq 0 \end{cases} \tag{2.7}$$

Then, by applying the local rule at each cell, we obtain

$$F(c)[i] = \begin{cases} f(0,0,1) = 1 & \text{if } i = -1 \\ f(0,1,0) = 1 & \text{if } i = 0 \\ f(1,0,0) = 1 & \text{if } i = 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

and

$$F^2(c)[i] = \begin{cases} f(0,0,1) = 1 & \text{if } i = -2 \\ f(0,1,1) = 1 & \text{if } i = -1 \\ f(1,1,1) = 0 & \text{if } i = 0 \\ f(1,1,0) = 0 & \text{if } i = 1 \\ f(1,0,0) = 1 & \text{if } i = 2 \\ 0 & \text{otherwise} \end{cases} \tag{2.9}$$

and so on.

FIGURE 2.2. Space-time diagram showing 50 steps of the evolution of the CA in Example 2.1 (ECA rule 30), from an initial configuration consisting of a single cell in state 1.

Writing configurations as strings over S, we have

$$c_0 = \ldots 0001000 \ldots \tag{2.7′}$$

$$c_1 = \ldots 0011100 \ldots \tag{2.8′}$$

$$c_2 = \ldots 0110010 \ldots \tag{2.9′}$$

and so on.

In the 1-dimensional case, we can depict the sequence of configurations visited by a CA using a *space-time diagram*. Take a 2-D array of pixels, and let the pixel at position (x, y) (where the positive y-axis points downwards) correspond to the state of cell x at time y. So the x-axis corresponds to "space", and the y-axis to "time". We assign a colour to each state in S, and colour the pixels accordingly; when the state set is $\mathbb{Z}_2$, we generally use white for state 0 and black for state 1. The evolution described in Equations 2.7 to 2.9, and the 47 subsequent configurations, are shown in Figure 2.2. ◊

ECAs were first studied in detail by Stephen Wolfram in 1983 [Wol83], who continued to investigate them extensively [Wol86b, Wol94]. ECAs continue to be the subject of present research. The reason for this continued interest is that, despite the simplicity of their definition and the relatively small size of their rule space, ECAs exhibit a wide variety of behaviour, including Turing completeness (Section 2.8).

2.4. Essentially different rules

For an ECA local rule f, define the *conjugate* of f by

$$f_c(x, y, z) = 1 - f(1 - x, 1 - y, 1 - z) \tag{2.10}$$

and the *reflection* of f by

$$f_r(x, y, z) = f(z, y, x)\,. \tag{2.11}$$

Example 2.2. The conjugate of ECA rule 30 (Equation 2.6) is

$$\begin{aligned} f_c(0,0,0) &= 1 - f(1,1,1) = 1 & f_c(1,0,0) &= 0 \\ f_c(0,0,1) &= 1 - f(1,1,0) = 1 & f_c(1,0,1) &= 0 \\ f_c(0,1,0) &= 1 - f(1,0,1) = 1 & f_c(1,1,0) &= 0 \\ f_c(0,1,1) &= 1 - f(1,0,0) = 0 & f_c(1,1,1) &= 1\,; \end{aligned} \tag{2.12}$$

this is ECA rule 135. The reflection of rule 30 is

$$\begin{aligned} f_r(0,0,0) &= f(0,0,0) = 0 & f_r(1,0,0) &= 1 \\ f_r(0,0,1) &= f(1,0,0) = 1 & f_r(1,0,1) &= 0 \\ f_r(0,1,0) &= f(0,1,0) = 1 & f_r(1,1,0) &= 1 \\ f_r(0,1,1) &= f(1,1,0) = 0 & f_r(1,1,1) &= 0\,; \end{aligned} \tag{2.13}$$

this is ECA rule 86. ◊

These transformations leave the dynamics of the CA essentially unchanged: if the initial configuration undergoes a similar transformation (exchange of states 0 and 1, or left-right reflection) then the subsequent configurations undergo the same transformation but are otherwise unchanged. Transformations of this type are discussed further in Section 7.2.

Say that two ECA rules are "equivalent" if one can be obtained from the other by conjugation, by reflection, or by applying both of these operations in sequence. This partitions the rule space into equivalence classes; choosing a representative from each equivalence class yields the *essentially different* ECAs. By convention, the representative from each class is the rule whose number is smallest.

Theorem 2.3. *There are 88 essentially different ECAs.*

This well-known result is easy to verify by enumeration, but here we provide an analytical proof. (As far as we are aware, this result is not proved by this method in the literature; Li and Packard [LP90] use a similar method, but do not invoke the Orbit Counting Theorem directly.)

Proof. Let R be the set of ECA rules, and let $r : R \to R$ and $c : R \to R$ be the operations of reflection and conjugation respectively. Let $G = \{I, r, c, r \circ c\}$ where I is the identity function and $\circ$ denotes functional composition. It follows directly from the definitions that G is a group under $\circ$, and that G acts on R by function application (Definition A.20). By the Orbit Counting Theorem (Theorem A.27), the number of essentially different ECAs is

$$\frac{1}{|G|} \sum_{g \in G} \chi(g) = \frac{\chi(I) + \chi(c) + \chi(r) + \chi(r \circ c)}{4}\,, \tag{2.14}$$

where

$$\chi(g) = |\{f \in R \ : \ g(f) = f\}| \tag{2.15}$$

is the number of rules fixed by g.

Every rule is fixed by the identity function, so $\chi(I) = 256$.

For a rule to be fixed by c, we must have

$$1 - f(1,1,1) = f(0,0,0) \tag{2.16}$$

$$1 - f(1,1,0) = f(0,0,1) \tag{2.17}$$

$$1 - f(1,0,1) = f(0,1,0) \tag{2.18}$$

$$1 - f(1,0,0) = f(0,1,1)\,. \tag{2.19}$$

Such a rule is completely determined by choosing $f(0,0,0)$, $f(0,0,1)$, $f(0,1,0)$ and $f(0,1,1)$. There are $2^4 = 16$ choices, thus $\chi(c) = 16$. (Alternatively, there are four constraints on the rule, so the number of choices is $2^8/2^4 = 16$.)

For a rule to be fixed by r, we must have

$$f(0,0,1) = f(1,0,0) \tag{2.20}$$

$$f(0,1,1) = f(1,1,0)\,. \tag{2.21}$$

There are no constraints on $f(0,0,0)$, $f(0,1,0)$, $f(1,0,1)$ and $f(1,1,1)$. Thus $\chi(r) = 2^6 = 64$.

For a rule to be fixed by $r \circ c$, we must have

$$1 - f(1,1,1) = f(0,0,0) \tag{2.22}$$

$$1 - f(1,1,0) = f(1,0,0) \tag{2.23}$$

$$1 - f(1,0,1) = f(0,1,0) \tag{2.24}$$

$$1 - f(1,0,0) = f(1,1,0)\,. \tag{2.25}$$

Thus $\chi(r \circ c) = 16$.

Substituting into Equation 2.14, there are

$$\frac{256 + 16 + 64 + 16}{4} = 88 \tag{2.26}$$

essentially different ECA rules. □

The 88 essentially different ECAs are tabulated in Appendix B.

The fact that there are only 88 of them makes ECAs a particularly attractive subject for study: 88 is small enough to enumerate exhaustively, especially when compared to the number of rules for "larger" CAs. To emphasise this point, Table 2.1 shows the dramatic effect of increasing the neighbourhood radius and/or the number of states by 1, on both the total number of rules (given by $|S|^{|S|^{2r+1}}$) and the number of essentially different rules (obtained by a similar argument to the proof of Theorem 2.1; note

r	$\lvert S\rvert$	Total number of rules	Essentially different rules
1	2	256	88
2	2	4.29×10^9	1.07×10^9
1	3	7.63×10^{12}	6.36×10^{11}
2	3	8.72×10^{115}	7.27×10^{114}

TABLE 2.1. Numbers of rules for 1-dimensional CAs with the given neighbourhood radii and numbers of states.

that in the definition of "essentially different" for $|S| > 2$, the operation of conjugation is replaced with the group of all permutations of S).

2.5. Classification

Wolfram [Wol84] observes that ECAs (and CAs in general [PW85]) seem to exhibit four qualitative classes of behaviour (quoted descriptions are from Wolfram [Wol84]):

Class 1: "Evolution leads to a homogeneous state".

Class 2: "Evolution leads to a set of separated simple stable or periodic structures".

Class 3: "Evolution leads to a chaotic pattern".

Class 4: "Evolution leads to complex localized structures, sometimes long-lived".

Space-time diagrams illustrating these four classes are shown in Figure 2.3.

Li and Packard [LP90, LPL90] suggest a refinement of Wolfram's classification, in which class 2 is subdivided into three further classes:

Class LP2: Evolution leads to a heterogeneous fixed point.

Class LP3: Evolution leads to periodic behaviour, with period greater than 1.

Class LP4: Evolution leads to locally chaotic behaviour, with regions of chaos separated by fixed walls.

An example of class LP4 is shown in Figure 2.4. Classes LP1, LP5 and LP6 are equivalent to Wolfram's classes 1, 3 and 4 respectively.

It is worth noting that these classes are not formally defined, and classification of a given CA is a subjective matter. Much effort has been made to "formalise" these classes, or rather, to align these qualitative classes with quantitative properties which can be measured or proven. Sutner [Sut09] gives an overview of some of these efforts. Culik and Yu [CY88] suggest a formalism of Wolfram's classes in terms of computability theory, and show that classification according to this scheme is undecidable. This seems to

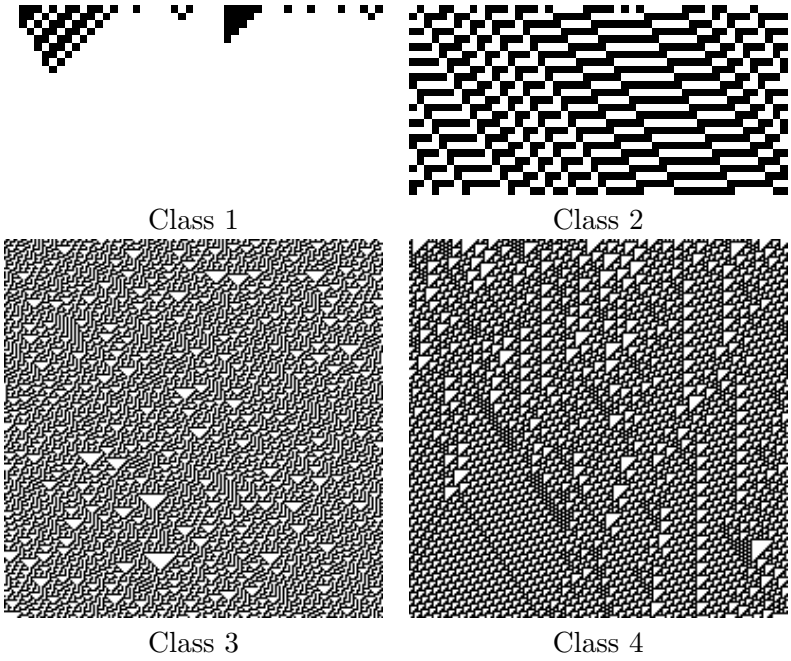


FIGURE 2.3. Examples of Wolfram's four classes. Space-time diagrams are shown for ECA rules 168 (class 1), 27 (class 2), 30 (class 3) and 110 (class 4). The lattice is $\mathbb{Z}_{50}$ for rules 168 and 27, and $\mathbb{Z}_{200}$ for the others. The initial configuration in each case is random, in the sense that each cell is assigned state 0 or 1, each with probability $\frac{1}{2}$.

suggest that Wolfram's classification is undecidable, to the extent that notions such as decidability can meaningfully be applied to something that is not formally defined to begin with. Despite this, more statistical methods (e.g. [OdOO01]) have been applied to the problem with some success.

Instead of starting with subjective descriptions of dynamics, an alternative approach to classification is to start with a formal classification and relate it back to qualitative behaviour. For example, Kůrka [Kůr97] suggests a classification scheme based on formal languages and the theory of dynamical systems (the notion of *equicontinuity*, discussed in Section 5.5, forms a part of this classification). While this approach has the advantage of providing unambiguous (and, hopefully, decidable) criteria for class membership, it still leaves the question of relating the classes back to the behaviour of the CA: Kůrka does give a mapping between his classification and Wolfram's, but by no means is this mapping obvious.

Although Wolfram's classification is not without detractors (e.g. [Epp]), it remains the most widely used classification scheme for CAs. We refer to Wolfram's classes throughout this thesis, but less as a rigid taxonomy

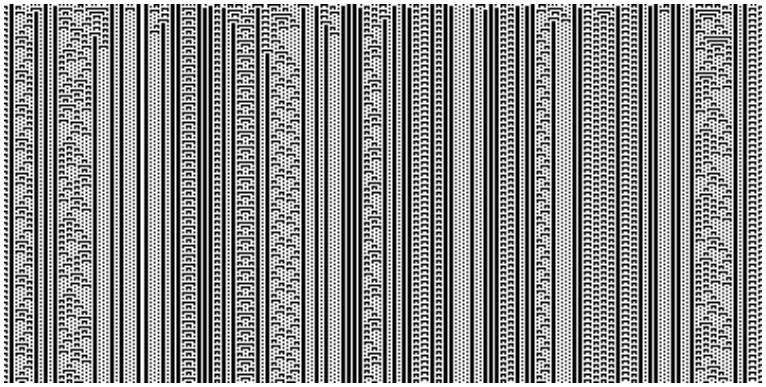


FIGURE 2.4. Space-time diagram for ECA rule 73 on the lattice $\mathbb{Z}_{400}$. This is an example of a CA in Li and Packard's class LP4.

and more as a shorthand for the trichotomy of simple versus chaotic versus complex behaviour.

Appendix B gives Wolfram classes for the 88 essentially different ECAs. We determined these classes by the following method. For several values of N (here $N = 98, \ldots, 102$), and several trials for each value of N (here 50 trials), choose a random initial configuration and allow the CA to evolve for a large number of time steps (here 500). There are three cases:

(1) If the final configuration is homogeneous for all N and all trials, then the ECA is in class 1.
(2) By examining the final four configurations, determine whether the ECA is periodic, modulo cyclic shift, with a period of 4 or less. If so for all N and all trials, then the ECA is in class 2.
(3) In all other cases, the class is determined by inspection. In particular, this is the case if the ECA exhibits aperiodic (or periodic with a period greater than 4) behaviour for *any* trial for any value of N.

The intention here is not to obtain a complete classification, but to eliminate those cases in which the classification is "obvious" so that only the non-obvious cases (case (3) above) require human intervention. The classification was found to be "obvious" in this sense for 68 of the 88 essentially different ECAs. Of the remaining 20, two (rules 54 and 110) are in class 4, thirteen (all the rules listed as class 3 in Appendix B) are in class 3, and five (rules 25, 26, 41, 94 and 154) are in class 2 but have periodic cycles of length greater than 4.

2.6. Speed of propagation

There is a limit to the speed at which information can propagate in a CA. Let c be the Euclidean distance between a cell and its most distant neighbour. The number of neighbours is finite, so c is finite. Now it is easy to see that information cannot possibly propagate over a distance of more than c cells in one time step; in effect, c is the "speed of light". For a 1-D CA with neighbourhood radius r, $c = r$; for Conway's Game of Life, $c = \sqrt{2}$ (the distance between a cell and its diagonal neighbour).

We say a state $q \in S$ is *quiescent* if $f(q, \ldots, q) = q$. If a region of the lattice has every cell in state q at time t, then that region, less a boundary of width c, must remain in state q at time $t + 1$.

In Example 2.1, state 0 is quiescent. At time $t = 0$, we know that $s_0(i) = 0$ for $|i| > 0$. Therefore at time $t = 1$, we will have $s_1(i) = 0$ for $|i| > 1$. In general we have $s_t(i) = 0$ for $|i| > t$, and so we only need to apply the local rule to the $2t + 1$ cells i with $|i| \leq t$, safely ignoring the rest of the cells.

More generally, say that a configuration is *finite* if only finitely many cells are not in some quiescent state q. Such an initial configuration guarantees existence of a hypercube, outside of which all cells are quiescent; this "active hypercube" then expands outwards at speed no greater than c. This idea is analogous to the concept of *light cones* in physics. This allows us to simulate an infinite lattice CA with finite (but unbounded as t increases) computational resources, provided the CA has a quiescent state and the initial configuration is finite.

2.7. Gliders

One phenomenon which is characteristic of class 4 CAs is that of "moving" structures. We have already seen an example of this: the glider in Conway's Game of Life (Section 2.2). This section generalises that notion.

Let c be a finite initial configuration of a CA with lattice $\mathbb{L}$. Say that c is a *glider* if, for some positive integer p and some non-zero $v \in \mathbb{L}$, we have

$$F^p(c)[i] = c[i - v] \tag{2.27}$$

for all cells i.

So evolving the CA for p generations is equivalent to shifting the configuration c by an offset v. For the glider in Conway's Game of Life, we have $p = 4$ and $v = (1, -1)$; by rotating the initial pattern, we can easily obtain gliders in Life for which v is any of the four vectors $(\pm 1, \pm 1)$.

The definition above requires the initial configuration to be finite (in the sense of Section 2.6); in fact, it is often useful to relax this condition

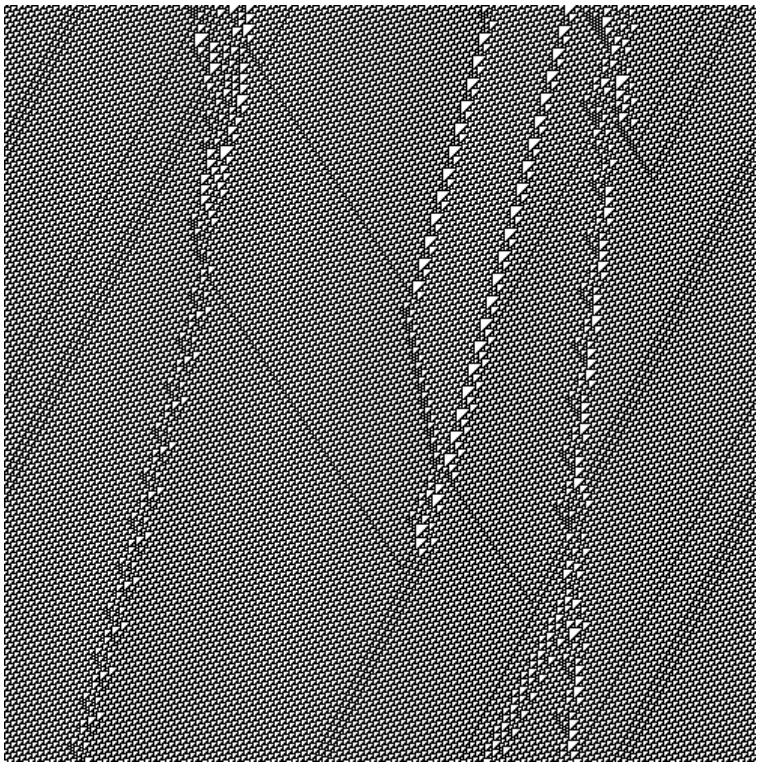


FIGURE 2.5. Gliders in rule 110.

and instead require only that the initial configuration is periodic outside a finite region of the lattice. This periodic pattern is the *ether* through which the glider moves. Figure 2.5 shows several gliders in ECA rule 110, moving through an ether which consists of repetitions of the string 00010011011111 (visible in the figure as a regular pattern of small triangles).

2.8. Turing completeness

It seems reasonable that simulation of a Turing machine must require "complex localized structures, sometimes long-lived", and so Turing completeness is sufficient for a CA to be in class 4. Wolfram [Wol84] conjectures that the converse is also true: every class 4 CA is Turing complete.

Strictly speaking, since the length of a Turing machine's tape is unbounded, Turing completeness is only possible in CAs with an infinite lattice.

Several CAs can be shown to be Turing complete:

2.8.1. Smith's CA

Smith [Smi71] shows that an arbitrary Turing machine (TM) with m tape symbols and n states can be simulated by a 1-D CA with neighbourhood

template $\langle -1, 0, 1, 2\rangle$ and $m + n$ states [Smi71, Theorem 7]. Furthermore, the simulation runs in *real-time*, meaning that one step in the execution of the TM is completed in one time step in the CA. Smith's construction works by associating the cells of the CA lattice with the cells of the TM tape, such that the state of a tape cell is encoded in the state of the corresponding CA cell. The CA rule is set up in such a way that it emulates the movement of the tape head across the static tape, thus simulating the TM.

It is possible to make trade-offs between the neighbourhood size, state set size and running time; indeed, Smith proves that the neighbourhood template $\langle 0, 1\rangle$ is sufficient, for a CA with $m(n + 3)$ states running in 2-times real time (i.e. with one step in the TM taking two steps in the CA) [Smi71, Theorem 5].

2.8.2. Conway's Game of Life

It is possible to construct logic gates, memory units etc. in Conway's Game of Life, by exploiting the different types of glider collision which can occur. This possibility is put forward by Conway [BCG82], and a detailed construction is provided by Rendell [Ren02]. Toffoli and Margolus [TM87] point out that such a construction is *"something of a* tour de force *— a bit like showing that one can make a computer out of the collisions of billiard balls!"*

2.8.3. ECA rule 110

Cook [Coo04] proves that the interactions between gliders in rule 110 (see Figure 2.5) can be used to implement a Turing complete system known as a *cyclic tag system*. If the construction in Conway's Game of Life is analogous to computing with billiard balls, the construction in rule 110 is akin to computing with waves colliding in a trough of water.

In Appendix B, we identify two ECAs as class 4, namely rules 54 and 110. If Wolfram's conjecture is true, this implies that rule 54 is also Turing complete; however, no proof (or disproof) of this exists.

2.8.4. Trid CAs

Cook's result implies that any system capable of simulating ECA rule 110 is capable of universal computation. This gives an easy way of proving Turing completeness for a variety of relatively simple systems.

A *Trid CA* is a 2-dimensional, binary state CAs with neighbourhood template $\langle (0, 1), (0, 0), (1, 0)\rangle$. A *Quad CA* is defined similarly, with $(1, 1)$ appended to the neighbourhood template. Toffoli [Tof08] posed the question of whether Quad or Trid CAs are Turing complete. Powley [Pow08]

shows that Trid CAs (and thus Quad CAs) are indeed Turing complete, by emulating ECA rule 110 in a Trid CA.

2.8.5. Discussion

As is often the case, this discussion of Turing machines is useful only for proving that CAs are in the class of computationally universal systems. If we want to use a CA to carry out an actual computation, direct simulation of a Turing machine is never the most practical choice. In particular, these simulations fail to exploit the parallelism inherent in the structure of the CA. The construction in Conway's Game of Life is a possible exception, as it simulates logic circuits instead of Turing machines, but it is still not the most efficient use of the CA's resources.

CHAPTER 3

Linear cellular automata

A common theme in mathematics is that linear systems tend to be easier to analyse than nonlinear systems. This is certainly true for CAs.

The structure of this chapter is as follows. Section 3.1 gives the necessary definitions. Section 3.2 shows how linear CAs can be studied in terms of finite rings of polynomials, specifically by encoding configurations and local rules as polynomials in such a way that the operation of the global map is equivalent to multiplication of polynomials. Section 3.3 uses this to give a "fast" algorithm for simulating linear CAs, specifically an algorithm that can simulate t time steps of the CA in $O(\log t)$ time. Section 3.4 uses polynomials to prove several properties of a particular linear CA, namely ECA rule 90.

3.1. Definition

Consider a CA whose state set S is a commutative ring. Commutative rings are defined in Definition A.29; the most important point is that S has commutative operations of "addition" and "multiplication" that satisfy the usual properties. A common choice for S is $\mathbb{Z}_n$, with the operations of addition and multiplication modulo n.

Such a CA is *linear*, or *additive*, if its local rule has the form

$$f(x_1, \ldots, x_m) = \lambda_1 x_1 + \cdots + \lambda_m x_m \tag{3.1}$$

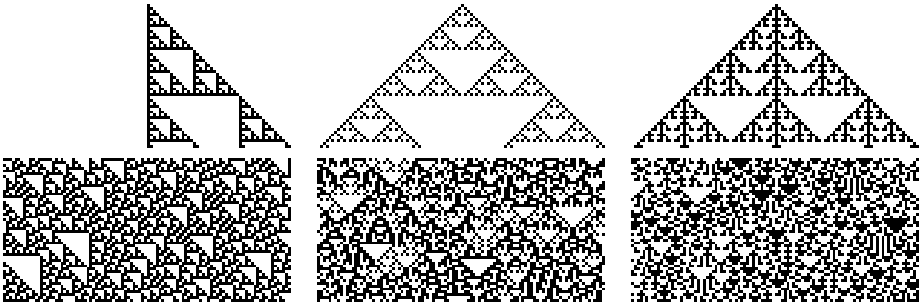
for some constants $\lambda_1, \ldots, \lambda_m \in S$. In other words, the local rule is a linear function (a polynomial of degree 1), with zero constant term. A linear CA is *non-trivial* if more than one of the coefficients $\lambda_1, \ldots, \lambda_m$ is nonzero, otherwise it is *trivial*.

Example 3.1. Of the 88 essentially different ECA rules, six are linear:

$$f(x_{-1}, x_0, x_1) = 0 \qquad \text{(rule 0)} \tag{3.2}$$

$$f(x_{-1}, x_0, x_1) = x_1 \qquad \text{(rule 170)} \tag{3.3}$$

$$f(x_{-1}, x_0, x_1) = x_0 \qquad \text{(rule 204)} \tag{3.4}$$



Rule 60 Rule 90 Rule 150

FIGURE 3.1. Space-time diagrams for the three non-trivial linear ECAs, on an initial configuration consisting of a single cell in state 1 (top) and an initial configuration in which each cell is randomly assigned state 0 or 1 (bottom).

$$f(x_{-1}, x_0, x_1) = x_{-1} + x_0 \qquad \text{(rule 60)} \qquad (3.5)$$

$$f(x_{-1}, x_0, x_1) = x_{-1} + x_1 \qquad \text{(rule 90)} \qquad (3.6)$$

$$f(x_{-1}, x_0, x_1) = x_{-1} + x_0 + x_1 \qquad \text{(rule 150)} \qquad (3.7)$$

Rules 0, 170 and 204 are trivial, and rules 60, 90 and 150 are non-trivial. In terms of global maps, rule 0 immediately maps every configuration to the homogeneous configuration of zeroes, rule 170 shifts the entire configuration by one cell to the left, and rule 204 is the identity. The global dynamics of the other three rules are illustrated in Figure 3.1. ♢

For two configurations u, v, define the sum $u + v$ by

$$(u + v)[i] = u[i] + v[i] \qquad (3.8)$$

for all $i \in \mathbb{L}$. Configurations of a linear CA obey a law of additive superposition:

Theorem 3.2. *Let F be the global map for a linear CA, and let u, v be two configurations. Then*

$$F(u + v) = F(u) + F(v)\,. \qquad (3.9)$$

PROOF. Follows directly from the definitions. □

3.2. As polynomials

Martin et al [MOW84] study linear CAs in terms of finite rings of polynomials. This technique allows a large body of work from algebra to be applied to linear CAs.

Consider a finite 1-D linear CA on N cells, whose state set S is a field (Definition A.30). A field is a ring in which every nonzero element has a multiplicative inverse. For example, $\mathbb{Z}_p$ is a field if (and only if) p is prime.

Denote by R_N^S the set of all polynomials of degree less than N with coefficients in S. Define two binary operations $+$ and $\times$ on R_N^S, corresponding to the usual operations of addition and multiplication of polynomials, but "factoring out" the equation $x^N - 1 = 0$. Effectively, all powers of x are computed modulo N. Under these operations, R_N^S is a commutative ring.

Example 3.3. Let $S = \mathbb{Z}_2$, and let $N = 4$. Consider the following two polynomials in $R_4^{\mathbb{Z}_2}$:

$$f(x) = x + x^2 + x^3 \tag{3.10}$$

$$g(x) = 1 + x^2. \tag{3.11}$$

Then

$$\begin{aligned} f(x) + g(x) &= (x + x^2 + x^3) + (1 + x^2) && (3.12)\\ &= 1 + x + 2x^2 + x^3 && (3.13)\\ &= 1 + x + x^3 && (3.14)\\ f(x) \times g(x) &= (x + x^2 + x^3) \times (1 + x^2) && (3.15)\\ &= (x + x^2 + x^3) + (x^3 + x^4 + x^5) && (3.16)\\ &= x + x^2 + x^4 + x^5 && (3.17)\\ &= x + x^2 + 1 + x && (3.18)\\ &= 1 + x^2. && (3.19) \end{aligned}$$

In Equations 3.14, 3.17 and 3.19, we make use of the fact that $2 \equiv 0$ in $\mathbb{Z}_2$. In Equation 3.18, we "factor out" $x^4 - 1 = 0$ to reduce indices to their residues modulo 4, so that x^4 becomes $x^0 = 1$ and x^5 becomes $x^1 = x$. $\Diamond$

A formal construction of R_N^S is given in Appendix A.5.

There is a bijective mapping between configurations of the CA and elements of R_N^S: specifically, map the configuration c to the polynomial A_c defined by

$$A_c(x) = c[0] + c[1]x + c[2]x^2 + \cdots + c[N-1]x^{N-1}\,. \tag{3.20}$$

Let the CA's local rule be defined by

$$f(a_{-r}, \ldots, a_r) = \lambda_{-r}a_{-r} + \cdots + \lambda_r a_r\,. \tag{3.21}$$

We associate with f a polynomial T_f:

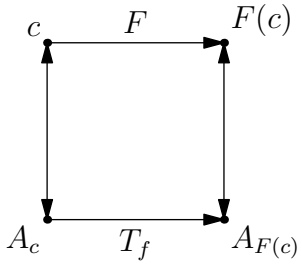
$$T_f(x) = \lambda_{-r}x^r + \cdots + \lambda_r x^{-r}\,. \tag{3.22}$$

Recall that powers of x are computed modulo N in R_N^S, so a polynomial can have "negative" powers of x; specifically, $x^{-i} = x^{N-i}$.

Theorem 3.4. *Let c be a configuration of a linear CA. Then*

$$A_{F(c)}(x) = T_f(x) A_c(x)\,. \tag{3.23}$$

In other words, applying the global map F to a configuration and then finding the associated polynomial is equivalent to multiplying the configuration's associated polynomial by the polynomial T_f. This is illustrated in the following commutative diagram:



PROOF. The x^i term in $T_f(x)A_c(x)$ is

$$\sum_{j=-r}^{r} \left(\lambda_j x^{-j}\right)\left(c[i+j]x^{i+j}\right) = \left(\sum_{j=-r}^{r} \lambda_j c[i+j]\right) x^i \tag{3.24}$$

$$= f(c[i-r], \ldots, c[i+r])x^i \tag{3.25}$$

$$= F(c)[i]x^i\,, \tag{3.26}$$

which is the x^i term in $A_{F(c)}(x)$. Hence the polynomials $T_f(x)A_c(x)$ and $A_{F(c)}(x)$ are equal. □

Example 3.5. The polynomial associated with ECA rule 60 (Equation 3.5) is

$$T_f(x) = x + 1\,. \tag{3.27}$$

Let $N = 8$, and consider the configuration $c = 00110001$. The associated polynomial is

$$A_c(x) = x^2 + x^3 + x^7\,. \tag{3.28}$$

Now

$$T_f(x)A_c(x) = (x+1)(x^2 + x^3 + x^7) \tag{3.29}$$

$$= x^3 + x^4 + x^8 + x^2 + x^3 + x^7 \tag{3.30}$$

$$= 1 + x^2 + x^4 + x^7\,. \tag{3.31}$$

The configuration associated with this polynomial is 10101001. It is easy to verify that $F(c) = 10101001$. ◊

```
1: let t_k t_{k-1} ... t_0 be the binary representation of t        ▷ k = ⌈log_2 t⌉
2: p(x) ← T_f(x)
3: q(x) ← 1
4: for i = 0, ..., k do
5:     if t_i = 1 then
6:         q(x) ← q(x) × p(x)
7:     p(x) ← p(x)^2                                    ▷ now p(x) = T_f(x)^{2^{i+1}}
8: now q(x) = T_f(x)^t
```

Algorithm 3.1: Finding $T_f(x)$ in time $O(\log t)$.

3.3. Fast simulation of linear CAs

For a general (nonlinear) CA, there is no faster way to compute the configuration at time t than letting the CA run for t time steps. For 1-D linear CAs, however, it is possible to compute the configuration at time t in $O(\log t)$ operations. To see this, note that the configuration at time t is $T_f(x)^t A_c(x)$, where $T_f(x)$ is the polynomial associated with the local rule and $A_c(x)$ is the polynomial associated with the configuration at time 0. Now $T_f(x)^t$ can be found in time $O(\log t)$, by writing t in binary notation and repeatedly squaring $T_f(x)$; see Algorithm 3.1.

In fact, the repeated squaring is not necessary if the state set is $\mathbb{Z}_2$:

Lemma 3.6. *In $R_N^{\mathbb{Z}_2}$, we have*

$$\left(\sum_{i=-r}^{r} a_i x^i\right)^{2^k} = \sum_{i=-r}^{r} \left(a_i x^{2^k i}\right) \tag{3.32}$$

for all nonnegative integers k.

This result applies to linear CAs by substituting λ_{-i} for a_i, giving an expression for $T_f(x)^{2^k}$.

Proof. By induction on k. The key observation is that when an expression of the form

$$\left(\sum_{i=-r}^{r} a_i x^{2^k i}\right)^2 \tag{3.33}$$

is expanded, all but the $\left(x^{2^k i}\right)^2$ terms have a coefficient of 2 (which is equivalent to 0 in $\mathbb{Z}_2$) and thus vanish. □

Example 3.7. The polynomial associated with ECA rule 60 (Equation 3.5) is

$$T_f(x) = x + 1\,. \tag{3.34}$$

By Lemma 3.6, we have

$$T_f(x)^{2^k} = x^{2^k} + 1 \tag{3.35}$$

for all nonnegative integers k.

Given that $100 = 64 + 32 + 4$, we thus have

$$T_f(x)^{100} = T_f(x)^{64} T_f(x)^{32} T_f(x)^4 \tag{3.36}$$
$$= (x^{64} + 1)(x^{32} + 1)(x^4 + 1) \tag{3.37}$$
$$= x^{100} + x^{96} + x^{68} + x^{64} + x^{36} + x^{32} + x^4 + 1\,. \tag{3.38}$$

Let $N = 9$. In $R_9^{\mathbb{Z}_2}$, this reduces to

$$T_f(x)^{100} = x^1 + x^6 + x^5 + x^1 + x^0 + x^5 + x^4 + 1 \tag{3.39}$$
$$= x^4 + x^6\,. \tag{3.40}$$

Let $c = 000110001$. Then

$$T_f(x)^{100} A_c(x) = (x^4 + x^6)(x^3 + x^4 + x^8) \tag{3.41}$$
$$= x^7 + x^8 + x^9 + x^{10} + x^{12} + x^{14} \tag{3.42}$$
$$= 1 + x + x^3 + x^5 + x^7 + x^8\,. \tag{3.43}$$

This implies that $F^{100}(c) = 110101011$, which can easily be verified. ◊

3.4. Properties of ECA rule 90

Martin et al [MOW84] use algebraic techniques to prove several properties of the linear ECA rule 90. The properties themselves are fairly unremarkable, given that rule 90 is not among the more complex ECAs, but it is worthwhile noting the relative ease with which they are proved, and contrasting this with the difficulties encountered when trying to prove similar properties for nonlinear CAs.

From Equation 3.6, the polynomial associated with rule 90 is

$$T_f(x) = x + x^{-1}. \tag{3.44}$$

A *Garden of Eden* configuration is a configuration c such that there is no configuration b with $F(b) = c$. Garden of Eden configurations are discussed further in Chapter 4.

Lemma 3.8 ([MOW84, Lemma 3.1])**.** *Configurations containing an odd number of cells in state* 1 *are Garden of Eden configurations.*

Proof. Notice that, for a polynomial $A(x)$, the sum of A's coefficients is given by $A(1)$. So for the polynomial $A_c(x)$ associated with a configuration of an ECA, $A_c(1)$ is congruent modulo 2 to the number of cells in c in state 1. In other words, $A_c(1) = 1$ if and only if the number of cells in state 1 is odd.

Now let c_0 be a configuration, and consider the next configuration $c_1 = F(c_0)$. We have

$$A_{c_1}(x) = T_f(x)A_{c_0}(x) = (x + x^{-1})A_{c_0}(x). \tag{3.45}$$

But $(1 + 1^{-1}) = 0$ in $\mathbb{Z}_2$, so we must have $A_{c_1}(1) = 0$. This shows that an even number of cells in c_1 are in state 1.

This argument works for all configurations c_0, so a configuration with an odd number of cells in state 1 must be a Garden of Eden configuration. □

Theorem 3.9 ([MOW84, Theorem 3.1 (b)])**.** *If N is odd, exactly half of all configurations are Garden of Eden configurations.*

Proof. By Lemma 3.8, it suffices to show that configurations with an even number of cells in state 1 are not Garden of Eden configurations. Let c be such a configuration. It can be shown that, since $A_c(1) = 0$ and $x - 1$ is a factor of $x^N - 1$ for all N, then $x - 1$ is a factor of $A_c(x)$. We are working in $\mathbb{Z}_2$, so $x - 1 = x + 1$. Thus there exists a polynomial $B(x)$ in R_N^S such that

$$A_s(x) = (x + 1)B(x). \tag{3.46}$$

Since N is odd, it is easy to verify that

$$x + 1 = (x + x^{-1})(x^2 + x^4 + \cdots + x^{N-1}) \tag{3.47}$$

in R_N^S. Substituting into Equation 3.46 gives

$$A_c(x) = (x + x^{-1})(x^2 + x^4 + \cdots + x^{N-1})B(x) \tag{3.48}$$

$$= T_f(x)(x^2 + x^4 + \cdots + x^{N-1})B(x). \tag{3.49}$$

Thus the configuration associated with the polynomial $(x^2+\cdots+x^{N-1})B(x)$ is mapped to c by the global map, and so c is not a Garden of Eden configuration. □

Theorem 3.10 ([MOW84, Theorem 3.1 (a)])**.** *If N is even, exactly three quarters of all configurations are Garden of Eden configurations.*

Proof (sketch). Let c_0, c_1 be configurations such that $F(c_0) = c_1$. We have

$$A_{c_1}(x) = (x + x^{-1})A_{c_0}(x), \tag{3.50}$$

and so

$$A_{c_1}(x) = (x^2 + 1)B(x) \tag{3.51}$$

for $B(x) = x^{-1}A_{c_0}(x)$. It can be shown, making use of the fact that N is even, that $\deg B(x) < N - 2$. Thus there are 2^{N-2} possible choices of $B(x)$, hence 2^{N-2} possibilities for $A_{c_1}(x)$, hence 2^{N-2} possibilities for c_1. These

2^{N-2} possibilities are one quarter of the entire configuration space, thus the remaining three quarters are all Garden of Eden configurations. □

Martin et al also prove some results about the topology of the transition graphs (Definition 6.1) for rule 90:

Theorem 3.11 ([MOW84, Theorem 3.3]). *If N is odd, all trees in the transition graph consist of a single edge.*

This result is illustrated in Figure 6.6.

Theorem 3.12 ([MOW84, Theorem 3.4]). *If N is even, all trees in the transition graph have the following properties:*

(1) *The distance from the root vertex to every leaf vertex is*
$$\frac{1}{2}\max\left\{2^j \;:\; 2^j \text{ is a factor of } N\right\}; \tag{3.52}$$
(2) *The root vertex has in-degree 3;*
(3) *Every non-root non-leaf vertex has in-degree 4.*

This result is illustrated in Figure 6.7, where Equation 3.52 gives the distance from root to leaf as $\frac{1}{2}2^2 = 2$.

Theorem 3.13 ([MOW84, Lemma 3.4]). *Let Π_N be the length of the cycle obtained from an initial configuration with a single cell in state 1 and all other cells in state 0. Then all other cycle lengths on the lattice $\mathbb{Z}_N$ are factors of Π_N.*

In fact, Theorem 3.13 holds for all linear CAs.

Theorem 3.14 ([MOW84, Lemmas 3.5 and 3.6]). *If N is a power of 2, $\Pi_N = 1$. If N is even but not a power of 2, $\Pi_N = 2\Pi_{\frac{N}{2}}$.*

The characterisation of Π_N is not quite so neat when N is odd:

Theorem 3.15 ([MOW84, Theorem 3.5]). *If N is odd, Π_N is a factor of $2^j - 1$ where*
$$j = \min\left\{i > 0 \;:\; 2^i \equiv \pm 1 \mod N\right\} \tag{3.53}$$

For example, Theorem 3.14 gives $\Pi_{12} = 2\Pi_6 = 4\Pi_3$. Now $2^1 \equiv -1 \mod 3$, so Theorem 3.15 tells us that Π_3 is a factor of $2^1 - 1$. This forces $\Pi_3 = 1$, thus $\Pi_{12} = 4$.

Also, since $2^5 = 32 \equiv -1 \mod 11$, Theorem 3.15 tells us that Π_{11} is a factor of $2^5 - 1 = 31$. But 31 is prime, so we must have either $\Pi_{11} = 1$ or $\Pi_{11} = 31$. It turns out that $\Pi_{11} = 31$.

Martin et al observe that $\Pi_N = 2^j - 1$ for the majority of odd N, the first few exceptions being

$$N = 37, 95, 101, 141, 197, 199, 203, \ldots . \tag{3.54}$$

In the notation of Equation 8.26, j is the value of the *suborder function* of 2 modulo N: $j = \mathrm{sord}_N(2)$.

3.5. Summary

The methods used by Martin et al to study rule 90 generalise to other linear CAs, including those in higher dimensions: rules and configurations in a D-dimensional CA are represented by polynomials in D variables. Of course, the methods do not generalise to nonlinear CAs.

We normally think of emergence as occurring in systems where "the whole is greater than (or at least different from) the sum of the parts". However, the principle of additive superposition (Theorem 3.2) suggests that the whole of a linear CA is *exactly* the sum of the parts. Does this mean that linear CAs cannot exhibit emergent properties? It is certainly true that linear CAs have interesting properties; whether these properties are classed as "emergent" or not depends more on the definition of emergence being used than on the nature of the system itself.

CHAPTER 4

Preimages

The *preimages* of a CA configuration u are those configurations v such that $F(v) = u$. In other words, the preimages of the configuration at time t are all the possible configurations at time $t - 1$.

A configuration of a CA is said to be *reachable* if it has at least one preimage, and *unreachable* otherwise. An unreachable configuration can only appear as the initial configuration of the CA; a reachable configuration can also appear on subsequent time steps. Unreachable configurations are often referred to as *Garden of Eden* configurations.

If no configuration of the CA has more than one preimage, then the CA is said to be *injective*. Equivalently, a CA is injective if and only if its global map is an injective (one-one) function. If every configuration is reachable, the CA is said to be *surjective*; this is the case if and only if the global map is a surjective (onto) function. If a CA is both injective and surjective, so that the global map is a bijection, then the CA is said to be *reversible*. Configurations of a reversible CA have exactly one preimage, so the CA "running in reverse" is a deterministic system (although not necessarily a CA itself).

It is well known that a function on a finite set is injective if and only if it is surjective. Hence the words "injective", "surjective" and "reversible" are synonymous for CAs on finite lattices, although not for CAs on infinite lattices.

Reversible CAs are exceptional: most CAs have Garden of Eden configurations and configurations with multiple preimages. In this case, the CA "in reverse" is not deterministic. Among the essentially different ECAs, only rules 204 (identity), 170 (shift left), 51 (exchange states 0 and 1) and 15 (exchange 0 and 1 and shift to the right) are reversible. Some ECAs are reversible for particular numbers of cells N; for example, rule 150 is reversible if and only if N is not a multiple of 3.

This chapter discusses the problems of finding and counting preimages of CA configurations. Section 4.1 gives an efficient algorithm for finding preimages, and Section 4.2 shows that this algorithm can be made even more efficient if the CA is linear. Section 4.3 describes how the number of preimages can be found without finding the preimages themselves.

```
1: procedure FindPreimages(c)
2:     for a ∈ S^{2r} do
3:         BuildPreimage(c, a)
4: procedure BuildPreimage(c, a)
5:     N ← length(c)
6:     k ← length(a)                                  ▷ we know 2r ≤ k < N
7:     for b ∈ S do
8:         if f(a[k − 2r], ..., a[k − 1], b) = c[k − r] then ▷ b is valid choice
9:             if k + 1 = N then                          ▷ length(ab) = N
10:                if F(ab) = c then                      ▷ see note in text
11:                    output ab as a preimage of c
12:            else                                       ▷ length(ab) < N
13:                BuildPreimage(c, ab)
```

Algorithm 4.1: The reverse algorithm, for finding the preimages of a given configuration c.

4.1. The reverse algorithm

Wuensche [WL92] gives an algorithm for finding preimages of configurations in 1-dimensional cellular automata. The algorithm works by iteratively building preimages from left to right. Suppose we have the partial preimage $a_0 \dots a_{k-1}$, where $k < N$. If we append state b, the newly completed neighbourhood of cell $k - r$ is $a_{k-2r} \dots a_{k-1}b$. For this to be a valid partial configuration, we want $f(a_{k-2r} \dots a_{k-1}b) = c_{k-r}$. If this is the case then the algorithm continues with the new partial configuration, otherwise this choice of b is rejected. This procedure continues until the partial preimage has length N.

At each stage in the algorithm, all possible choices for new state b must be tried. This is generally done by iterating over the choices for b, and backtracking whenever the algorithm either finds a candidate preimage or reaches a dead end where no choice of b is valid. Alternatively, a parallel implementation of the algorithm could explore all choices of b simultaneously.

Algorithm 4.1 describes the reverse algorithm in full, using recursion to implement backtracking. Note that this algorithm only works for $N > 2r$; for smaller N, exhaustive search can be used instead.

It is not strictly necessary to check $F(ab) = c$ at line 10: we already know, from previous executions of line 8, that c and $F(ab)$ agree on cells r to $N - 1 - r$ inclusive, so only the $2r$ cells outside this range need to be checked.

Clearly the time complexity of this algorithm is exponential with respect to N in the worst case, as the number of preimages is exponential in the

worst case. However, this worst case is relatively rare, and in most typical cases the running time is closer to linear than exponential.

4.2. The reverse algorithm for linear CAs

The need for backtracking or parallelism in the reverse algorithm arises from the possibility of multiple ways to extend the partial configuration. In certain cases, there is at most one way to extend the partial configuration, and so the backtracking or parallelism can be eliminated. This is the case when Wuensche's Z parameter (Section 5.2) is equal to 1. This section describes a family of such cases. (As far as we are aware, this work is original.)

Note that there being only one way to extend the partial preimage does not imply that there is only one preimage. The algorithm begins with multiple partial preimages, so may still end with multiple preimages.

Recall from Section 3.1 that a linear CA has state set $\mathbb{Z}_s$, and local rule

$$f(x_{-r}, \ldots, x_r) = \lambda_{-r} x_{-r} + \cdots + \lambda_r x_r \tag{4.1}$$

for some constants $\lambda_{-r}, \ldots, \lambda_r \in \mathbb{Z}_s$. For a linear CA, the condition in Algorithm 4.1 line 8 becomes

$$\lambda_{-r} a[k-2r] + \cdots + \lambda_{r-1} a[k-1] + \lambda_r b = c[k-r]\,, \tag{4.2}$$

or equivalently,

$$\lambda_r b = c[k-r] - \lambda_{-r} a[k-2r] - \cdots - \lambda_{r-1} a[k-1]\,. \tag{4.3}$$

If λ_r has a multiplicative inverse in $\mathbb{Z}_s$, then there is exactly one solution for b in this equation, and that solution can easily be computed. Thus the algorithm can be rewritten as shown in Algorithm 4.2, or without recursion as in Algorithm 4.3. For λ_r to have a multiplicative inverse, λ_r must be nonzero, and λ_r and s must be *coprime* (having no common factors other than 1). In particular, this is the case if s is prime (so that the state set $\mathbb{Z}_s$ is a field), or if λ_r is either prime or equal to 1.

The algorithm can be modified to apply to a wider class of linear CAs. Instead of requiring that λ_r has an inverse, we can require that the rightmost nonzero coefficient (λ_i such that $\lambda_{i+1} = \cdots = \lambda_r = 0$) has an inverse. In this case, extending a partial preimage of length k "completes" the neighbourhood of cell $k-i$; the unknown states to the right of cell $k-1$ can be ignored since the associated coefficients are zero.

We can also modify the algorithm so that preimages are built right-to-left instead of left-to-right; in this case, we require that the *leftmost* nonzero coefficient has an inverse. With these modifications, the algorithm applies

```
1: procedure BuildPreimage(c, a)
2:     N ← length(c)
3:     k ← length(a)
4:     b ← unique solution to Equation 4.3
5:     if k + 1 = N then
6:         if F(ab) = c then
7:             output ab as a preimage of c
8:     else
9:         BuildPreimage(c, ab)
```

Algorithm 4.2: The reverse algorithm for linear CAs in which the coefficient λ_r has a multiplicative inverse.

```
1: procedure BuildPreimage(c, a)
2:     N ← length(c)
3:     while length(a) < N do
4:         b ← unique solution to Equation 4.3
5:         append b to a
6:     if F(a) = c then
7:         output a as a preimage of c
```

Algorithm 4.3: A non-recursive version of Algorithm 4.2.

to all linear ECAs except rule 0, and indeed to every nonzero linear CA whose state set is a field.

This algorithm places an upper bound of s^{2r} on numbers of preimages in these linear CAs: each call to BuildPreimage yields at most one preimage, and BuildPreimage is called exactly s^{2r} times by FindPreimages.

4.3. Counting preimages with de Bruijn matrices

McIntosh [McI90] describes a method of counting preimages for CA configurations, by means of graphs (or more correctly, adjacency matrices of graphs) first introduced by de Bruijn [dB46]; see also [Ral82]. Jeras and Dobnikar [JD06] describe the method more fully.

For a state $s \in S$, the *de Bruijn graph* G_s is the directed bipartite graph (Definition A.55) with vertex set

$$\left\{u_L \ : \ u \in S^{2r}\right\} \cup \left\{u_R \ : \ u \in S^{2r}\right\} \tag{4.4}$$

and edge set

$$\left\{(x_{-r} \dots x_{r-1})_L \to (x_{-r+1} \dots x_r)_R \ : \ f(x_{-r}, \dots, x_r) = s\right\} . \tag{4.5}$$

In other words, the vertex set consists of two copies of S^{2r}, with subscripts L and R. The graph is bipartite, so all edges are directed from L to R. There is an edge from vertex u_L to vertex v_R, where $u, v \in S^{2r}$, if and only if the following two conditions hold: the last $2r - 1$ characters of u are the same

as the first $2r-1$ characters of v, and the string of length $2r+1$ obtained by appending the last character of v to u (or equivalently, prepending the first character of u to v) is mapped to s by the local rule f.

Example 4.1. The de Bruijn graphs for an ECA have vertex set

$$\{00_L, 01_L, 10_L, 11_L, 00_R, 01_R, 10_R, 11_R\} , \tag{4.6}$$

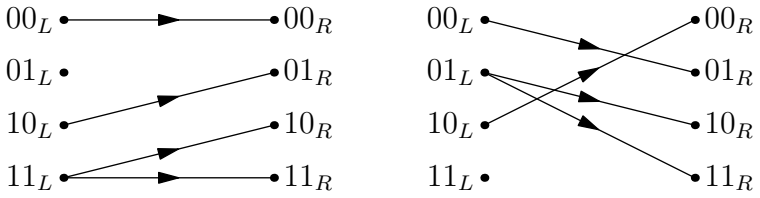
and there is an edge from ab_L to bc_R if and only if $f(a,b,c) = s$.

ECA rule 30 is defined by

xyz	111	110	101	100	011	010	001	000
$f(x,y,z)$	0	0	0	1	1	1	1	0

(4.7)

The de Bruijn graphs for states 0 and 1 are



respectively. ◊

The *preimage network* for a configuration c is the graph with vertex set

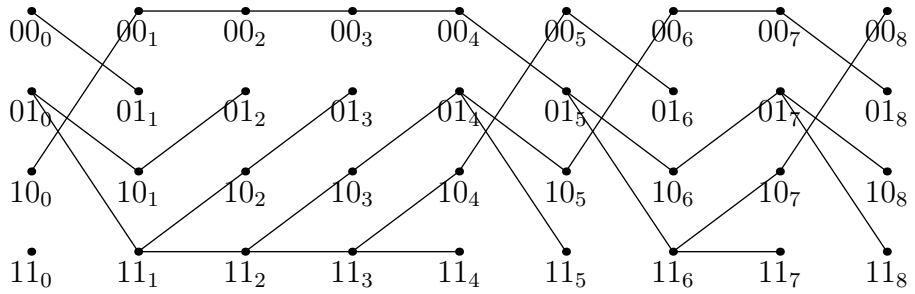
$$\left\{u_0 \ : \ u \in S^{2r}\right\} \cup \cdots \cup \left\{u_N \ : \ u \in S^{2r}\right\} , \tag{4.8}$$

and edge set

$$\{(x_{-r} \ldots x_{r-1})_i \to (x_{-r+1} \ldots x_r)_{i+1} \ : \ f(x_{-r}, \ldots, x_r) = c[i]\} . \tag{4.9}$$

In other words, the preimage network is constructed by "concatenating" the de Bruijn graphs for states $c[0], \ldots, c[N-1]$, so that the right-hand vertices of one graph become the left-hand vertices of the next.

Example 4.2. The preimage network for configuration 10001101 of ECA rule 30 is



◊

Theorem 4.3. *Consider the preimage network for configuration c. Let P_u be the number of paths in the preimage network from u_0 to u_N, and let $P = \sum_{u \in S^{2r}} P_u$. Then c has exactly P preimages.*

PROOF. We proceed by showing that preimages of c are in one-one correspondence with those paths counted by P.

Suppose that v is a preimage of c, so that $F(v) = c$. By definition of the global map, this means that

$$f(v[i-r], \ldots, v[i+r]) = c[i] \tag{4.10}$$

for all i, where the indices are computed modulo N. By definition of the preimage network, this implies that there exists an edge

$$(v[i-r] \ldots v[i+r-1])_i \to (v[i-r+1] \ldots v[i+r])_{i+1} \tag{4.11}$$

for all i. These edges form a path in the preimage network, from $(v[N-r] \ldots v[r-1])_0$ to $(v[N-r] \ldots v[r-1])_N$. Thus for each preimage there is a path counted by P.

Conversely, consider a path counted by P. By definition of the preimage network, such a path can be written as

$$(v[N-r] \ldots v[r-1])_0 \to (v[N-r+1] \ldots v[r])_1 \to \cdots \to (v[N-r] \ldots v[r-1])_N \tag{4.12}$$

for some states $v[0], \ldots, v[N-1]$. Existence of an edge

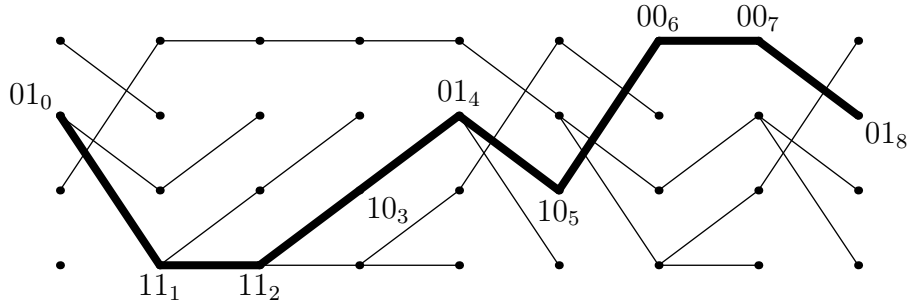
$$(v[i-r] \ldots v[i+r-1])_i \to (v[i-r+1] \ldots v[i+r])_{i+1} \tag{4.13}$$

implies that

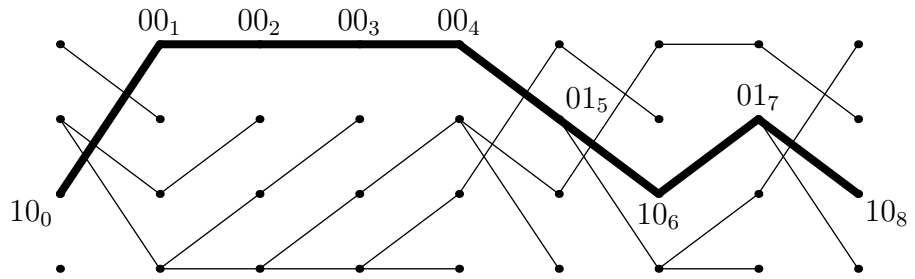
$$f(v[i-r], \ldots, v[i+r]) = c[i]\,. \tag{4.14}$$

This holds for all i, so $F(v) = c$ by definition of the global map. Thus for each path counted by P there is a preimage. Hence the result. □

Example 4.4. The configuration 10001101 of ECA rule 30 has two preimages, namely 11101000 and 00000101. In the preimage network, these correspond to the paths



and



respectively. Note that these are the only two paths in the graph from a vertex u_0 to the corresponding vertex u_8; the paths

begin and end at "different" vertices, and so do not correspond to preimages.

◊

The *de Bruijn matrix* for state s, denoted D_s, is the adjacency matrix (Definition A.57) of the de Bruijn graph for s.

Example 4.5. The de Bruijn matrix for a state s of an ECA is

$$D_s = \begin{pmatrix} \delta(000) & \delta(001) & 0 & 0 \\ 0 & 0 & \delta(010) & \delta(011) \\ \delta(100) & \delta(101) & 0 & 0 \\ 0 & 0 & \delta(110) & \delta(111) \end{pmatrix}, \tag{4.15}$$

where

$$\delta(abc) = \begin{cases} 1 & \text{if } f(a,b,c) = s \\ 0 & \text{otherwise.} \end{cases} \tag{4.16}$$

The de Bruijn matrices for ECA rule 30 are

$$D_0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \quad \text{and} \quad D_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{4.17}$$

◊

Theorem 4.6. *For a configuration c, the number of paths in the preimage network from vertex u_0 to vertex v_N is given by the entry in the matrix*

$$D_{c[0]} \times \cdots \times D_{c[N-1]} \tag{4.18}$$

in the row corresponding to u and the column corresponding to v.

This result is related to the following well-known result from graph theory: if A is the adjacency matrix of a (non-bipartite) graph, then the entry

in row u, column v of the matrix A^n is the number of paths of length n from vertex u to vertex v.

PROOF OF THEOREM 4.6. Proceed by induction on N. If $N = 1$, the matrix in question is simply the de Bruijn matrix $D_{c[0]}$, and the preimage network is simply the de Bruijn graph for $c[0]$. The u, v matrix entry is 1 if and only if there is an edge from u to v, otherwise it is 0. Also, there is a single path from u to v if and only if there is an edge from u to v, otherwise there are no paths. Thus the result holds in this case.

As an inductive hypothesis, suppose that the result holds for $N = M-1$. By definition of matrix multiplication, the u, v entry in $D_{c[0]} \times \cdots \times D_{c[M-1]}$ is $\mathbf{u} \cdot \mathbf{v}$, where $\mathbf{u}$ is the uth row in $D_{c[0]} \times \cdots \times D_{c[M-2]}$, $\mathbf{v}$ is the vth column in $D_{c[M-1]}$, and '$\cdot$' denotes the vector dot product.

By the inductive hypothesis, the wth entry in $\mathbf{u}$ is the number of paths from vertex u to vertex w in the corresponding preimage network. By definition of the de Bruijn matrix, the wth entry in $\mathbf{v}$ is 1 if there is an edge from w to v, or 0 otherwise. The product of these two entries is the number of paths from u to w if there is an edge from w to v, or 0 otherwise. In other words, recalling that the preimage network for $c[0] \dots c[M-1]$ is obtained by "appending" the de Bruijn graph for $c[M-1]$ to the preimage network for $c_0 \dots c[M-2]$, the product is the number of paths from u to v via w. The dot product $\mathbf{u} \cdot \mathbf{v}$ is the sum of these products for all w, and so is the total number of paths from u to v. Hence the result for $N = M$, and so for all N by induction. □

The *trace* of a matrix, denoted Tr, is the sum of the entries on the main diagonal. The following theorem is the main result of this section.

Theorem 4.7. *The number of preimages of a configuration c is given by*

$$\mathrm{Tr}\left(D_{c[0]} \times \cdots \times D_{c[N-1]}\right) . \tag{4.19}$$

PROOF. By Theorem 4.3, the number of preimages is $\sum_{u \in S^{2r}} P_u$, where P_u is the number of paths from u_0 to u_N. By Theorem 4.6, P_u is the entry in row u, column u of $D_{c[0]} \times \cdots \times D_{c[N-1]}$. Thus the number of preimages is the sum of these diagonal entries. □

Note that this result allows the number of preimages to be found in time $O(N)$. This is unusual in this thesis, and in the study of CAs in general: due to the exponential nature of the configuration space, it is far more common for time complexities to be exponential in N.

Example 4.8. The number of preimages for configuration 10001101 of ECA rule 30 is

$$\mathrm{Tr}(D_1 \times D_0 \times D_0 \times D_0 \times D_1 \times D_1 \times D_0 \times D_1) \tag{4.20}$$

$$= \mathrm{Tr} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{4.21}$$

$$= 0 + 1 + 1 + 0 \tag{4.22}$$

$$= 2\,. \tag{4.23}$$

Note that the off-diagonal nonzero entries in the matrix correspond to the two paths identified in Example 4.4 as *not* corresponding to preimages. ◊

CHAPTER 5

Other properties of cellular automata

This chapter reviews some miscellaneous properties of CAs. Section 5.1 describes *Langton's λ parameter*, a simple parameter on the CA's local rule table. Studying how the CA's Wolfram class changes with λ suggests that complex (class 4) behaviour exists at a phase transition between periodic (class 2) and chaotic (class 3) behaviour. Section 5.2 describes *Wuensche's Z parameter*, which measures the "determinism" of the reverse algorithm described in Section 4.1. Section 5.3 describes two entropy measures on space-time patterns of CAs, and suggests that these two measures together give an indication of a CA's Wolfram class.

The sections described above are somewhat empirical; the remainder of this chapter is more theoretical. Section 5.4 introduces the *common descendent problem* for CAs. This is the decision problem which effectively asks whether two configurations are in the same basin of attraction. In general, for CAs on infinite lattices, this problem is undecidable. Finally, Section 5.5 shows how ideas from dynamical systems theory, particularly the concept of *sensitive dependence on initial conditions*, can be applied to CAs.

5.1. Langton's λ parameter

5.1.1. Definition

Consider a CA with state set S. Choose an arbitrary state $q \in S$, and call it the *quiescent* state (although we do not require that it is quiescent in the sense of Section 2.6). Let n_q be the number of neighbourhood states mapped to state q by f:

$$n_q = \left|\{(x_1, \ldots, x_m) \in S^m \ : \ f(x_1, \ldots, x_m) = q\}\right|. \tag{5.1}$$

Then *Langton's λ parameter* [Lan90] is defined by

$$\lambda = \frac{|S|^m - n_q}{|S|^m}. \tag{5.2}$$

In other words, λ is the proportion of neighbourhood states which f maps to non-quiescent states (states other than q).

Langton's λ parameter is a measure of the homogeneity of the local update rule. If $\lambda = 0$, all neighbourhood states are mapped to q and so the

rule is as homogeneous as possible. Conversely, if $\lambda = 1 - \frac{1}{|S|}$ for each choice of q, then f maps to each state with equal frequency, and so the rule is as heterogeneous as possible.

5.1.2. Dynamics and λ

Langton [Lan90] studies the qualitative dynamics of 1-D CAs on the periodic lattice $\mathbb{Z}_N$, with four states and neighbourhood radius 2. The λ parameter is increased from $\lambda \approx 0$ to $\lambda \approx 0.75$, by starting with the zero rule and progressively changing entries in the rule table from q to other states.

In the following descriptions, the *length of transients* refers to the number of generations before the CA settles into its long-term behaviour, whatever that long-term behaviour may be.

- $0 \leq \lambda \leq 0.15$**:** The CA evolves to a homogeneous configuration. The length of transients ranges from a single generation ($\lambda \approx 0$) to five generations ($\lambda \approx 0.15$). See Figure 5.1 (a).
- $0.2 \leq \lambda \leq 0.3$**:** Periodic structures appear, having periods 1 or 2. The length of transients continues to grow with λ. See Figure 5.1 (b).
- $0.35 \leq \lambda \leq 0.4$**:** Periodic structures with larger periods (up to 40 for $\lambda \approx 0.4$) appear, and transients grow more rapidly (60 generations for $\lambda \approx 0.4$). See Figure 5.1 (c).
- $\lambda \approx 0.45$**:** Moving periodic structures (i.e. gliders in the sense of Section 2.7) appear. Their presence causes a sharp increase in the length of transients, since the slow-moving gliders must collide and annihilate each other before the CA settles into its long-term, periodic dynamics. See Figure 5.2 (a).
- $\lambda \approx 0.5$**:** The area of non-quiescent cells now tends to expand as well as contract, whereas for $\lambda < 0.5$ the tendency was to contract or remain stable. This fluctuation in the active area seems to be caused by a large number of gliders, whose collisions can result either in annihilation of the gliders, or creation of new gliders. The long-term behaviour is periodic, but the transients persist for around 12000 generations. See Figure 5.2 (b).
- $\lambda \approx 0.55$**:** Langton describes this as a *"new dynamical regime"*. The tendency is now to settle into chaotic, rather than periodic, behaviour. See Figure 5.2 (c).
- $0.6 \leq \lambda \leq 0.75$**:** The CA continues to exhibit chaotic behaviour in the long term, with the length of transients now decreasing as λ increases. See Figure 5.2 (d), (e).

In summary, this progression seems to show the following correlation between λ and Wolfram's classes:

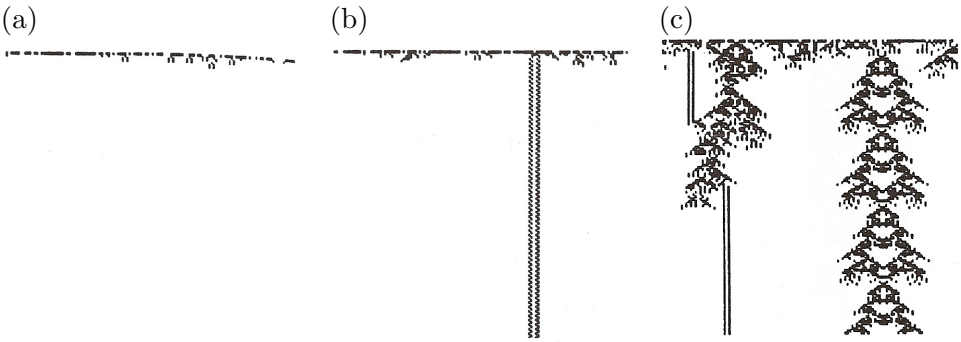


FIGURE 5.1. Typical space-time diagrams for CAs with $|S| = 4$, $r = 2$, and (a) $\lambda = 0.15$; (b) $\lambda = 0.2$; (c) $\lambda = 0.4$. [Lan90, Fig. 1]

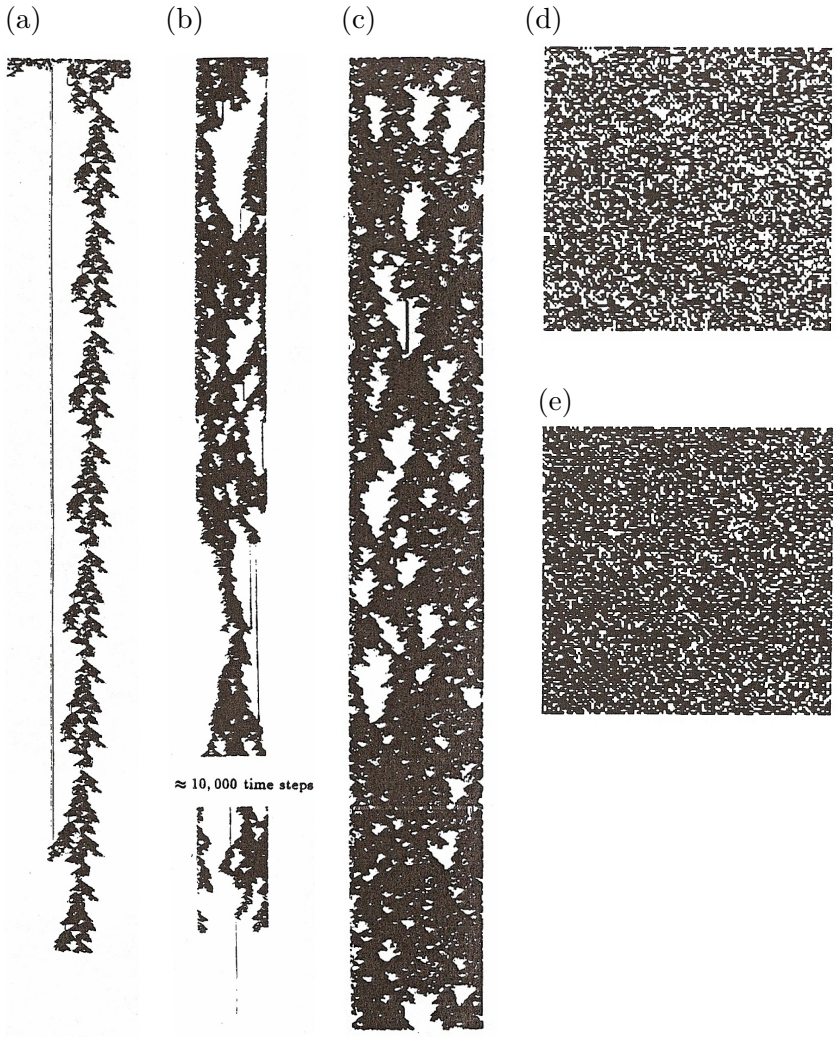


FIGURE 5.2. Typical space-time diagrams for CAs with $|S| = 4$, $r = 2$, and (a) $\lambda = 0.45$; (b) $\lambda = 0.5$; (c) $\lambda = 0.55$; (d) $\lambda = 0.65$; (e) $\lambda = 0.75$. [Lan90, Fig. 1]

Class 1: $0 \leq \lambda \leq 0.15$;
Class 2: $0.2 \leq \lambda \leq 0.4$;
Class 4: $0.45 \leq \lambda \leq 0.5$;
Class 3: $0.55 \leq \lambda \leq 0.75$.

The same progression, but with different values of λ, is seen for other CAs. This suggests that complex (class 4) behaviour occurs at the *phase transition* between ordered (class 2) and chaotic (class 3) dynamics. It is worth emphasising that the exact values of λ at which the phase transitions take place are different for different runs of this experiment, but the sequence of phase transitions (i.e. the sequence of Wolfram classes encountered as λ increases) is always the same.

Langton points out that λ is a good indicator of class for CAs with large neighbourhoods and many states, but not so good for CAs with small neighbourhoods and fewer states. In particular, for ECAs, λ is *"only roughly correlated with dynamical behavior"* [Lan90]. Indeed, Marr and Hütt [MH05] observe that examples of all four of Wolfram's classes can be found among the ECAs with $\lambda = 0.5$.

5.2. Wuensche's Z parameter

5.2.1. Definition

Wuensche's Z parameter [WL92] arises from Wuensche's reverse algorithm (see Section 4.1). The Z parameter is a measure of the degree of nondeterminism or parallelism in the algorithm. Suppose that we have a partial preimage for some configuration, and wish to choose a state with which to extend the preimage. There are three possible situations:

The *deterministic* case: Only one choice of state is valid. In Algorithm 4.1, this means the **if** statement of line 8 is true for exactly one iteration of the **for** loop of line 7.
The *ambiguous* case: There is more than one valid choice of state.
The *forbidden* case: There is no valid choice.

Note that the reverse algorithm as described in Section 4.1 works from left to right; it is possible to define a similar algorithm that works from right to left.

The Z parameter measures the probability of encountering the deterministic case, as opposed to the other two cases, at each step of the reverse algorithm. If Z_{left} is the probability of the deterministic case when computing preimages from left to right, and Z_{right} similarly when working from right to left, we take $Z = \max\{Z_{\text{left}}, Z_{\text{right}}\}$.

5.2.2. Calculating Z for ECAs

Let c be a configuration of an ECA, and suppose we have a partial preimage p of length $k < N$. What are the valid choices of state b such that pb is a partial preimage of length $k+1$? For the deterministic case, we must have exactly one choice of b such that

(1) $f(p[k-2], p[k-1], b) = c[k-1]$,
(2) $f(p[k-1], b, x) = c[k]$ for all x, or
(3) $f(b, x, y) = c[k+1]$ for all x, y.

The respective probabilities of these situations are:

(1) The proportion R_1 of pairs (α, β) which, for some x, give $f(\alpha, \beta, 0) = x$ and $f(\alpha, \beta, 1) \neq x$;
(2) The proportion R_2 of states α which, for some x and for all y, give $f(\alpha, 0, y) = x$ and $f(\alpha, 1, y) \neq x$;
(3) $R_3 = 1$ if, for some x and for all y, z, we have $f(0, y, z) = x$ and $f(1, y, z) \neq x$; $R_3 = 0$ otherwise.

Example 5.1. Consider ECA rule 184:

xyz	111	110	101	100	011	010	001	000
$f(x, y, z)$	0	0	0	1	0	0	1	1

(5.3)

The only pair satisfying condition (1) is $(1, 0)$, so $R_1 = \frac{1}{4}$.

State 0 satisfies condition (2), since $f(0, 0, 0) = f(0, 0, 1) = 1$ and $f(0, 1, 0) = f(0, 1, 1) = 0$. However, state 1 does not satisfy (2), since $f(1, 0, 0) \neq f(1, 0, 1)$. Thus $R_2 = \frac{1}{2}$.

Condition (3) is not satisfied, so $R_3 = 0$. ◊

Now the probability Z_{left} is the union of the independent probabilities R_1, R_2 and R_3:

$$Z_{\text{left}} = R_1 + (1 - R_1)R_2 + (1 - R_1)(1 - R_2)R_3 \,. \tag{5.4}$$

We calculate Z_{right} in an analogous fashion, effectively by reversing the order of f's parameters in the calculations of R_1, R_2 and R_3.

Example 5.2. Continuing Example 5.1, for rule 184 we have

$$Z_{\text{left}} = \frac{1}{4} + \left(1 - \frac{1}{4}\right)\frac{1}{2} + \left(1 - \frac{1}{4}\right)\left(1 - \frac{1}{2}\right)0 = \frac{5}{8} \,. \tag{5.5}$$

It turns out that $Z_{\text{right}} = \frac{5}{8}$ also, so we have

$$Z = \max\{Z_{\text{left}}, Z_{\text{right}}\} = \frac{5}{8} \,. \tag{5.6}$$

◊

Values of Z for the 88 essentially different ECAs are shown in Appendix B.

5.2.3. Dynamics and Z

The Z parameter is a measure of the "convergence" of the CA. Small Z suggests that there are configurations with a large number of preimages, which means that many trajectories converge on those configurations. This kind of convergence seems to suggest class 2, or class 1 in the extreme, so Wuensche [Wue02] suggests that Z varies from 0 to 1 as the dynamics of the CA vary from "order" to "chaos". This is in contrast to Langton's λ parameter, where the dynamics vary from "order" to "chaos" to "order" as λ varies from 0 to $1 - \frac{1}{|S|}$ to 1. However, as with Langton's λ, class 4 behaviour generally occurs at the transition between ordered and chaotic dynamics.

By a similar "convergence" argument, we can also consider Z as a measure of the "bushiness" of the transition graph (Definition 6.1). The in-degree of a vertex in the transition graph is precisely the number of preimages of the corresponding configuration, so small Z implies that there are vertices with large in-degree.

Note that those linear CAs meeting the criteria for the reverse algorithm described in Section 4.2 have $Z = 1$. As noted at the beginning of Section 4.2, $Z = 1$ does not imply surjectivity.

5.3. Word entropy and Shannon entropy

Marr and Hütt [MH05] propose a pair of entropy measures, which measure the statistical properties of a sampling of space-time patterns generated by a CA.

Definition 5.3 (Shannon entropy)**.** Denote by $p_i(x)$ the probability of state $x \in S$ appearing in the sequence of states for cell i. The *Shannon entropy* for cell i is

$$S_i = -\sum_{x \in S} p_i(x) \log_2 p_i(x). \tag{5.7}$$

The Shannon entropy S for the entire CA is an average over all cells.

In a class 1 CA, we expect $S = 0$: there is a state x such that, in the long term, we have $p_i(x) = 1$ for all cells i. We also expect $S = 0$ in a similar situation where the state x is allowed to be different for each cell i; this occurs for the identity rule, for instance, or more generally for any CA that eventually reaches a fixed point. Conversely, in a class 3 CA, we expect S to be maximal ($S = \log_2 |S|$) since all states should occur with equal probability.

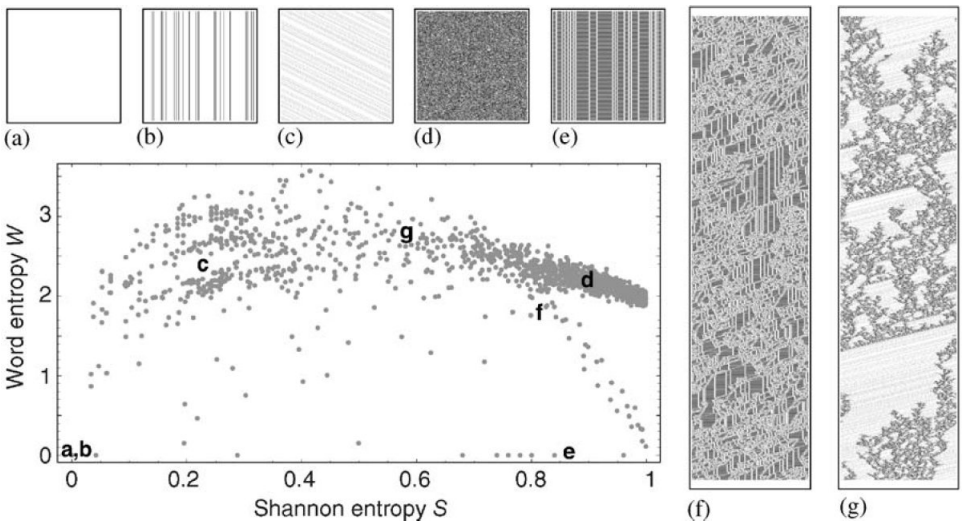


FIGURE 5.3. The WS plane. Values for several 1-D CAs (binary state set, neighbourhood size 11, lattice $\mathbb{Z}_{500}$) are plotted. Space-time diagrams for the particular points (a)–(g) are shown. [MH05, Fig. 1]

Definition 5.4 (Word entropy)**.** Let l be a positive integer, and denote by $p_i(l)$ the probability of cell i being constant for l consecutive time steps; that is, the probability that

$$c_t[i] = c_{t+1}[i] = \cdots = c_{t+l-1}[i] \tag{5.8}$$

for some t, where c_t denotes the configuration of the CA at time t. Then the *word entropy* for cell i over L time steps is

$$W_i = -\sum_{l=1}^{L} p_i(l) \log_2 p_i(l). \tag{5.9}$$

The word entropy W for the entire CA is an average over all cells.

We expect $W = 0$ in a class 1 CA, since all cells are eventually constant, giving $p_i(l) = 1$ for all l. Conversely, large W implies that $p_i(l)$ is around $\frac{1}{2}$ (or more precisely, $\frac{1}{e} \approx 0.37$), so constant and non-constant intervals are equally likely in i's sequence of states. This seems to imply the presence of structure across multiple scales, of the type we might expect from a class 4 CA.

A plot of W against S, along with space-time diagrams for some representative points within this plot, is shown in Figure 5.3. Marr and Hütt divide the WS plane according to Wolfram's classes as shown in Figure 5.4; this division is certainly plausible based on the points identified in Figure 5.3 and the intuitions for extreme values of W and S, but clearly it is intended only as an indication rather than a formal classification.

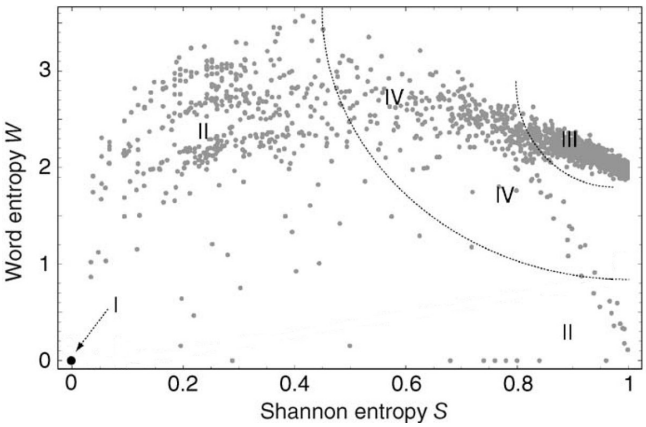


FIGURE 5.4. The regions of the WS plane which Marr and Hütt identify with Wolfram's classes. [MH05, Fig. 2]

5.4. The common descendent problem

The *common descendent problem* is the decision problem for the following question:

> Let c and d be two initial configurations for a given CA.
> Do t and t' exist such that $F^t(c) = F^{t'}(d)$?

In other words, do the trajectories starting from c and d have configurations in common?

Clearly the common descendent problem is decidable on finite lattices: there are only finitely many configurations, so enumeration suffices. Pedersen [Ped92] investigates this problem for CAs on the infinite 1-D lattice $\mathbb{Z}$ by way of the theory of *semigroups* (Definition A.3), and in particular the *word problem* on semigroups.

A semigroup S can be defined by a *presentation* $S = \langle X; R\rangle$, where:

(1) X is a set of *generators*; that is, a subset of S such that every element of S can be written as a product of elements of X.
(2) R is a set of *relations*; that is, equations of the form

$$x_1 * x_2 * \cdots * x_k = y_1 * y_2 * \cdots * y_l \tag{5.10}$$

where $x_i, y_j \in X$. The set R is a basis of equations in X; that is, any equation which holds in X can be derived from the equations in R.

The *word problem* for a semigroup $S = \langle X; R\rangle$ is the decision problem for the following question:

> Given $x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_l \in X$, does the equation

$$x_1 * x_2 * \cdots * x_k = y_1 * y_2 * \cdots * y_l \tag{5.11}$$

hold in S? Equivalently, can Equation 5.11 be derived from the equations in R?

Effectively, the word problem is a problem of formal language theory. We can think of the equations in R as productions in a grammar, and we are asking whether these productions can be applied to transform $x_1x_2\ldots x_k$ and $y_1y_2\ldots y_l$ into the same string.

Clearly the word problem is decidable if S is finite, so again, we are only interested in the infinite case.

Pedersen [Ped92] shows how to associate a semigroup with a given CA, in such a way that products of generators correspond to configurations, and repeated application of the relations corresponds to applying the global map to those configurations. The details of this construction are given in [Ped92], and reproduced in [Pow07].

This construction leads to the following result:

Theorem 5.5 ([Ped92, Theorem 1])**.** *The common descendent problem is decidable for a given CA if and only if the word problem is decidable for the corresponding semigroup.*

There exist semigroups for which the word problem is undecidable [Pos47], which suggests that there exist CAs for which the common descendent problem is undecidable. Indeed, the common descendent problem must be undecidable for Turing complete CAs, given that the halting problem is undecidable for a universal Turing machine. It is also easy to see that the common descendent problem is trivially decidable for class 1 CAs, since the homogeneous configuration is a descendent common to all configurations. Pedersen conjectures that the common descendent problem is decidable for class 2 CAs, but undecidable for class 3 and 4 CAs.

5.5. Equicontinuity and sensitive dependence

Sensitive dependence on initial conditions is an important concept in dynamical systems theory. Sensitive dependence describes the tendency for trajectories in the system which start off close together to diverge as time progresses. A popular example of this (e.g. [Gle88]) is the "butterfly effect", the idea being that the perturbation of the state of the planet's weather system caused by the movement of a butterfly's wing is sufficient to dramatically alter the subsequent behaviour of the system.

To apply the notion of sensitive dependence to cellular automata, we first need a measure of the "distance" between two configurations of a given CA. More specifically, we need to define a *metric* (Definition A.38) on the

configuration space. (Metrics on the configuration space are also discussed in Chapter 10.)

Consider a CA with state set S and infinite D-dimensional lattice $\mathbb{Z}^D$. Gamber [Gam06] defines a metric d on the set of configurations $S^{\mathbb{Z}^D}$ as follows. Let c and s be configurations, and define $d(c, s)$ by

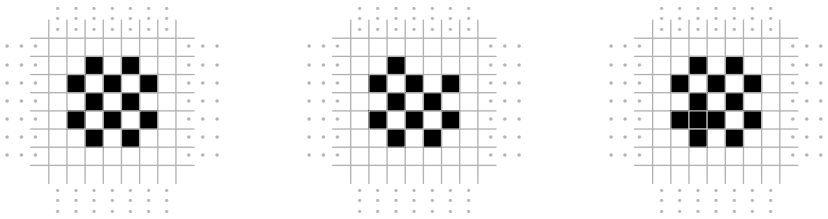
$$d(c,s) = \begin{cases} 2^{-k} & \text{if } k = \min\left\{\|i\| \ : \ i \in \mathbb{Z}^D, c[i] \neq s[i]\right\} \\ 0 & \text{if such a } k \text{ does not exist} \end{cases} \tag{5.12}$$

where $\|i\|$ denotes the maximum absolute value among i's components:

$$\|(i_1, i_2, \ldots, i_D)\| = \max\left\{|i_1|, |i_2|, \ldots, |i_D|\right\}. \tag{5.13}$$

Thus k measures the size of the largest hypercube, centred at the origin of the lattice, within which the two configurations are identical. Specifically, k is the perpendicular distance of each face of the hypercube from the origin, or half the side length. Intuitively, two configurations are close together according to d if the differences between the two configurations are a large distance from the origin. If k does not exist (i.e. if the configurations are completely identical) then k is effectively infinite.

Example 5.6. Consider a 2-D CA with lattice $\mathbb{Z}^2$ and state set $\{0, 1\}$, and consider the following three configurations, where all cells outside the depicted regions are in state 0:



c u v

Suppose that the origin is the central cell in each diagram. The cell at coordinates $(1, 2)$ is the only difference between configurations c and u, so $d(c, u) = 2^{-2} = \frac{1}{4}$. However, for c and v, cell $(-1, -1)$ is different, so $d(c, v) = 2^{-1} = \frac{1}{2}$.

If we instead take the origin to be the top right cell in each diagram, we get $d(s, q) = \frac{1}{4}$ and $d(s, r) = \frac{1}{16}$. ◊

Definition 5.7. A configuration c is an *equicontinuity point* if, for all configurations s close to c, the subsequent trajectories from s and c remain close

together for all time:

$$\begin{array}{l} \forall \varepsilon > 0, \\ \quad \exists \delta > 0 \text{ such that} \\ \quad\quad \forall s \in S^{\mathbb{L}} \text{ with } d(c,s) < \delta, \\ \quad\quad\quad d(F^t c, F^t s) < \varepsilon \text{ for all } t. \end{array} \tag{5.14}$$

We say that the CA is *equicontinuous* if every configuration is an equicontinuity point.

Definition 5.8. Let c be a configuration of a CA. We say that c is a *sensitive point* if there is a configuration s which is close to c, but where the subsequent trajectories from s and c eventually diverge:

$$\begin{array}{l} \exists \varepsilon > 0 \text{ such that} \\ \quad \forall \delta > 0, \\ \quad\quad \exists s \in S^{\mathbb{L}} \text{ with } d(c,s) < \delta \text{ such that} \\ \quad\quad\quad d(F^t c, F^t s) \geq \varepsilon \text{ for some } t. \end{array} \tag{5.15}$$

We say that the CA exhibits *sensitive dependence on initial conditions* if every configuration is a sensitive point.

In a sense, equicontinuity is the opposite of sensitive dependence. In an equicontinuous system, a perturbation which occurs a large distance away from the origin cannot possibly affect the origin; in a system with sensitive dependence, it is always possible to find such a perturbation which, no matter how far from the origin it occurs, will end up affecting the origin.

Gamber describes several other properties of CAs arising from the theory of metric spaces, and proves a number of theorems relating these properties to equicontinuity and sensitive dependence.

5.5.1. Periodicity

Definition 5.9. We say a CA with global update rule F is *eventually periodic* or *pre-periodic* if there exist integers k, p with $k \geq 0$, $p > 0$ and

$$F^{k+p} = F^k. \tag{5.16}$$

If $k = 0$, we say that the CA is *periodic*.

So if a CA is eventually periodic, then for every initial configuration c, we have $F^{k+p}(c) = F^k(c)$. Note that the same values of k and p must work for every initial configuration. However, the CA is eventually periodic even if different initial configurations have different periods and transient lengths, by taking k as the maximum transient length and p as the lowest common multiple of the periods.

Theorem 5.10 ([Gam06, Theorem 3.2])**.** *A CA is equicontinuous if and only if it is eventually periodic.*

SKETCH PROOF. The "only if" part is true because it is always possible to find a spatially periodic configuration close to any given configuration. The trajectory from a spatially periodic configuration is eventually periodic for all $i \in \mathbb{Z}^D$. Clearly the trajectory from s is periodic. Since equicontinuity forces the trajectories from c and s to remain close together, it also forces the trajectory from c to be eventually periodic, since it restricts that trajectory to a finite region of configuration space.

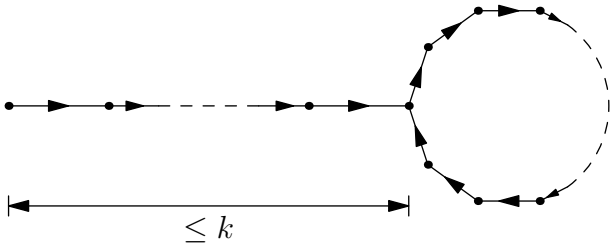
Let us now consider the "if" case. If the CA is eventually periodic, then there is a maximum distance over which information can propagate, given that there is an upper bound on the speed of propagation (Section 2.6 and information can only propagate for the first $k + p$ time steps. Thus any difference between configurations which is more than this distance from the origin cannot possibly affect the origin. Equicontinuity follows. □

Theorem 5.11 ([Gam06, Theorem 3.3])**.** *A CA is equicontinuous and surjective if and only if it is periodic.*

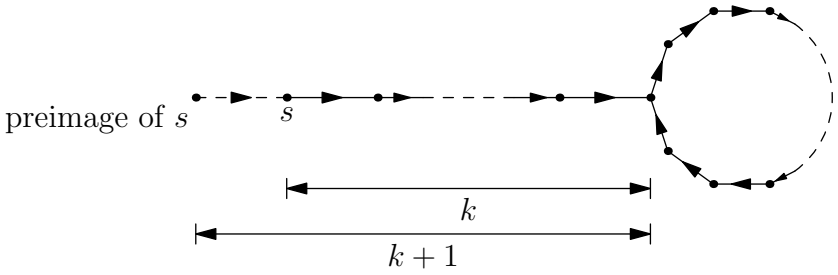
SKETCH PROOF. This follows almost immediately from Theorem 5.10. Periodicity clearly implies surjectivity: if a configuration is on a cycle then it certainly has a preimage, and all configurations of a periodic CA are on cycles.

For the converse, it suffices to show that surjectivity implies that an eventually periodic CA is in fact periodic. In fact, it is easier to see the contrapositive: if a CA is eventually periodic but not periodic, then it cannot be surjective.

Suppose that the CA is eventually periodic, with k and p as in Definition 5.9, and suppose further that $k > 0$. A trajectory in the CA consists of a "tail" of length no more than k, leading onto a cycle:



Furthermore, if k is minimal for this CA, there must be at least one such tail of length k exactly. But the CA is surjective, so the configuration s at the end of the tail must have a preimage.



Thus there is a tail of length $k+1$, so we have obtained a contradiction. Therefore an eventually periodic CA with $k > 0$ cannot be surjective. □

5.5.2. Almost equicontinuity

A CA is *almost equicontinuous* if the set of equicontinuity points contains an intersection of dense open sets (Definitions A.44 and A.48). An intersection of dense open sets is itself dense. So if a CA is almost equicontinuous, then the set of equicontinuity points is dense. This means that, for any configuration c and any positive integer k, we can find a configuration e, with e an equicontinuity point, such that c and e are identical within a hypercube of side length k.

Gamber [Gam06, Theorem 6.3] shows that a CA is almost equicontinuous if it has a *fully blocking pattern* of a sufficient size. A fully blocking pattern is a pattern of states fixed under F when it appears in a configuration. Thus a fully blocking pattern acts as a barrier to information propagation in the CA.

5.5.3. Summary

Figure 5.5 summarises the relationships between the properties described in this section, and shows how they partition the rule space.

Equicontinuity and sensitivity seem to be characteristic of Wolfram's classes 2 and 3 respectively. By analogy with the discussion of Langton's λ parameter in Section 5.1, we might expect class 4 behaviour to appear on the boundary between equicontinuity and sensitivity.

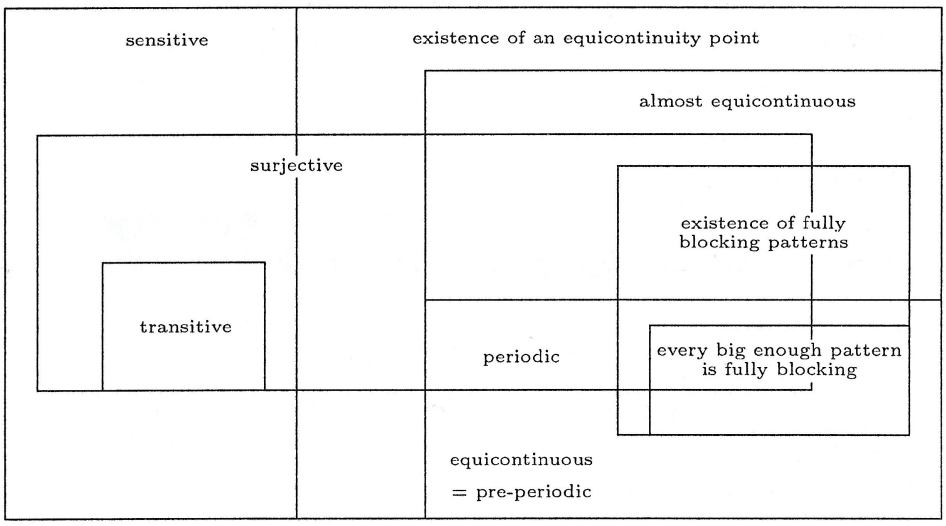


FIGURE 5.5. How the properties described in Section 5.5 partition the rule space. [Gam06, Figure 16]

Part 2

New results

CHAPTER 6

Transition graphs

While we can easily depict the evolution of a CA from a particular initial configuration, e.g. by drawing a space-time diagram, it is sometimes useful to get a picture of the overall dynamics of a CA from all (or at least a large number of) initial configurations. One way of doing this is by drawing a directed graph (Definition A.54) called a *transition graph*.

Definition 6.1. Consider a CA with state set S, lattice $\mathbb{L}$ and global update rule F. The *transition graph* for this CA is $\mathcal{G} = (S^{\mathbb{L}}, \mathcal{E})$, with

$$\mathcal{E} = \left\{ (s, F(s)) \ : \ s \in S^{\mathbb{L}} \right\} \tag{6.1}$$

That is, the vertices represent the configurations, and the edges represent transitions from configuration to configuration: there is an edge from vertex r to vertex s if and only if $F(r) = s$, i.e. if and only if the CA evolves from configuration r to configuration s in one time step.

The transition graph is the *functional graph* [Har69] for the function F. Functional graphs have the property that every vertex has out-degree 1, and such a graph always has a "circles of trees" topology: one or more disjoint cycles, with trees rooted at the vertices in the cycles; edges within the trees are directed towards the root. Another name for this type of graph is a *pseudoforest*, with each connected component being a *pseudotree* [GT88].

Examples of transition graphs are shown in Figures 6.1 to 6.3. Transition graphs for the 88 essentially different ECAs are shown in Appendices C and D, for the finite lattices $\mathbb{Z}_{10}$ and $\mathbb{Z}_{11}$ respectively.

These graphs are most closely associated with Wuensche [WL92, Wue97, Wue02], but were used earlier by Wolfram et al [MOW84]. Some authors refer to transition graphs as *state transition diagrams*; those authors use the word "state" to mean the global configuration, and not the local state of a cell as in our terminology.

In dynamical systems terms, paths in the graph are trajectories. The cycles in the graph are *attractors*, and each connected component of the graph (consisting of one cycle and the trees rooted at its vertices) is a *basin of attraction*. Thus we refer to the connected components of the transition graph as its *basins*.

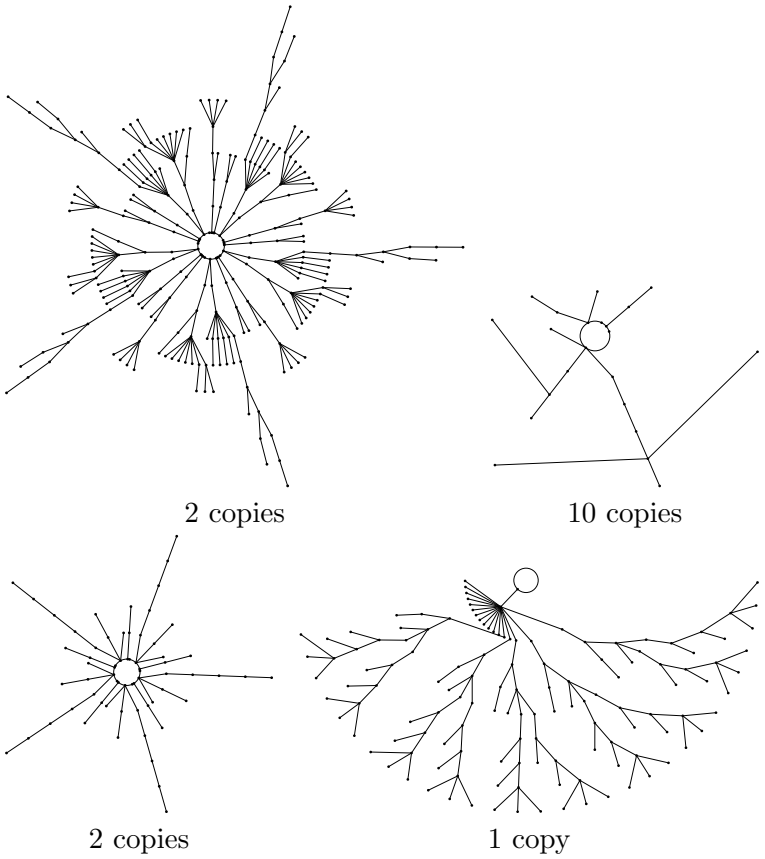


FIGURE 6.1. Transition graph for ECA rule 110 on the periodic lattice $\mathbb{Z}_{10}$. Edges in trees are directed towards the cycle; edges in cycles are directed clockwise. Numbers of "copies" refer to the numbers of connected components (basins) in the transition graph which are isomorphic to the graph shown.

This chapter reviews some results from the literature on transition graphs, and gives some new algorithms and results. Section 6.1 describes how to find the basin of attraction containing a given configuration. Section 6.2 gives an algorithm for testing whether trees are isomorphic, which is used frequently in the following section (Section 6.3) on drawing transition graphs. Here "drawing" refers both to finding the vertices and edges of the graph, and to laying the graph out on the page. Finally, Section 6.4 gives some properties of transition graphs for linear CAs. Chapters 7 and 8 study *automorphisms* (symmetries or self-isomorphisms) of transition graphs.

6.1. Finding the attractor

In which basin of attraction is a given configuration?

First, we need some way of identifying the basin of attraction. One obvious choice is to identify it with the configuration which comes first in

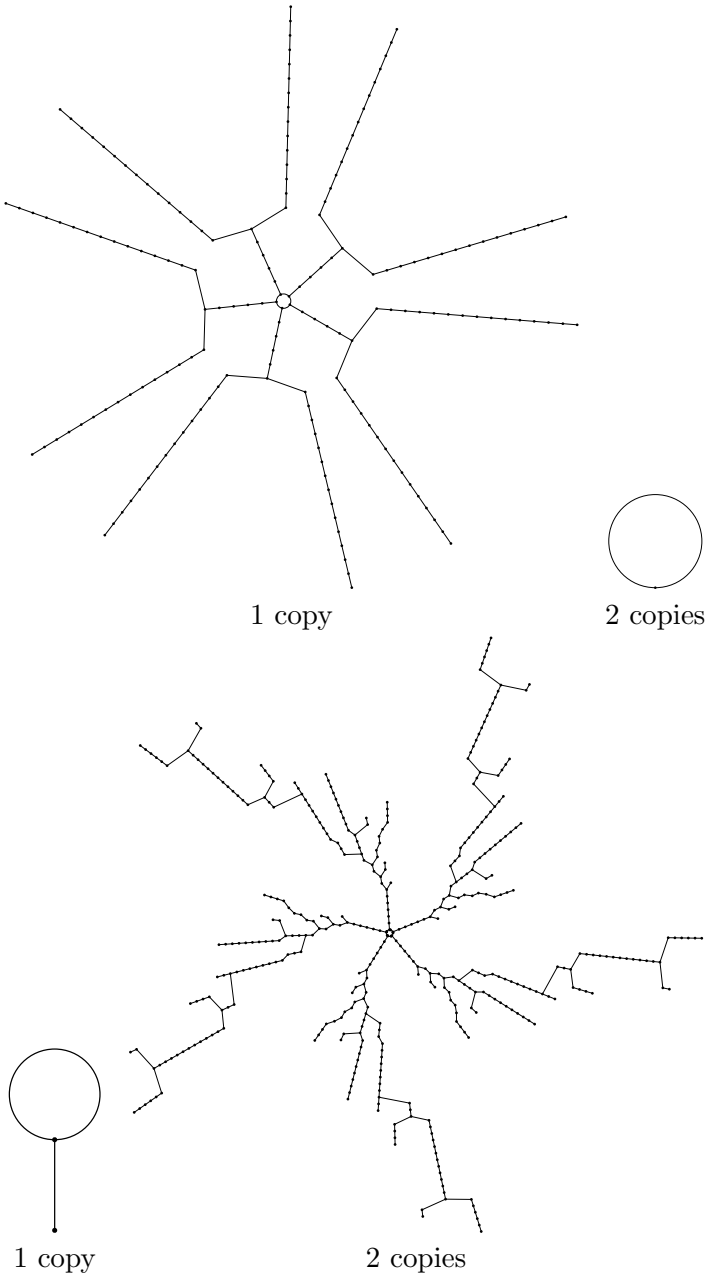


FIGURE 6.2. Transition graph for ECA rule 30 on the periodic lattice $\mathbb{Z}_{10}$.

lexicographical order amongst all configurations on the attractor cycle. This configuration can easily be found once we have found some configuration on the cycle, by circumnavigating the cycle and keeping a record of the lexicographically first configuration seen so far. Thus the problem of finding configuration c's basin of attraction reduces to that of finding any configuration on the attractor cycle reached from c.

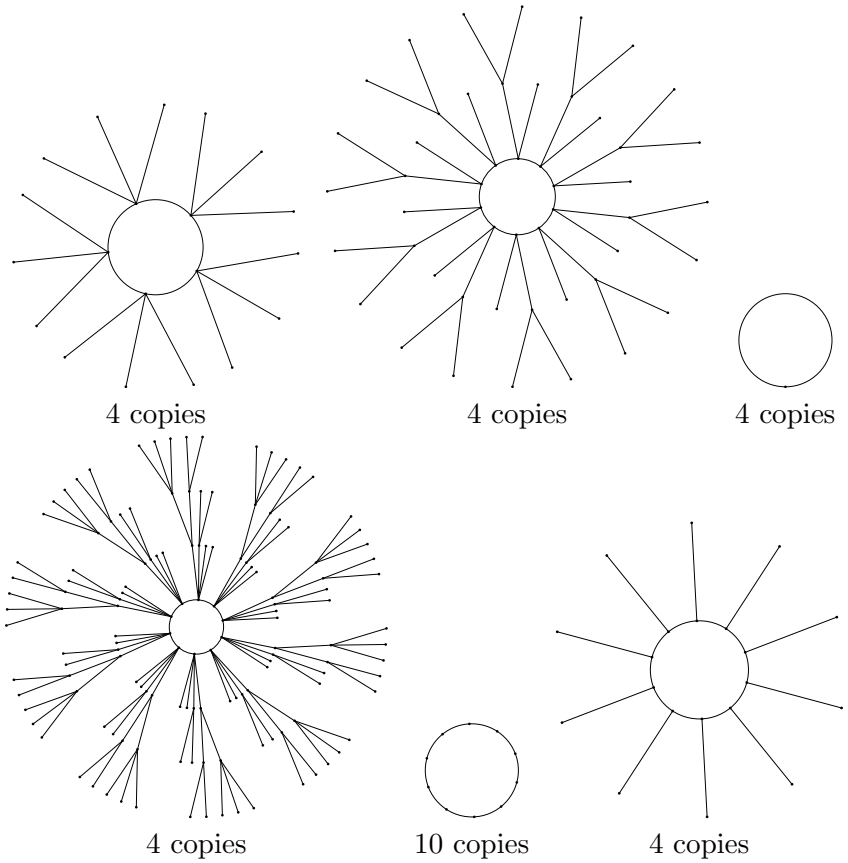


FIGURE 6.3. Transition graph for ECA rule 142 on the periodic lattice $\mathbb{Z}_{10}$.

The naïve way of finding the attractor cycle is as follows: repeatedly apply the global map F to s, remembering all visited configurations. When we reach a configuration visited before, that configuration must be on the attractor cycle. If the length of the transient is l_t and the length of the cycle is l_c, this algorithm has complexity $O(l_t + l_c)$ in both space and time. It seems likely that this is the best we can do in terms of time complexity, but we can do much better in terms of space complexity.

The *tortoise and hare algorithm* (or more prosaically, *Floyd's cycle finding algorithm*) is often used to detect cycles in linked list data structures. Floyd [Flo67] gives a nondeterministic version of the algorithm for finding cycles in graphs; Knuth [Knu69, page 7, exercise 6(b)] attributes the deterministic algorithm to Floyd, but without citation. See also [Ost08]. For finding cycles in functional graphs (of which, incidentally, the singly linked list is an example), the deterministic version suffices.

The tortoise and hare algorithm proceeds as follows. Let c be a configuration, and define two sequences of configurations by

$$T_0 = c \tag{6.2}$$

$$T_{i+1} = F(T_i) \tag{6.3}$$

$$H_0 = F(c) \tag{6.4}$$

$$H_{i+1} = F^2(H_i)\,. \tag{6.5}$$

If $T_i = H_i$ for some i, then the configuration T_i is on the attractor cycle. Intuitively, we have two agents traversing the graph: the "tortoise" T and the "hare" H. The hare is given a head-start of one step, and moves two steps per unit time, while the tortoise moves only one step per unit time. After $\lceil (l_t - 1)/2 \rceil$ time units, the hare reaches the attractor cycle, and proceeds to circumnavigate it. The tortoise, however, does not reach the cycle until l_t time steps have passed. The two agents may not meet immediately, but the tortoise is a steps "ahead" of the hare, with $0 \leq a < l_c$. The hare closes this lead by one step per unit time, and so the two agents meet after a time steps.

The time complexity of the tortoise and hare algorithm is $O(l_t + l_c)$, as it was for the naïve algorithm (although the leading constant is larger for the tortoise and hare algorithm, as it requires three times as many evaluations of F per time step). However, there is no need to keep a record of visited configurations, so the tortoise and hare algorithm's space complexity is constant. Given that l_t and l_c can grow exponentially with respect to the number of cells, this is a significant improvement on the naïve algorithm.

Whichever algorithm is used to find the cycle, the representative configuration for the basin can be found in time $O(l_c)$ and constant space. This extra step does not affect the asymptotic complexity of either algorithm.

Note that, unlike the naïve algorithm, the tortoise and hare algorithm does not in general find the point where the transient meets the attractor. However, since we need only to identify some configuration on the cycle, this does not actually make any difference.

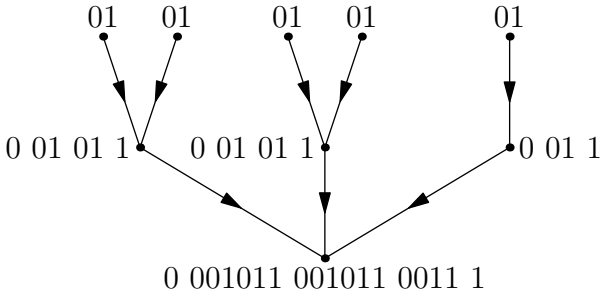
6.2. Testing for isomorphisms

Campbell and Radford [CR91] give an algorithm for testing whether two trees are isomorphic. The idea is to construct a string which uniquely describes the "shape" of the tree, so that two trees have the same string if and only if they are isomorphic. This is done recursively as follows:

(1) A single vertex tree has string 01.

(2) Consider a tree rooted at vertex r, where the subtrees rooted at r's children have strings $s_1, \ldots, s_k$. Permute the children of r so that $s_1, \ldots, s_k$ are in lexicographical order. Then the tree rooted at r has string $0s_1 \ldots s_k 1$.

Example 6.2. In the following tree, each vertex is labelled with the string representing the subtree rooted there:



The whole tree has string 000101100101100111. ◊

Theorem 6.3. *Two trees have the same string representation if and only if they are isomorphic.*

Proof. Proceed by induction on the depth of the tree. A tree of depth 0 consists of a single vertex; all such trees are isomorphic and have string 01, so the result is true in this case.

Suppose, as an inductive hypothesis, that the result holds for trees of depth less than d. Let T be a tree of depth d, whose root vertex is r, where the children of r are $c_1, \ldots, c_k$. Similarly let T' be another tree of depth d, whose root r' has children $c'_1, \ldots, c'_{k'}$. We wish to show that T and T' are isomorphic if and only if they have the same string representation.

Note that T and T' are isomorphic if and only if there is a bijection

$$\alpha : \{c_1, \ldots, c_k\} \to \{c'_1, \ldots, c'_{k'}\} \tag{6.6}$$

such that the subtree of T rooted at c_j is isomorphic to the subtree of T' rooted at $\alpha(c_j)$, for all j. (Such a bijection clearly does not exist if $k \neq k'$, but the following argument still applies, since T and T' are clearly not isomorphic in this case.) The subtrees in question all have depth less than d, so by the inductive hypothesis, T and T' are isomorphic if and only if there is a bijection α such that the subtrees rooted at c_j and $\alpha(c_j)$ have the same string representation.

Let s_i be the string representation of the tree rooted at c_i, and similarly for s'_j and c'_j. Then the bijection α described above exists if and only if the multisets $\{s_1, \ldots, s_k\}$ and $\{s'_1, \ldots, s'_{k'}\}$ are equal. (Again, this is clearly false if $k \neq k'$.) But the string representation of T is obtained by concatenating the elements of $\{s_1, \ldots, s_k\}$ in lexicographical order, prepending 0 and

appending 1, and similarly for T'. Thus these multisets are equal if and only if the string representations of T and T' are equal. Hence the result. □

If we only want to test whether two trees are isomorphic, then their strings can be condensed as the algorithm proceeds. This is the approach taken by Campbell and Radford [CR91]: as h ranges from the depth of the trees down to zero, the strings for the vertices at level h of both trees are found, and the two lists of strings are sorted and compared. If the lists are different then the trees are not isomorphic; if they are the same, each vertex is assigned a new, shorter string, which encodes the position in the sorted list of its old string.

Condensing the strings is beneficial for a single isomorphism test: shorter strings require less storage space, and less time to test for equality. However, the condensed strings are only meaningful in the context of a particular instance of the algorithm. For our purposes, there are two advantages to leaving the strings in their uncondensed form. First, the strings can be cached between isomorphism tests, so that the string for each tree need only be computed once even if that tree is tested for isomorphism against many other trees. Second, the tree (without vertex labels) can be reconstructed from the uncondensed string, so the string can be used as a compact representation of the tree.

Some properties of the tree can be extracted directly from the uncondensed string, without reconstructing the tree. (These results are outside the scope of [CR91], and we are not aware of them appearing elsewhere in the literature.) For example:

Theorem 6.4. *The number of 0s at the beginning of the string representation of a tree is one greater than the depth of that tree.*

Corollary 6.5. *If two trees T_1, T_2 have depths d_1, d_2 respectively with $d_1 > d_2$, then the string representing T_1 lexicographically precedes the string representing T_2. In other words, lexicographical ordering of strings orders trees in descending order of depth.*

Proof of Theorem 6.4. Proceed by induction on the depth of the tree. The only tree of depth 0 is the single vertex tree, which has string 01. The result holds in this case.

Consider a tree T of depth $d > 0$ rooted at vertex r, and suppose, as an inductive hypothesis, that the result holds for all trees of depth less than d. The subtrees rooted at r's children have depth less than d, and at least one of these subtrees has depth $d - 1$.

Note that Corollary 6.5 holds if Theorem 6.4 holds for trees of depths d_1 and d_2. Thus the inductive hypothesis implies Corollary 6.5 with $d_1, d_2 < d$.

Suppose that the strings representing the subtrees, arranged in lexicographical order, are $s_1, \ldots, s_k$. By Corollary 6.5, s_1 must correspond to one of the subtrees of depth $d-1$. Thus, by the inductive hypothesis, s_1 begins with d 0s, and thus the string representing T (i.e. $0s_1 \ldots s_k1$) begins with $d+1$ 0s. Hence the result. □

Theorem 6.6. *The number of 0s (or equivalently 1s) in the string representation of a tree is equal to the number of vertices in the tree.*

PROOF. Follows immediately from the definition. □

Theorem 6.7. *The number of occurrences of the substring* 01 *in the string representation of a tree is equal to the number of leaf vertices in the tree.*

PROOF. Follows immediately from the definition. □

6.3. Drawing transition graphs

Functional graphs, and thus transition graphs, are *planar*, meaning that it is always possible to draw them in such a way that no edges intersect. This is obvious, given that trees are planar and functional graphs are "circles of trees".

Several software packages exist for drawing graphs, one of the best known being the open source Graphviz package [GN00]. Graphviz consists of several command-line programs, each of which takes as input a textual description of a graph (a list of vertices and edges) and produces as output an image of the graph in one of several graphics file formats, or alternatively a list of vertex coordinates suitable for further processing.

Two programs from the Graphviz package are used here, namely `neato` and `dot`. The `neato` program uses a physics-inspired algorithm due to Kamada and Kawai [KK89], in which the vertices of the graph are modelled as particles connected by springs. The `dot` program is intended for drawing trees, and works by determining the rank (distance from the root) of each vertex, assigning y coordinates to vertices based on their rank, and choosing x coordinates to eliminate edge intersections and minimise edge lengths.

Perhaps the simplest way to draw a transition graph is to feed a specification of the entire graph into `neato`. This specification can easily be generated, by iterating through all configurations of the CA, and outputting an edge $c \to F(c)$ for each configuration c. A transition graph produced in this way is shown in Figure 6.4. This approach is not ideal: it often fails to eliminate all edge intersections, and there is no guarantee that isomorphic trees or basins will be drawn in an identical way.

Rather than drawing the entire transition graph at once, it is preferable to draw each basin individually. The transition graph for an individual basin

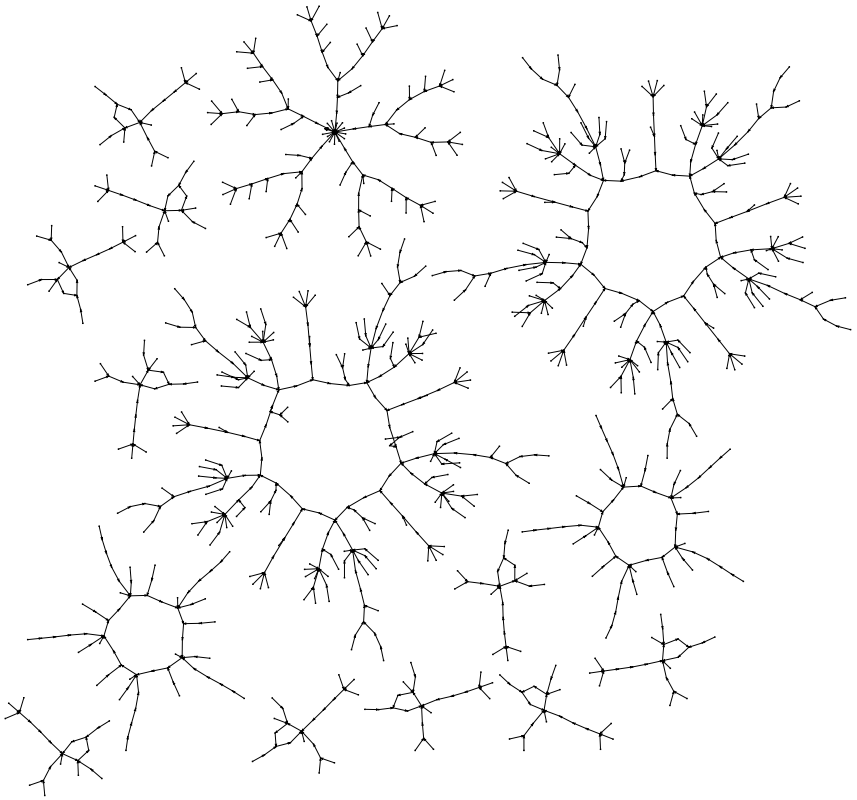


FIGURE 6.4. The transition graph for ECA rule 110 on the lattice $\mathbb{Z}_{10}$, drawn using `neato`.

can be constructed by using the tortoise and hare algorithm (Section 6.1) to find the vertices on the cycle, and iteratively using the reverse algorithm (Section 4.1) to construct the trees from root to leaves.

Wuensche [WL92] draws the basins by laying out each tree individually and arranging them in a circle. However, Wuensche uses a rather simplistic tree layout algorithm that allows edges to intersect; see Figure 6.5.

The transition graphs in this thesis are drawn by a similar method to Wuensche's, but using the rather more sophisticated tree layout algorithm of `dot`. The algorithm is outlined in Algorithm 6.1, but the basic idea is to use `dot` to obtain vertex coordinates for each of the trees rooted on the cycle, and transform these coordinates to arrange the trees in a circle. The layout for each tree is cached and reused for isomorphic copies of that tree; Section 6.2 gives an algorithm for determining when trees are isomorphic. In fact, the implementation of this algorithm uses the strings described in Section 6.2 as its internal representation of trees, so isomorphism is tested by string comparison.

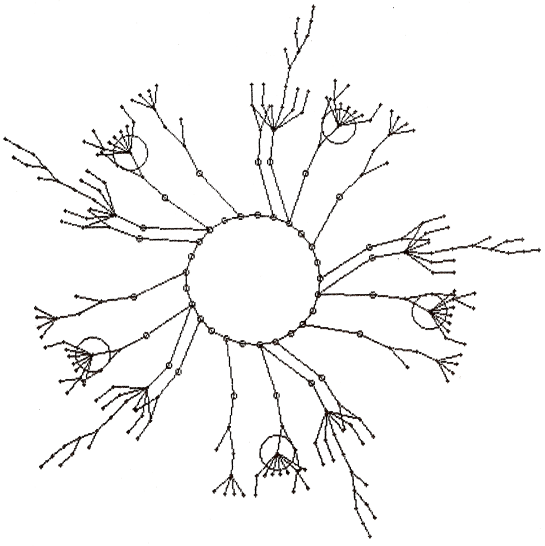


FIGURE 6.5. A basin from the transition graph for ECA rule 110 on the lattice $\mathbb{Z}_{10}$. Reproduced from [WL92].

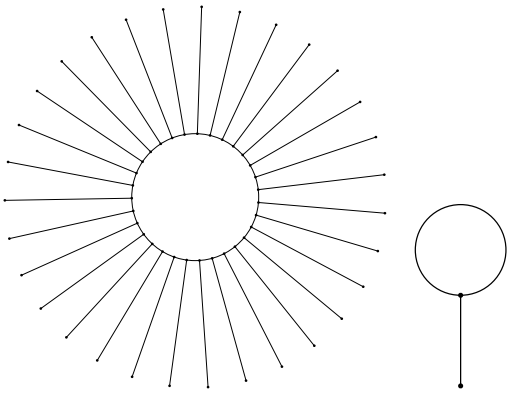


FIGURE 6.6. Transition graph for ECA 90 on the periodic lattice $\mathbb{Z}_{11}$.

Examples of transition graphs drawn using this algorithm are shown in Figures 6.1, 6.2, 6.3, 6.6 and 6.7, with many more examples in Appendices C and D.

6.4. Transition graphs for linear CAs

Martin et al [MOW84] show that linearity (see Chapter 3) imposes several restrictions on the transition graph. Figures 6.6 and 6.7 show transition graphs for a linear CA.

Lemma 6.8 ([MOW84, Lemma 3.3])**.** *In the transition graph for a linear CA, all trees rooted at vertices in cycles are isomorphic.*

```
 1: procedure DrawGraph
 2:     let T be the set of trees rooted on cycles
 3:     for t ∈ T do
 4:         process t through dot, obtaining coordinates for t's vertices
 5:     let B be the set of basins
 6:     for b ∈ B do
 7:         w ← 0
 8:         for each tree t rooted on b's cycle do
 9:             determine width(t) from the dot output of Line 4
10:             w ← w + width(t)
11:         δ ← 2π/w
12:         θ ← 0
13:         for each tree t rooted on b's cycle do
14:             for each edge u → v in t do
15:                 p_u ← coordinates of u from the dot output of Line 4
16:                 p_v ← coordinates of v from the dot output of Line 4
17:                 draw a line from Polarise(p_u) to Polarise(p_v)
18:             θ ← θ + δ × width(t)
19: function Polarise((x, y))
20:     φ ← θ + δ × x
21:     ρ ← y
22:     return (ρ cos φ, ρ sin φ)
```

Algorithm 6.1: The layout algorithm for transition graphs.

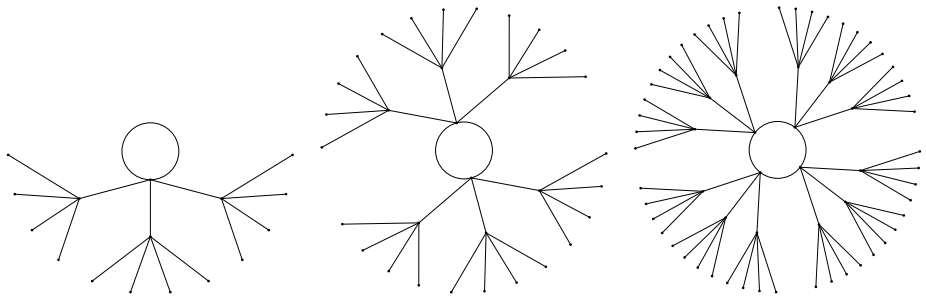


FIGURE 6.7. Transition graph for ECA 90 on the periodic lattice $\mathbb{Z}_{12}$.

Note that this result applies to *all* trees in a given transition graph, whether rooted at vertices in the same cycle or in different cycles. An immediate consequence of this result is that two basins are isomorphic if and only if their cycles have the same length.

Now that we know the trees are all isomorphic, the following result gives us some insight into their structure. This result is not explicitly stated by Martin et al, but is an obvious corollary to stated results.

Lemma 6.9. *All reachable configurations of a linear CA have the same number of preimages.*

PROOF. The zero configuration is always reachable (since $F(0) = 0$ for a linear CA), so it suffices to show that any reachable configuration has the same number of preimages as the zero configuration.

By additive superposition (Theorem 3.2), for all configurations u, v we have $F(u) = F(v)$ if and only if $u = v+q$ for some preimage q of the zero configuration [MOW84, Lemma 3.2]. Let w be a reachable configuration, and choose a preimage v of w. Then the preimages of w are the configurations

$$\{u \ : \ F(u) = F(v) = w\} = \{v + q \ : \ F(q) = 0\} \ . \tag{6.7}$$

Note that $v + q = v + q'$ implies $q = q'$, so each value of q yields a unique element of this set. In other words, the preimages of w are in one-one correspondence with the possible values of q, which themselves are the preimages of the zero configuration. □

Suppose that the zero configuration has p preimages. If we restrict our attention to a tree in the transition graph, we see that all non-root non-leaf vertices have in-degree p. However, one of the preimages of the root vertex is the preceding vertex in its cycle, so within the tree the root vertex has in-degree $p - 1$. This can be seen in Figure 6.7, where $p = 4$.

Some other properties of transition graphs for linear CAs, particularly for ECA rule 90, are given in Section 3.4.

CHAPTER 7

Counting automorphisms of transition graphs

A high degree of symmetry is apparent in the transition graphs shown in Appendices C and D. Many of the individual basins are isomorphic to each other, many basins within themselves exhibit rotational symmetry, and trees tend to contain subtrees which are isomorphic to each other. These symmetries are all examples of *automorphisms* of the transition graph.

This chapter investigates automorphisms of transition graphs. Section 7.1 gives a formal definition of automorphisms. Section 7.2 introduces a physics-inspired notion of *symmetries* of a CA, and shows (in Corollary 7.9) that the symmetries of the CA are precisely the automorphisms of the transition graph.

Some transition graphs have more symmetry than others. For example, referring back to the examples of transition graphs in Chapter 6, class 3 rule 30 (Figure 6.2) appears less symmetric than class 4 rule 110 (Figure 6.1), which in turn appears less symmetric than class 2 rule 142 (Figure 6.3).

One measure of the degree of symmetry of a graph is its number of automorphisms: the more automorphisms, the more symmetry. In this chapter, we investigate the following question: is the order of the automorphism group of the transition graph correlated with the Wolfram class of the CA, with class 1 and class 2 CAs having the most automorphisms, class 3 CAs having the least, and class 4 CAs having an intermediate number?

To investigate this question, we first need a method for computing the number of automorphisms. Section 7.3 presents such a method, and Section 7.4 uses it to compute numbers of automorphisms for the 88 essentially different ECAs. Considering the number of automorphisms as a function of the number of cells N partitions the space of ECA rules into three classes, differentiating the non-trivial linear rules and the rules with chaotic behaviour or long transients from the remainder.

The method of Section 7.3 gives a recursive expression for the number of automorphisms. This expression can be split into three distinct parts, corresponding to the three types of symmetry (isomorphic basins, rotational symmetry, isomorphic trees) mentioned at the beginning of this chapter. Section 7.5 briefly investigates these three parts, and shows that they all

have a roughly equal contribution to the total number of automorphisms (or at least that one of them does not dominate the others).

7.1. Automorphisms

Definition 7.1. An *isomorphism* from graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ is a bijection $\alpha : \mathcal{V} \to \mathcal{V}'$ such that

$$(x, y) \in \mathcal{E} \iff (\alpha(x), \alpha(y)) \in \mathcal{E}' . \tag{7.1}$$

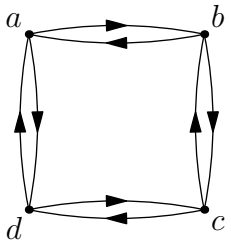
An *automorphism* on graph $\mathcal{G}$ is an isomorphism from $\mathcal{G}$ to $\mathcal{G}$.

In other words, an isomorphism is a bijection from vertices of $\mathcal{G}$ to vertices of $\mathcal{G}'$ that preserves edges: there is an edge between two vertices in $\mathcal{G}$ if and only if there is an edge between the two corresponding vertices in $\mathcal{G}'$. An automorphism is a permutation of the vertex set that preserves edges.

Theorem 7.2. *For a given graph, the set of all automorphisms forms a group under composition of functions.*

Proof. Each of the group axioms follows almost immediately from the definition. □

It is reasonable to talk about the group of automorphisms for a graph as being the group of symmetries for that graph. For example, the automorphisms of the graph



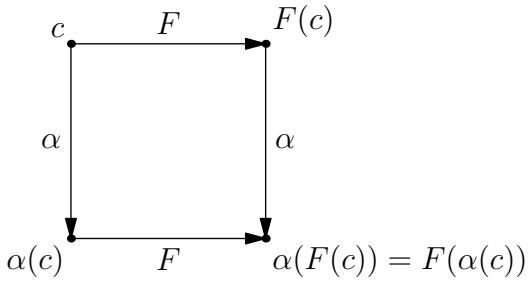
are in one-to-one correspondence with the symmetries of the square (see Example A.12).

7.2. Symmetries

Consider Conway's Game of Life (Section 2.2), evolving from a given initial configuration. If we rotate the initial configuration through 90°, the entire evolution will be rotated through 90°, but will otherwise be unchanged. Likewise, if we reflect the initial configuration in a vertical axis, the evolution will be reflected but otherwise identical. However, if we were to swap the values of 0 and 1 in the initial configuration, or apply a shear transformation, the evolution would be changed completely for almost all initial configurations.

It seems natural to identify transformations such as the rotation and reflection identified above as *symmetries* of the CA.

Definition 7.3. Consider a CA with global map F. The *symmetries* of the CA are the bijections on the configuration space which commute with F, i.e. the bijections α with $F \circ \alpha = \alpha \circ F$. This is illustrated in the following commutative diagram:



So, if the sequence of configurations resulting from initial configuration s_0 is

$$s_0, s_1, s_2, \dots , \tag{7.2}$$

and if α is a symmetry of the CA, then the sequence of configurations resulting from initial configuration $\alpha(s_0)$ is

$$\alpha(s_0), \alpha(s_1), \alpha(s_2), \dots \tag{7.3}$$

Symmetries can equivalently be defined in terms of *topological conjugations*:

Definition 7.4. Consider two functions $F, G : X \to X$, where X is a discrete topological space (Definition A.53). A bijection $\alpha : X \to X$ is a *topological conjugation* from F to G if $G \circ \alpha = \alpha \circ F$.

The symmetries of a CA are thus the topological conjugations from the global map to itself. The definition of a *discrete topological space* is not particularly important for our purposes; suffice it to say, the configuration space of a CA is always a discrete topological space.

There is a class of symmetries possessed by every CA:

Definition 7.5. Consider a CA with lattice $\mathbb{L}$, and choose $v \in \mathbb{L}$. Then the transformation σ_v with

$$\sigma_v(c)[i] = c[i + v] \text{ for all } i \in \mathbb{L} \tag{7.4}$$

is a symmetry of the CA. We call σ_v a *shift symmetry*, because it corresponds to a shift (or translation) of the configuration by a fixed offset v.

Note that the set of symmetries $\{\sigma_v \ : \ v \in \mathbb{L}\}$ forms a group. The fact that the shifts are symmetries for every CA is not surprising, given that spatial homogeneity is inherent in the definition of a CA.

There are two other common (although not ubiquitous) symmetries for 1-D CAs:

Definition 7.6. Consider a CA with 1-D lattice $\mathbb{L}$ (so $\mathbb{L} = \mathbb{Z}$ or $\mathbb{Z}_N$), and choose $v \in \mathbb{L}$. If the transformation ρ_v with

$$\rho_v(c)[i] = c[v - i] \text{ for all } i \in \mathbb{L} \tag{7.5}$$

is a symmetry, we call it a *reflection symmetry.*

The symmetry ρ_v corresponds to a reflection about the point $\frac{v}{2}$. Note that any reflection can be obtained from any other by appropriate composition with shifts, so it suffices to take a single reflection symmetry about a convenient point.

Definition 7.7. Consider a CA with state set $\{0, 1\}$. If the transformation C with

$$C(c)[i] = 1 - c[i] \text{ for all } i \in \mathbb{L}. \tag{7.6}$$

is a symmetry, we call it the *conjugation symmetry.*

The conjugation symmetry generalises to CAs with larger state sets: any permutation of the state set can potentially yield a symmetry. The conjugation symmetry corresponds to the only non-identity permutation of $\{0, 1\}$.

Compare these transformations on configurations with the similarly named transformations on rules defined in Section 2.4. Indeed, a CA has reflection symmetry if and only if its local rule is its own reflection, and similarly for conjugation.

It is conventional to talk in terms of groups of symmetries *acting on* sets of objects; see Definition A.20. It is easy to show that the set of all symmetries for a particular CA forms a group under composition of functions. By Example A.21, it follows that the group of symmetries acts on the configuration space. Thus the language of group actions can be used when discussing symmetries of CAs.

Theorem 7.8. *Consider a function $F : X \to X$. A bijection $\alpha : X \to X$ is an automorphism of the functional graph of F if and only if α is a topological conjugation from F to F.*

This follows immediately from the definitions. Applying this theorem to transition graphs, we obtain the following result:

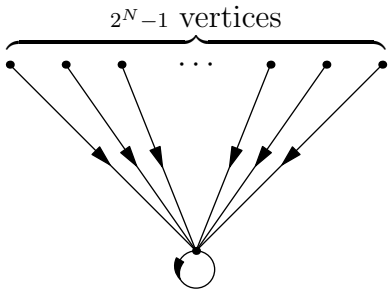
Corollary 7.9. *A bijection α is a symmetry of a CA if and only if α is an automorphism of the CA's transition graph.*

7.3. Counting automorphisms

Let $A(f, N)$ denote the order (number of elements) of the automorphism group for the transition graph of ECA rule f on the lattice of N cells. It is easy to find $A(f, N)$ for some ECAs:

Example 7.10. For rule 204 (the identity rule), the transition graph consists of 2^N basins, each containing a single configuration. Every permutation of configurations is an automorphism, and so there are $2^N!$ automorphisms in total. Clearly this is an upper bound on the number of automorphisms for any ECA. ◊

Example 7.11. For rule 0, the transition graph is



Any permutation of the indicated $2^N - 1$ vertices (corresponding to the configurations other than the all-zeroes configuration) is an automorphism, and so there are $(2^N - 1)!$ automorphisms in total. ◊

The results in this section describe how $A(f, N)$ may be computed in the general case, by induction over the "circles of trees" structure of the transition graph. Intuitively:

(1) An automorphism of the transition graph is obtained by permuting the basins in a way that preserves their isomorphism classes, and applying an automorphism to each basin individually;
(2) An automorphism of a basin is obtained by "rotating" the configurations on the cycle in a way that preserves the isomorphism classes of the trees rooted on the cycle, and applying an automorphism to each of these trees individually;
(3) An automorphism of a tree (rooted on the cycle or not) is obtained by permuting the preimages of the root in a way that preserves the isomorphism classes of the trees rooted at those preimages, and applying an automorphism to each of these subtrees individually;

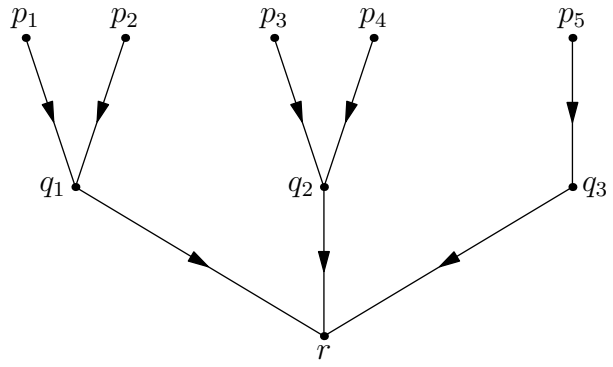


FIGURE 7.1. The tree used in Example 7.13.

(4) There is only one automorphism of a single vertex tree, namely the identity function.

Note that item (3) is recursive, with item (4) as base case. In all cases, a permutation which "preserves isomorphism classes" is one which maps subgraphs to isomorphic subgraphs; such a permutation can be obtained by permuting the elements of each isomorphism class amongst themselves.

The following result, applied recursively, gives the number of automorphisms of a tree:

Lemma 7.12. *Consider a tree with root vertex v, and let $\{u_1, \ldots, u_n\}$ be the vertices in the tree such that there is an edge from u_i to v. Denote by $\{u_i\} / \cong$ the set of isomorphism classes of $\{u_1, \ldots, u_n\}$, considering two vertices to be isomorphic if the subtrees rooted at those vertices are isomorphic. Denote by $A(v)$ the order of the isomorphism group of the tree rooted at v. Then $A(v)$ is defined inductively by*

$$A(v) = \left(\prod_{I \in \{u_i\}/\cong} |I|! \right) \left(\prod_{i=1}^{n} A(u_i) \right) . \tag{7.7}$$

We refer to the vertices $u_1, \ldots, u_n$ as the *children* of vertex v.

In terms of transition graphs, the configurations $u_1, \ldots, u_n$ are the preimages of v, with one exception: if v is on a cycle, then the preimage which precedes it on the cycle is *not* included, because we do not consider the corresponding vertex to be in the tree.

Example 7.13. Consider the tree in Figure 7.1. The isomorphism classes of r's children are $\{q_1, q_2\}$ and $\{q_3\}$. Substituting into Equation 7.7 gives

$$A(r) = (|\{q_1, q_2\}|! \times |\{q_3\}|!) \, (A(q_1)A(q_2)A(q_3)) \tag{7.8}$$

$$= 2A(q_1)^2 A(q_3) \, , \tag{7.9}$$

$\alpha(r)$	$\alpha(q_1)$	$\alpha(q_2)$	$\alpha(q_3)$	$\alpha(p_1)$	$\alpha(p_2)$	$\alpha(p_3)$	$\alpha(p_4)$	$\alpha(p_5)$
r	q_1	q_2	q_3	p_1	p_2	p_3	p_4	p_5
r	q_1	q_2	q_3	p_2	p_1	p_3	p_4	p_5
r	q_1	q_2	q_3	p_1	p_2	p_4	p_3	p_5
r	q_1	q_2	q_3	p_2	p_1	p_4	p_3	p_5
r	q_2	q_1	q_3	p_3	p_4	p_1	p_2	p_5
r	q_2	q_1	q_3	p_4	p_3	p_1	p_2	p_5
r	q_2	q_1	q_3	p_3	p_4	p_2	p_1	p_5
r	q_2	q_1	q_3	p_4	p_3	p_2	p_1	p_5

TABLE 7.1. The automorphisms for the graph in Example 7.13.

noting that $A(q_2) = A(q_1)$ since the trees rooted at q_1 and q_2 are isomorphic. Now q_1's children form a single isomorphism class $\{p_1, p_2\}$, so

$$A(q_1) = |\{p_1, p_2\}|!A(p_1)A(p_2) = 2 \times 1^2 = 2\,, \tag{7.10}$$

using the fact that p_1 has no children and so $A(p_1) = 1$, and similarly for $A(p_2)$. Similarly,

$$A(q_3) = |\{p_5\}|!A(p_5) = 1\,. \tag{7.11}$$

Substituting into Equation 7.9 gives

$$A(r) = 2 \times 2^2 \times 1 = 8. \tag{7.12}$$

In this example, it is not difficult to list all automorphisms, and verify that there are indeed eight of them. See Table 7.1. ◊

Now that we can find the number of automorphisms in a tree, the following result gives the number of automorphisms of a basin of attraction:

Lemma 7.14. *Consider a basin B in a transition graph, and suppose that the vertices of the basin's cycle are $\langle v_1, \ldots, v_m\rangle$. Let $q > 0$ be minimal such that, for all i, the tree rooted at v_i is isomorphic to the tree rooted at v_{i+q}. Clearly q must divide m. Then the order of the automorphism group of the basin is*

$$A(B) = \frac{m}{q}\prod_{i=1}^{m} A(v_i)\,, \tag{7.13}$$

with $A(v_i)$ as defined in Lemma 7.12.

Example 7.15. Consider the graph in Figure 7.2. Here $m = 6$ and $q = 2$, so Lemma 7.14 gives

$$A(H) = \frac{6}{2}\prod_{i=1}^{6} A(r_i)\,. \tag{7.14}$$

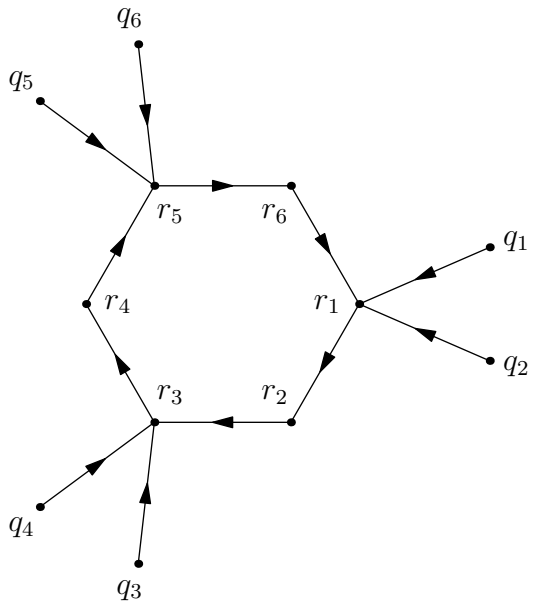


FIGURE 7.2. The graph used in Example 7.15.

Since $A(r_1) = A(r_3) = A(r_5)$ and $A(r_2) = A(r_4) = A(r_6)$ by isomorphism of trees, we have

$$A(H) = 3 \times A(r_1)^3 \times A(r_2)^3 . \tag{7.15}$$

By Lemma 7.12 we have $A(r_1) = 2$ and $A(r_2) = 1$, so

$$A(H) = 3 \times 2^3 \times 1^3 = 24. \tag{7.16}$$

Table 7.2 enumerates the automorphisms, showing that there are indeed 24 of them. ◊

Theorem 7.16. *Consider a transition graph composed of basins* $\{B_1, \ldots, B_k\}$*, and denote the set of isomorphism classes of* $\{B_1, \ldots, B_k\}$ *by* $\{B_i\} / \cong$*. Then the order of the automorphism group of the transition graph is*

$$A(f, N) = \left(\prod_{I \in \{B_i\}/\cong} |I|! \right) \left(\prod_{i=1}^{k} A(B_i) \right) , \tag{7.17}$$

with $A(B_i)$ *as defined in Lemma 7.14.*

Numbers of automorphisms for transition graphs tend to be combinatorially large, as we shall see later. Thus it is often more useful to work with their logarithms.

$\alpha(r_1)$	$\alpha(r_2)$	$\alpha(r_3)$	$\alpha(r_4)$	$\alpha(r_5)$	$\alpha(r_6)$	$\alpha(q_1)$	$\alpha(q_2)$	$\alpha(q_3)$	$\alpha(q_4)$	$\alpha(q_5)$	$\alpha(q_6)$
r_1	r_2	r_3	r_4	r_5	r_6	q_1	q_2	q_3	q_4	q_5	q_6
r_1	r_2	r_3	r_4	r_5	r_6	q_2	q_1	q_3	q_4	q_5	q_6
r_1	r_2	r_3	r_4	r_5	r_6	q_1	q_2	q_4	q_3	q_5	q_6
r_1	r_2	r_3	r_4	r_5	r_6	q_2	q_1	q_4	q_3	q_5	q_6
r_1	r_2	r_3	r_4	r_5	r_6	q_1	q_2	q_3	q_4	q_6	q_5
r_1	r_2	r_3	r_4	r_5	r_6	q_2	q_1	q_3	q_4	q_6	q_5
r_1	r_2	r_3	r_4	r_5	r_6	q_1	q_2	q_4	q_3	q_6	q_5
r_1	r_2	r_3	r_4	r_5	r_6	q_2	q_1	q_4	q_3	q_6	q_5
r_3	r_4	r_5	r_6	r_1	r_2	q_3	q_4	q_5	q_6	q_1	q_2
$\vdots$					$\vdots$	$\vdots$					$\vdots$
r_3	r_4	r_5	r_6	r_1	r_2	q_4	q_3	q_6	q_5	q_2	q_1
r_5	r_6	r_1	r_2	r_3	r_4	q_5	q_6	q_1	q_2	q_3	q_4
$\vdots$					$\vdots$	$\vdots$					$\vdots$
r_5	r_6	r_1	r_2	r_3	r_4	q_6	q_5	q_2	q_1	q_4	q_3

TABLE 7.2. The automorphisms for the graph in Example 7.15.

Corollary 7.17. *In the notation of Lemmas 7.12 and 7.14 and Theorem 7.16, we have*

$$\log A(v) = \sum_{I \in \{u_i\}/\cong} \log |I|! + \sum_{i=1}^{n} \log A(u_i)\,, \tag{7.18}$$

$$\log A(B) = \log m - \log q + \sum_{i=1}^{m} \log A(v_i)\,, \tag{7.19}$$

and

$$\log A(f, N) = \sum_{I \in \{B_i\}/\cong} \log |I|! + \sum_{i=1}^{k} \log A(B_i)\,. \tag{7.20}$$

Computing $A(f, N)$ requires that the entire transition graph of 2^N vertices be generated and traversed, so the computational time complexity is at least exponential with respect to N. We have computed $A(f, N)$ for the 88 essentially different ECAs for all $N \leq 17$, and this seems to approach the limit of what can be computed on a modern desktop PC within a reasonable length of time. Furthermore, the exponential complexity means that even an orders-of-magnitude increase in computational power would not significantly increase this limit. However, the data we are able to generate are sufficient to make some interesting observations.

7.4. Numerical results

Figure 7.3 plots $A(f, N)$ for each of the 88 essentially different ECAs, and Table 7.3 gives selected numerical values. By inspection of Figure 7.3,

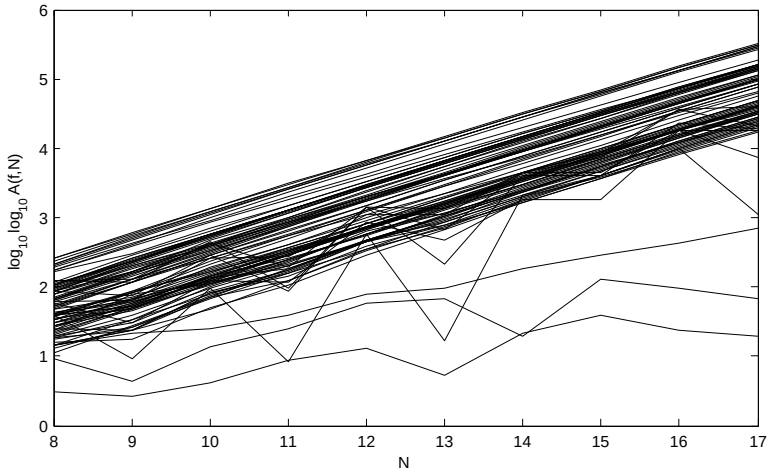


FIGURE 7.3. Plot of $\log_{10}\log_{10} A(f,N)$ against N, for $8 \leq N \leq 17$ and for all 88 essentially different ECA rules.

		N							
Set	Rule	10	11	12	13	14	15	16	17
$\mathcal{L}$	204	3.422	3.770	4.115	4.455	4.792	5.126	5.458	5.788
	15	2.419	2.430	3.034	3.041	3.645	3.649	4.254	4.257
	60	1.976	0.923	2.765	1.221	3.335	3.616	3.994	3.053
$\mathcal{Z}$	90	2.451	1.935	3.141	2.318	3.614	3.617	4.354	4.283
	105	2.644	1.984	3.149	2.833	3.646	3.868	4.555	4.311
	150	2.642	2.282	3.149	3.133	3.646	3.923	4.555	4.612
	154	2.091	2.077	2.945	2.674	3.269	3.268	4.252	3.856
	30	0.624	0.937	1.106	0.724	1.336	1.594	1.369	1.296
$\mathcal{C}$	45	1.144	1.388	1.756	1.822	1.293	2.115	1.987	1.833
	106	1.389	1.590	1.906	1.991	2.258	2.468	2.627	2.849

TABLE 7.3. Values, to 3 decimal places, of $\log_{10}\log_{10} A(f,N)$ for rule 204 (the identity rule, for which $\log_{10}\log_{10} A(f,N)$ is maximal) and the rules in sets $\mathcal{Z}$ and $\mathcal{C}$.

the ECAs can be partitioned into three sets depending on the behaviour of $A(f,N)$ with respect to N: the first and largest set shows an approximate linear relationship between $\log_{10}\log_{10} A(f,N)$ and N, the second shows alternation between larger and smaller values for successive values of N, and the third shows a reduced rate of growth with neither linear nor alternating behaviour. We denote these sets $\mathcal{L}$, $\mathcal{Z}$ and $\mathcal{C}$ respectively, according to whether the relationship between $\log_{10}\log_{10} A(f,N)$ and N is "linear", "zigzag", or "chaotic".

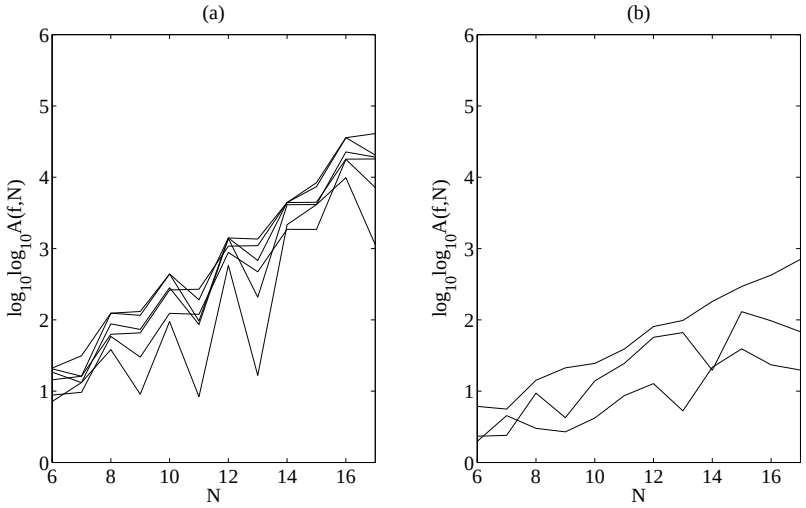


FIGURE 7.4. As Figure 7.3, but for $6 \leq N \leq 17$, and only showing the ECAs in (a) set $\mathcal{Z}$; (b) set $\mathcal{C}$.

7.4.1. Set $\mathcal{L}$: approximate linear relationship

The majority of ECAs show an approximately linear relationship between $\log_{10} \log_{10} A(f, N)$ and N. Furthermore, all of the lines have approximately the same gradient.

Set $\mathcal{L}$ contains examples from all four of Wolfram's classes, including the identity rule 204 and the "Turing complete" rule 110.

7.4.2. Set $\mathcal{Z}$: alternating between large and small values

Figure 7.4 (a) plots $A(f, N)$ for ECA rules 15, 60, 90, 105, 150, and 154. For $N \leq 14$, these rules are characterised by $A(f, N)$ alternating between large values for even N, and smaller values for odd N. This pattern seems to break down, or at least become less pronounced, for $15 \leq N \leq 17$.

All but one of the rules in set $\mathcal{Z}$ are linear (Chapter 3), or linear plus a constant term. All four non-trivial rules of this type (rules 60, 90, 105 and 150) are in $\mathcal{Z}$, as is one of the trivial rules (rule 15); the four remaining trivial rules (0, 51, 170 and 204) are in $\mathcal{L}$.

The single non-linear rule in $\mathcal{Z}$, rule 154, shares one property with the non-trivial linear rules: from an initial configuration consisting of a single cell in state 1, a self-similar "Sierpiński gasket" pattern is produced. On other initial configurations, rule 154 produces periodic patterns, in contrast to the chaotic patterns produced by the non-trivial linear rules.

Recall from Section 6.4 that transition graphs for linear CAs have several special properties. In Section 8.1 we use these properties to investigate numbers of automorphisms for linear CAs, thus shedding some light on the behaviour of the number of automorphisms with respect to N for set $\mathcal{Z}$.

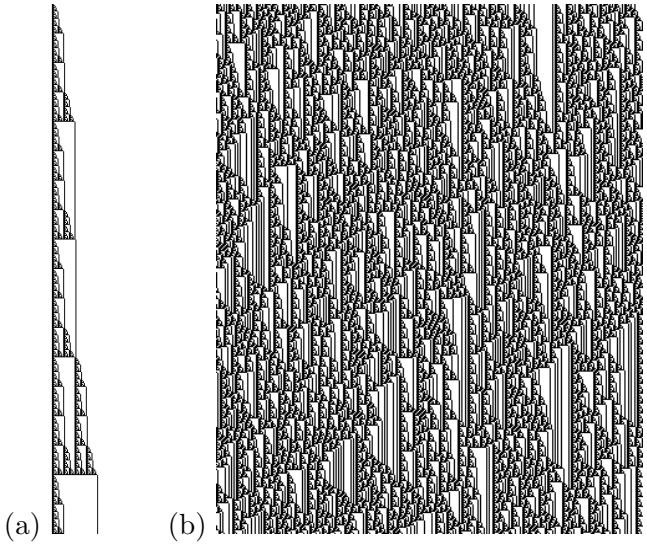


FIGURE 7.5. Evolution of ECA rule 106 from (a) the configuration $\ldots 001100 \ldots$; (b) a random initial configuration. To make the patterns clearer, each configuration has been offset to the right by one cell relative to the previous configuration.

7.4.3. Set $\mathcal{C}$: neither linear nor alternating

Figure 7.4 (b) plots $A(f, N)$ for rules 30, 45, and 106. This plot shows neither a linear nor an oscillating relationship, and the rate of growth of $A(f, N)$ with respect to N seems significantly lower than that for the ECAs in sets $\mathcal{L}$ and $\mathcal{Z}$.

Wolfram [Wol86a] identifies rules 30 and 45 as being particularly suited to random number generation: if the initial configuration assigns state 1 to a single cell and state 0 to all others, then the sequence of states taken by that special cell subsequently is, by several measures, a good pseudo-random sequence over $\mathbb{Z}_2$. Rules 30 and 45 do seem to be unique in this respect: other "chaotic" ECAs which fall into Wolfram's class 3 generally only exhibit this property on a random initial configuration; we may think of them as merely "preserving" the randomness already present in the initial configuration, whereas rules 30 and 45 are uniquely capable of "generating" randomness from an ordered initial configuration.

The evolution of rule 106 from a single cell in state 1 is rather uninteresting: the state 1 simply propagates to the left by one cell per generation. However, two adjacent cells in state 1 yield the self-similar pattern shown in Figure 7.5 (a). On a random initial configuration, rule 106 produces chaotic patterns interspersed with ordered regions, as depicted in Figure 7.5 (b). Again, rule 106 seems to be unique among the ECAs: self-similar patterns

are not uncommon, but the pattern depicted in Figure 7.5 (a) seems more complex than the "Sierpiński gasket"-like patterns typical of other ECAs.

One property that rules 30, 45 and 106 seem to share, and that distinguishes them from other ECAs, is that their transition graphs tend to contain long transients. Indeed, this is one of the properties that make rules 30 and 45 suitable for random number generation.

7.4.4. Discussion

We have obtained a classification of the ECAs according to how their numbers of automorphisms vary with the number of cells. This seems to identify three classes of CA: those with long transients (set $\mathcal{C}$), those that are linear or "almost" linear (set $\mathcal{Z}$), and the rest (set $\mathcal{L}$). This is not a direct analogue of Wolfram's classification, in that it does not correspond to the same trichotomy of ordered versus chaotic versus complex behaviour. However, the sets $\mathcal{Z}$ and $\mathcal{C}$ are (with some exceptions) subclasses of Wolfram's class 3, with $\mathcal{C}$ containing CAs in which chaotic behaviour is "created" in long transients, and $\mathcal{Z}$ containing CAs in which the randomness of the initial configuration is "preserved" as a result of the principle of additive superposition for linear CAs (Theorem 3.2).

7.5. Splitting the expression

From Corollary 7.17, the logarithm of the number of automorphisms of the transition graph can be written in the form

$$\left(\sum_{I\in\{B_i\}/\cong} \log|I|!\right)+\left(\sum_{i=1}^{k}(\log m_i-\log q_i)\right)+\left(\sum_{i=1}^{k}\sum_{i=1}^{m}\log A(v_i)\right)\ . \quad (7.21)$$

These three terms correspond respectively to permutations of basins, rotations of basins, and automorphisms of trees.

Figure 7.6 plots separately the three terms in Equation 7.21. These plots show that none of these terms dominates the overall expression in general, although the difference between the three classes seems most dramatic for the third (tree automorphisms) term. Furthermore, these plots show that there are fluctuations in the first two terms even for some CAs in set $\mathcal{L}$, which is not so apparent in Figure 7.3. The presence or absence of these fluctuations may form the basis of a classification subdividing set $\mathcal{L}$, but this is a subject for future work.

7.6. Conclusion

This chapter began by posing the question of whether the number of automorphisms is correlated with the Wolfram class of the CA, with "simple"

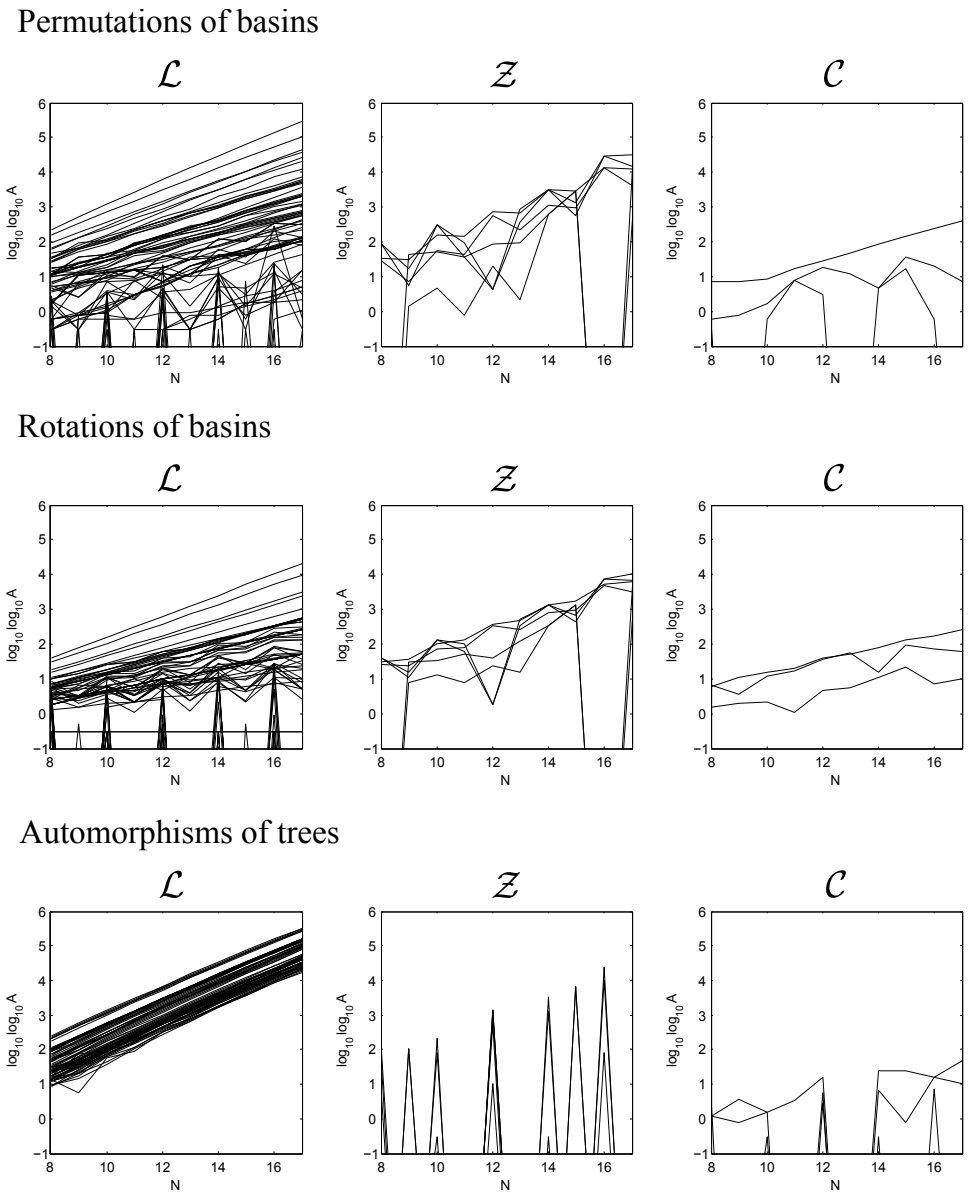


FIGURE 7.6. Plots of the three terms in Equation 7.21, for the ECAs in each of the three classes $\mathcal{L}$, $\mathcal{Z}$ and $\mathcal{C}$.

(classes 1 and 2) CAs having the most automorphisms, and "chaotic" (class 3) CAs having the fewest. There is certainly some truth in this: the members of set $\mathcal{C}$ (including rules 30 and 45) have fewer automorphisms than those of set $\mathcal{Z}$, the difference increasing with N. However, membership of set $\mathcal{C}$ seems to have more to do with transient length than with Wolfram class.

One disappointing aspect of this work is that the vast majority (79 out of 88) of ECA rules, including examples from all four of Wolfram's classes, are in set $\mathcal{L}$. It would be interesting to see whether this remains true for

larger values of N ($N \geq 100$ say), although investigating this would require a different, possibly approximate, method. For example, one might try to express the number of automorphisms so that it is a sum of contributions from each vertex, so that the whole number can be approximated by sampling over the vertex set. However, it is not obvious how to arrange this so that the contributions from each vertex are roughly equal. This is essential: if a handful of vertices have a much larger contribution than the others, then the result obtained will be too dependent on whether the sample includes those vertices.

The interesting aspect of this work is not necessarily the numbers themselves, or even the classification, but rather the suggestion of a relationship between "symmetry" and "complexity". The relationship between sets $\mathcal{L}$ and $\mathcal{C}$ already suggest that, for sufficiently large N, chaotic ECAs such as rules 30 and 45 exhibit far fewer symmetries than the more orderly CAs in set $\mathcal{L}$. It is tempting to conjecture that, as N is made larger still, complex ECAs such as rule 110 will exhibit a critical amount of symmetry somewhere in the region between sets $\mathcal{L}$ and $\mathcal{C}$. This would be in keeping with the "edge of chaos" phenomena observed with parameters such as Langton's λ (Section 5.1) and Wuensche's Z (Section 5.2).

An important result in physics is *Noether's theorem*, which, informally, states that there is a correspondence between symmetries of a physical system and conservation laws of that system [Bae02]. Conservation laws on cellular automata have been studied [TS90, HT91]. Boykett [Boy03] conjectures an analogue of Noether's theorem for CAs, but for a different definition of symmetry. It would be interesting to investigate whether our definition of symmetry has an analogue of Noether's theorem.

CHAPTER 8

Counting automorphisms for linear CAs

In Section 7.3 we give recursive expressions for the number of automorphisms of a transition graph for a CA in general, in terms of the sizes of isomorphism classes of certain families of subgraph. Unfortunately, we are unable to extend this method to large lattices, due to the exponential size of the transition graph. However, this limitation can be overcome in the case of the linear CAs, by exploiting the special properties of their transition graphs described in Section 6.4. Section 8.1 describes the method, and Sections 8.2 and 8.3 apply it to a particular linear ECA to extend the numerical results of Section 7.4.

This work is not directly related to the classification problem, but is an interesting diversion nonetheless, and relates to the theme mentioned in Chapter 1 of overcoming the exponential nature of the configuration space.

8.1. Counting automorphisms

Theorem 8.1. *Consider a transition graph for a linear CA, in which the distinct cycle lengths are $l_1, \ldots, l_k$, and there are m_i cycles of length l_i. Suppose that each of the trees in this transition graph has $A(v)$ automorphisms. Then the transition graph has*

$$A(f, N) = \prod_{i=1}^{k} \left(m_i! \times l_i^{m_i} \times A(v)^{l_i m_i} \right) \tag{8.1}$$

automorphisms.

Proof. Recall that all trees rooted at vertices in cycles are isomorphic in a transition graph for a linear CA (Lemma 6.8).

Consider a basin B_i whose cycle length is l_i. In the notation of Lemma 7.14 we have $m = l_i$, and $q = 1$ since all trees are isomorphic. Substituting into Equation 7.13, we have

$$A(B_i) = \frac{l_i}{1} \prod_{i=1}^{l_i} A(v) = l_i A(v)^{l_i} . \tag{8.2}$$

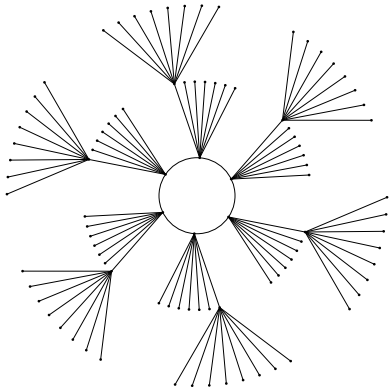


FIGURE 8.1. A basin in the transition graph for the linear CA with state set $\mathbb{Z}_2$ and local rule $f(x_{-3}, \ldots, x_3) = x_{-3} + x_0 + x_2 + x_3$, on $N = 12$ cells. The entire transition graph comprises 40 isomorphic copies of this basin, five basins with cycle length 3, and one basin with cycle length 1. The trees are unbalanced. For example, the configurations $u = 111111111111$ and $v = 001100110011$ are preimages of the zero configuration 000000000000, and neither u nor v is on the cycle. Configuration u is reachable (one of its preimages is 000100010001), but v is unreachable, as are the other five non-zero preimages of the zero configuration.

Two basins are isomorphic if and only if they have the same cycle length. Thus the basins form k isomorphism classes of sizes $m_1, \ldots, m_k$. Thus Theorem 7.16 gives

$$A(f,N) = \left(\prod_{i=1}^{k} m_i!\right)\left(\prod_{i=1}^{k} A(B_i)^{m_i}\right) \tag{8.3}$$

$$= \prod_{i=1}^{k} \left(m_i! \times \left(l_i A(v)^{l_i}\right)^{m_i}\right) \tag{8.4}$$

$$= \prod_{i=1}^{k} \left(m_i! \times l_i^{m_i} \times A(v)^{l_i m_i}\right), \tag{8.5}$$

as required. □

Note that the overall exponent of $A(v)$ in $A(f,N)$ is $\sum_{i=1}^{k} l_i m_i$, which is the number of configurations which appear in cycles.

Say that a tree is *balanced* if all leaf vertices are the same distance from the root. Trees in transition graphs for linear CAs are not always balanced, although empirical investigation shows counterexamples to be rare enough to make balanced trees worth studying. The trees shown in Figures 6.6 and 6.7 are balanced; Figure 8.1 shows an example of a transition graph with unbalanced trees.

The *depth* of a balanced tree is the distance from the root to a leaf. The trees shown in Figure 6.6 have depth 1, and those shown in Figure 6.7 have depth 2.

Theorem 8.2. *Consider a balanced tree of depth d in which the root vertex has in-degree $p-1$, and all other non-leaf vertices have in-degree p. Such a tree has*

$$A_T(d,p) = \frac{p!^{p^{d-1}}}{p} \tag{8.6}$$

automorphisms.

PROOF. If $d = 0$, the tree consists of a single vertex (the root vertex). Clearly this vertex has zero in-degree, and so we must have $p = 1$. (Indeed, the converse of this argument also holds, so that $d = 0$ if and only if $p = 1$.) This "tree" has only a single automorphism (the identity), and

$$A_T(0,1) = \frac{1!^{1^{-1}}}{1} = 1 \tag{8.7}$$

as required.

Suppose $d > 0$, and proceed by induction on d. If $d = 1$, then the tree has $p-1$ leaf vertices, each an immediate predecessor of the root vertex. The automorphisms of this tree are precisely the permutations of the leaves, of which there are $(p-1)!$. Substituting $d = 1$ into Equation 8.6 yields

$$A_T(1,p) = \frac{p!^{p^0}}{p} = \frac{p!}{p} = (p-1)!\,, \tag{8.8}$$

as required.

Now let $d > 1$, and assume, as an inductive hypothesis, that

$$A_T(d-1,p) = \frac{p!^{p^{d-2}}}{p}\,. \tag{8.9}$$

Denote by T_d the tree, as described in the statement of the theorem, of depth d. Now T_d can be obtained from T_{d-1} by taking each leaf v in T_{d-1}, and adding p new leaves as immediate predecessors of v (so that v is no longer a leaf but the root of a subtree of depth 1).

An automorphism of T_d is an automorphism of T_{d-1} composed with an automorphism of each of the new subtrees added in transforming T_{d-1} into T_d. There are $(p-1)p^{d-2}$ such subtrees, each having $p!$ automorphisms, and

so T_d has

$$A_T(d,p) = A_T(d-1,p) \times p!^{(p-1)p^{d-2}} \tag{8.10}$$

$$= \frac{p!^{p^{d-2}} \times p!^{(p-1)p^{d-2}}}{p} \tag{8.11}$$

$$= \frac{p!^{(1+p-1)p^{d-2}}}{p} \tag{8.12}$$

$$= \frac{p!^{p^{d-1}}}{p} \tag{8.13}$$

automorphisms, as required. □

Here p is the number of preimages for a reachable configuration (recall from Lemma 6.9 that all reachable configurations of a linear CA have the same number of preimages). Chapter 9 describes how p may be computed.

8.2. Example: elementary rule 90

Recall that rule 90 is the linear ECA with local rule

$$f(x_{-1}, x_0, x_1) = x_{-1} + x_1 \,. \tag{8.14}$$

The properties of rule 90 are studied extensively by Martin et al [MOW84]; see Sections 3.4 and 6.4. In this section, we use these properties to derive an expression for the numbers of automorphisms in rule 90's transition graphs.

The results in Section 6.4 describe some of the properties of the transition graph for a linear CA. For rule 90, we can go further than this, and can completely describe the trees in the transition graphs:

Lemma 8.3. *Let N be odd. The transition graph for rule 90 on N cells has the following properties:*

1. *All trees consist of a single edge* [MOW84, Theorem 3.3]*, and thus are balanced and have depth $d = 1$;*
2. *Reachable configurations have two preimages, so $p = 2$* [MOW84, Theorem 3.2]*;*
3. *The number of vertices in cycles is 2^{N-1}, i.e. precisely half of all vertices* [MOW84, Corollary to Theorem 3.3].

Lemma 8.4. *Let N be even, and let $D_2(N)$ be the largest power of 2 which divides N:*

$$D_2(N) = \max\left\{2^j \;:\; 2^j | N\right\} \,. \tag{8.15}$$

The transition graph for rule 90 on N cells has the following properties:

1. *All trees rooted at vertices in cycles are balanced and have depth $d = \frac{1}{2}D_2(N)$* [MOW84, Theorem 3.4]*;*

(2) *Reachable configurations have four preimages, so* $p = 4$ [MOW84, Theorem 3.2]*;*
(3) *The number of vertices in cycles is* $2^{N-D_2(N)}$ [MOW84, Corollary to Theorem 3.2].

The only elements missing for a complete description of the transition graph are the cycle lengths and their multiplicities. As far as we are aware, there is no simple expression for these. Martin et al [MOW84] give an algorithm for computing the cycle lengths and multiplicities. A full description is beyond the scope of this thesis, but Appendix E gives an implementation.

The following theorem is obtained from Theorems 8.1 and 8.2 in the cases described in Lemmas 8.3 and 8.4:

Theorem 8.5. *Suppose that, on some number of cells* N*, rule 90 has cycles of distinct lengths* $l_1, \ldots, l_k$*, with* m_i *cycles of length* l_i*. Let*

$$A_T^c = \begin{cases} 1 & \text{if } N \text{ is odd} \\ 24^{2^{N-2}}/4^{2^{N-D_2(N)}} & \text{if } N \text{ is even}, \end{cases} \tag{8.16}$$

where $D_2(N)$ *is as defined in Equation 8.15. Then the transition graph for rule 90 on* N *cells has*

$$A(90, N) = \left(\prod_{i=1}^{k} m_i! \times l_i^{m_i} \right) \times A_T^c \tag{8.17}$$

automorphisms.

Proof. By Theorem 8.1, it suffices to show that

$$A_T^c = \prod_{i=1}^{k} A_T(d, p)^{l_i m_i}\,, \tag{8.18}$$

where $A_T(d, p)$ is as defined in Theorem 8.2, and d and p are chosen appropriately. Note that

$$c = \sum_{i=1}^{k} l_i m_i \tag{8.19}$$

is the total number of vertices on cycles, and so

$$\prod_{i=1}^{k} A_T(d, p)^{l_i m_i} = A_T(d, p)^c\,. \tag{8.20}$$

If N is odd, Lemma 8.3 gives $d = 1$ and $p = 2$. Now

$$A_T(1, 2) = \frac{2! 2^0}{2} = 1\,. \tag{8.21}$$

Indeed, it is easy to see that a tree consisting of a single edge admits only a single automorphism, namely the identity. Thus we have $A_T(1,2)^c = 1$ as required, regardless of the value of c.

If N is even, Lemma 8.4 gives $d = \frac{1}{2}D_2(N)$ and $p = 4$, so

$$A_T(d,p) = \frac{4!4^{\frac{1}{2}D_2(N)-1}}{4} \tag{8.22}$$

$$= \frac{24^{2^{D_2(N)-2}}}{4}\,. \tag{8.23}$$

Lemma 8.4 also gives $c = 2^{N-D_2(N)}$, so

$$A_T(d,p)^c = \frac{24^{2^{D_2(N)-2}\times 2^{N-D_2(N)}}}{4^{2^{N-D_2(N)}}} \tag{8.24}$$

$$= \frac{24^{2^{N-2}}}{4^{2^{N-D_2(N)}}} \tag{8.25}$$

as required. □

8.3. Numerical results for rule 90

Theorem 8.5 yields the number of automorphisms for rule 90 on N cells, if the cycle lengths l_i and multiplicities m_i are known. Martin et al [MOW84] give an algorithm for computing the l_is and m_is. This allows us to compute numbers of automorphisms for any value of N, although naturally some values of N require more computation than others.

Some results are shown in Figure 8.2. Observe that the relationship between the double logarithm of the number of automorphisms and N is approximately linear when N is even; fewer automorphisms occur when N is odd, and there seems to be a lower bound achieved on some (but not all) prime values of N.

Martin et al [MOW84] define the *suborder function* of 2 modulo N, denoted $\mathrm{sord}_N(2)$, by

$$\mathrm{sord}_N(2) = \begin{cases} \min\left\{j \;:\; 2^j \equiv \pm 1 \mod N\right\} & \text{if } N \text{ is odd} \\ 0 & \text{if } N \text{ is even.} \end{cases} \tag{8.26}$$

The suborder function is plotted in Figure 8.3. The suborder function $\mathrm{sord}_N(2)$ has an upper bound of $\frac{1}{2}(N-1)$, achieved on some (but not all) prime values of N.

Comparing Figure 8.3 with Figure 8.2, a correlation is apparent: values of N which yield many automorphisms give small values of $\mathrm{sord}_N(2)$, and vice versa. Indeed, numerical fitting gives that, for $N < 185$,

$$\log_{10}\log_{10} A(90,N) \approx 0.30N - 0.28\,\mathrm{sord}_N(2) - 0.04\,. \tag{8.27}$$

The error in this approximation is illustrated in Figures 8.4 and 8.5.

Theorem 8.5 states that, if N is odd, the number of automorphisms is given by

$$A(90, N) = \left(\prod_{i=1}^{k} m_i! \times l_i^{m_i} \right) . \tag{8.28}$$

The l_is are factors of the maximal cycle length Π_N, and by Lemma 3.15, Π_N is a factor of $2^{\mathrm{sord}_N(2)} - 1$. The m_is have no immediately apparent relationship with $\mathrm{sord}_N(2)$. Thus it is somewhat mysterious that, of all the quantities which determine the cycle lengths and multiplicities, it should be $\mathrm{sord}_N(2)$ which, along with N, ends up dominating the expression for the number of automorphisms.

8.4. Conclusion

Section 7.4 investigates numerically how the number of automorphisms of the transition graph varies with the number of cells. Upon first inspection, this variation seems (with a few exceptions) to be more erratic for the linear CAs than for the nonlinear CAs: in Figure 7.3, it is the linear ECAs which display the most dramatic "zig-zag" patterns. Intuitively, it seems that the non-trivial linear CAs are much more sensitive to changes in the number of cells. Perhaps this is not so surprising: for example, if N is a power of 2, it can be shown that the only attractor for rule 90 on N cells is a fixed point, namely the zero configuration. So there are cases where a small change in the number of cells significantly changes the CA's qualitative dynamics: indeed, changing the number of cells for rule 90 from a non-power of 2 to a power of 2 changes the Wolfram class of the CA from 3 to 1.

Of course, the range of the data shown in Figure 7.3 is insufficient to draw any real conclusions; it is only when the numerical results are extended to much larger values of N, such as in Figure 8.2, that a pattern becomes apparent. The numerical results also show the argument above regarding power-of-2 values of N to be somewhat misleading: in fact, it is around the powers of 2 that we observe the *smallest* fluctuations in numbers of automorphisms.

The data plotted in Figure 7.4 (a) suggest that the numbers of automorphisms for other linear CAs exhibit similar behaviour to that for rule 90. To test this theory, it is desirable to generalise the results of Section 8.2 to other linear CAs. In particular, to apply Theorems 8.1 and 8.2 (which take care of the common, though not universal, case where the trees are balanced), four quantities must be known:

(1) The number of predecessors of a reachable configuration;
(2) The depth of a tree (or equivalently, the maximum transient length);

(3) The set of cycle lengths;
(4) The multiplicity of each cycle length.

Chapter 9 describes how the first of these quantities can be determined. For the other three, it seems plausible that results can be found for individual linear CAs via similar techniques to those used by Martin et al [MOW84] for rule 90. Whether results can be found which apply to *all* linear CAs remains to be seen.

This work is based entirely on two facts about transition graphs for linear CAs: all the trees are isomorphic, and every non-leaf vertex has the same in-degree. These properties certainly do not hold in general, so the approach described here is not applicable to nonlinear CAs. To extend those numerical results to larger N, a completely different approach would be needed.

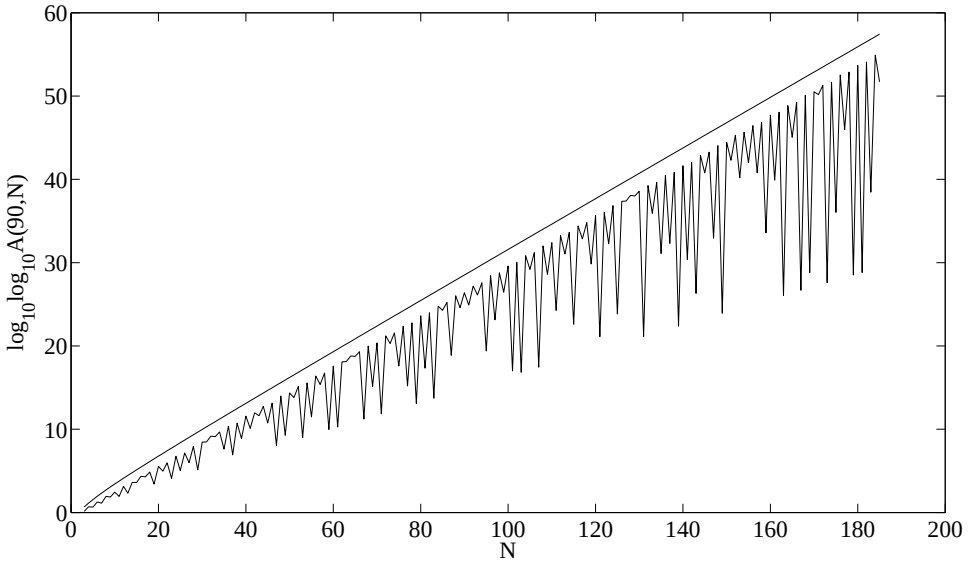


FIGURE 8.2. Plot of $\log_{10}\log_{10} A(90, N)$ (lower line) against N, for $3 \leq N \leq 185$. For comparison, $\log_{10}\log_{10} A(204, N) = 2^N!$ is also plotted (upper line).

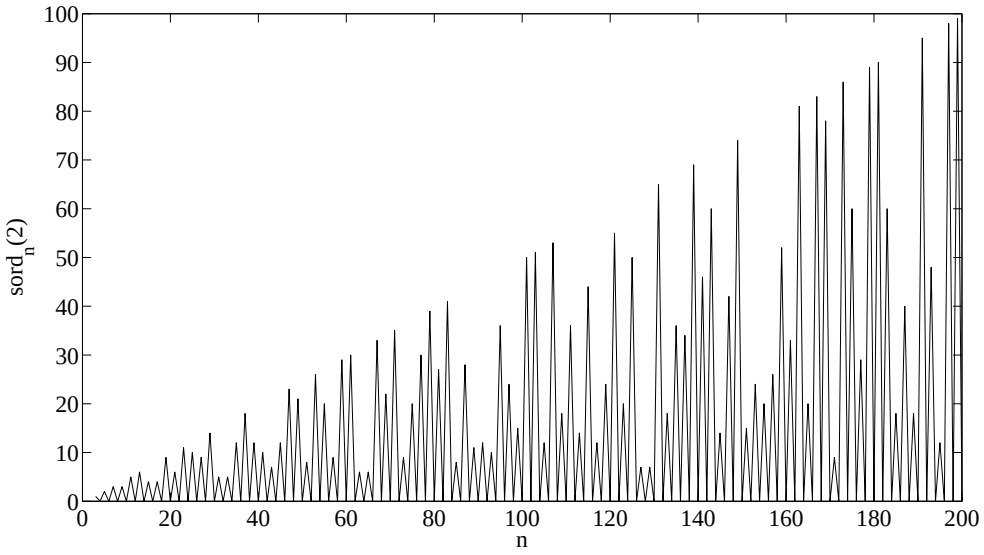


FIGURE 8.3. Plot of the suborder function of 2 modulo n.

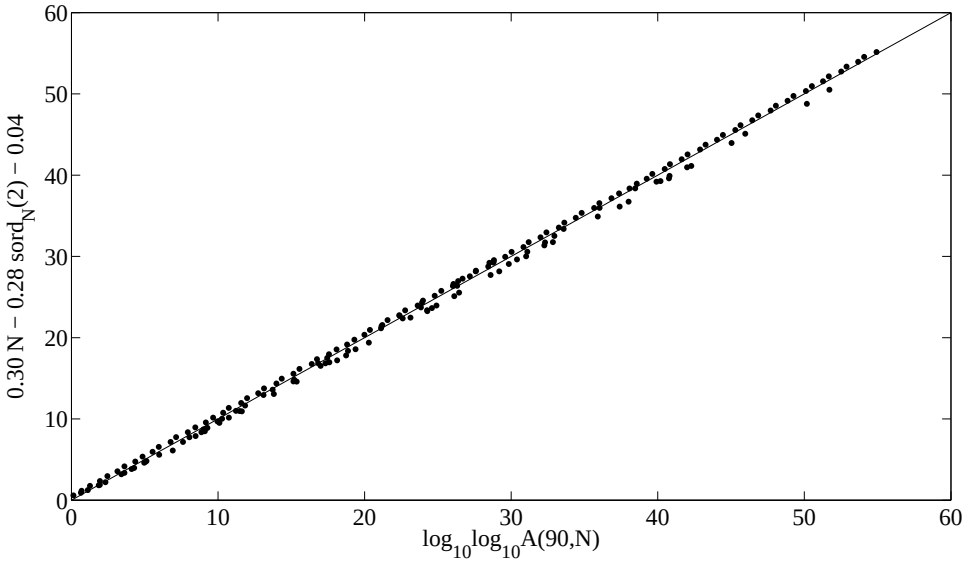


FIGURE 8.4. Plot of the two sides of Equation 8.27. The diagonal "$y = x$" line is plotted for comparison.

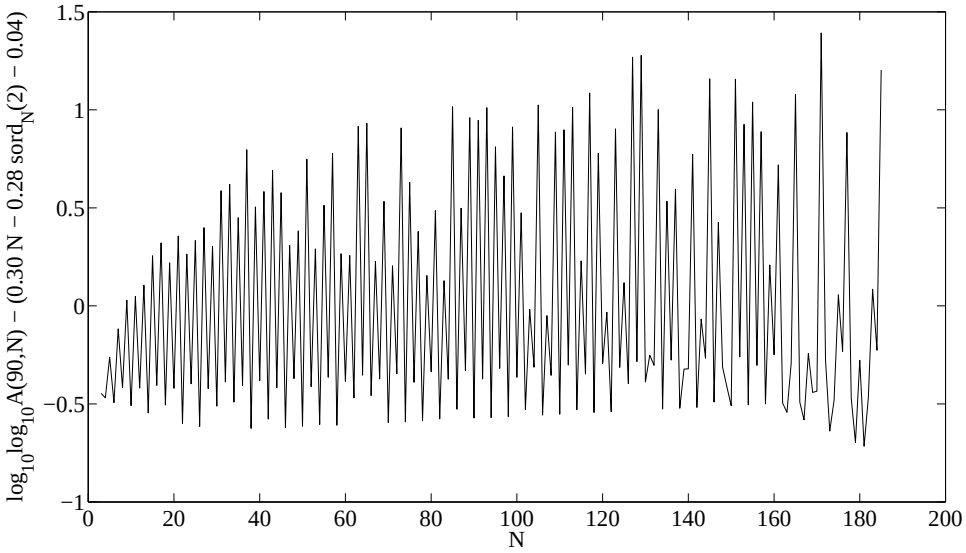


FIGURE 8.5. Plot of the difference between the two sides of Equation 8.27 against N.

CHAPTER 9

Preimages of homogeneous configurations

Theorem 8.2 gives the number of automorphisms of a balanced tree in a transition graph for a linear CA, in terms of two parameters: the depth d of the tree, and the in-degree p of the non-root non-leaf vertices. This p is the number of preimages of a reachable configuration; recall from Lemma 6.9 that all reachable configurations of a linear CA have the same number of preimages, and from the proof of Lemma 6.9 that the zero configuration of a linear CA is always reachable. Thus the problem of finding p reduces to that of counting the preimages of the zero configuration.

This chapter describes two different approaches to counting the preimages of the zero configuration. In fact, the methods are equally applicable to any *homogeneous* configuration (that is, any configuration in which all cells have the same state). The methods do not use the properties of linear CAs, so they are also applicable to homogeneous configurations of nonlinear CAs.

The first approach (Section 9.1) works by considering the possible lengths of sequences of consecutive cells in the same state. When the number of preimages is considered as a function of the number of cells, we see three distinct types of behaviour: constancy, periodicity, and exponential growth. The string length approach gives an intuitive understanding of when and why these behaviours arise.

The second approach (Section 9.2) applies results from spectral graph theory to the de Bruijn graphs of the CA. Specifically, we consider the eigenvalues of the de Bruijn matrices. While this method is not quite as intuitive as considering string lengths, it is far more useful for producing numerical results.

In Section 9.3, we show that this second approach can also be applied to configurations which are heterogeneous but periodic, with period less than N. It can also be applied if the period is N (and thus can be applied to all configurations); however, the period N case is precisely the same as the application of de Bruijn matrices described in Section 4.3, so nothing further is gained in this case.

9.1. String lengths

In this section, we determine numbers of preimages by considering the possible lengths of sequences of consecutive cells in the same state.

9.1.1. For ECAs

Every heterogeneous configuration of an ECA is a cyclic shift of a configuration of the form

$$0^{l_{0,1}}1^{l_{1,1}}\dots 0^{l_{0,k}}1^{l_{1,k}}\,, \tag{9.1}$$

for some positive integers $k, l_{0,1}, \dots, l_{1,k}$. In other words, every heterogeneous configuration can be written, modulo cyclic shift, as strings of zeroes alternated with strings of ones.

Example 9.1. Consider the configuration

$$111001000111110100011111011\,. \tag{9.2}$$

Shifting this configuration by 3 cells to the left gives

$$001000111101000111101111\,, \tag{9.3}$$

which can be written as

$$0^2 1^1 0^3 1^4 0^1 1^1 0^3 1^4 0^1 1^5\,, \tag{9.4}$$

which is in the form of Equation 9.1 with $k = 5$, $l_{0,1} = 2$, $l_{1,1} = 1$, $l_{0,2} = 3$, $l_{1,2} = 4$, and so on. ◊

Since the state set has only two elements, heterogeneity is sufficient to ensure that the configuration contains at least one string of each state: if a configuration does not contain a string of 1s, say, then it must consist solely of 0s, and is thus homogeneous.

Consider the homogeneous configuration q^N. Let x be a state, and let $\overline{x} = 1 - x$ be the other state, so that $\{x, \overline{x}\} = \{0, 1\}$. Assume that there exist preimages of q^N which contain the state $\overline{x}$ (this assumption may turn out to be false; see case (1) below). Consider the permitted lengths $l_{x,i}$ of strings composed of x in such preimages. The assumption that the preimage contains $\overline{x}$ guarantees that these strings do not cover the entire lattice, so each string has a beginning and an end. By considering the effect of the local rule at the beginning, middle and end of the strings, we can write down necessary conditions for various string lengths:

a. $l_{x,i} = 1$ can occur only if $f(\overline{x}, x, \overline{x}) = q$;
b. $l_{x,i} = 2$ can occur only if $f(\overline{x}, x, x) = f(x, x, \overline{x}) = q$;
c. $l_{x,i} \geq 3$ can occur only if $f(\overline{x}, x, x) = f(x, x, x) = f(x, x, \overline{x}) = q$.

Note that condition c implies condition b; that is, if $l_{x,i} \geq 3$ is permitted then $l_{x,i} = 2$ is also permitted. Also note that these conditions are necessary but not sufficient: for example, if $f(\overline{x}, \overline{x}, x) \neq q$ and $f(x, \overline{x}, x) \neq q$, then $l_{x,i} = 1$ cannot occur, even if condition a holds.

We have the following three cases:

(1) If none of these conditions are met then our initial assumption, that there are preimages containing both states x and $\overline{x}$, is contradicted. Thus q^N has no heterogeneous preimages.

(2) Suppose that, for each choice of x, there is precisely one possibility for $l_{x,i}$; say $l_{0,i} = l_0$ and $l_{1,i} = l_1$. This means that, for each x, either condition a or condition b (but not both, and not condition c) is met. Thus $l_0, l_1 \in \{1, 2\}$. Then a heterogeneous preimage of q^N, modulo cyclic shift, has the form

$$0^{l_0}1^{l_1} \dots 0^{l_0}1^{l_1} . \tag{9.5}$$

This defines a valid configuration if and only if N is divisible by $l_0 + l_1$. This configuration is unique up to cyclic shifts, and there are precisely $l_0 + l_1$ distinct configurations which are cyclic shifts of this configuration. Thus the number of heterogeneous preimages of q^N in this case is

$$\begin{cases} l_0 + l_1 & \text{if } l_0 + l_1 | N \\ 0 & \text{otherwise.} \end{cases} \tag{9.6}$$

Note that, since $l_0, l_1 \in \{1, 2\}$, we have $l_0 + l_1 \in \{2, 3, 4\}$.

(3) Suppose that there is more than one possibility for $l_{x,i}$, for either value of x (or indeed for both). Each time a string of x appears, there are several choices for its length. As N increases, so does the number of strings of x and thus the number of choices. A "combinatorial explosion" takes place as N grows larger, and so the number of preimages grows exponentially with respect to N.

These three cases account for the heterogeneous preimages. In addition, there may be zero, one or two homogeneous preimages: specifically, x^N is a preimage of q^N if and only if $f(x, x, x) = q$. The number of homogeneous preimages is thus independent of N.

Example 9.2. Recall the rule table for ECA rule 110:

xyz	111	110	101	100	011	010	001	000
$f(x, y, z)$	0	1	1	0	1	1	1	0

(9.7)

For $q = 0$ and $x = 0$, none of the conditions a, b or c hold, since $f(1, 0, 1) \neq 0$ and $f(0, 0, 1) \neq 0$. Thus the configuration 0^N has no heterogeneous preimages. However, we have $f(0, 0, 0) = 0$ and $f(1, 1, 1) = 0$, so the homogeneous configurations 0^N and 1^N are preimages of 0^N.

For $q = 1$ and $x = 0$, condition a holds since $f(1, 0, 1) = 1$. Neither condition b nor condition c hold, since $f(1, 0, 0) \neq 1$. For $q = 1$ and $x = 1$, conditions a and b hold, but condition c does not, since $f(1, 1, 1) \neq 1$. Thus a heterogeneous preimage of 1^N has the form

$$01^{l_{1,1}}01^{l_{2,1}} \dots 01^{l_{k,1}}, \tag{9.8}$$

modulo cyclic shift, where each $l_{i,1} \in \{1, 2\}$. We already know that both homogeneous configurations are preimages of 0^N, so by elimination 1^N has no homogeneous preimages. ◊

Note that if we have case 2 with $l_x = 2$ (so that condition b holds for this choice of x), then x^N cannot be a preimage, since $f(x, x, x) = q$ would imply condition c. Thus, in case 2 with $l_x = 2$ for one of the choices of x, there is at most one homogeneous preimage; if $l_x = 2$ for both choices of x, then there are no homogeneous preimages.

In summary, when the number of preimages of the homogeneous configuration q^N is considered as a function of N, there are three possible classes of behaviour:

(1) The number of preimages is constant. Indeed, the preimages are themselves homogeneous, and their number can be determined by considering $f(0, 0, 0)$ and $f(1, 1, 1)$ as described above.
(2) The number of preimages is periodic, with period 2, 3 or 4.
(3) The number of preimages grows exponentially with respect to N.

Furthermore, these three possibilities can easily be distinguished by examining the rule table. The results of this for the 88 essentially different ECAs are presented in Table F.1.

Example 9.3. Continuing from Example 9.2, we see that the number of preimages of 0^N is constant with respect to N; specifically, 0^N always has exactly two preimages.

The number of preimages of 1^N grows exponentially with N: such a preimage contains at least $N/3$ strings of 1s, with two choices for the length of each string, so the number of preimages certainly exceeds $2^{N/3}$. ◊

Example 9.4. It is relatively rare for numbers of preimages for an ECA to exhibit period 4 behaviour as a function of N. Indeed, we can show that only ECA rule 90 exhibits period 4 behaviour for preimages of the configuration 1^N.

For period 4 behaviour, the only permitted string length for both states must be 2. For strings of 0s of length 2 to be permitted, we must have

$$f(1,0,0) = f(0,0,1) = 1\,. \tag{9.9}$$

For strings of lengths other than 2 to be forbidden, we must have

$$f(1,0,1) = f(0,0,0) = 0\,. \tag{9.10}$$

A similar argument for strings of 1s shows that

$$f(1,1,0) = f(0,1,1) = 1 \tag{9.11}$$

and

$$f(0,1,0) = f(1,1,1) = 0\,. \tag{9.12}$$

But these conditions completely determine the ECA's rule table, and we find that the rule thus defined is rule 90.

By the same method, we can show that the only ECA exhibiting period 4 behaviour for preimages of 0^N is rule 165. This is the rule obtained from rule 90 by exchanging states 0 and 1. By the convention that each equivalence class is represented by its member with the smallest rule number, rule 165 is not among the 88 essentially different ECA rules. ◊

9.1.2. Beyond ECAs

The situation becomes considerably more complicated when we enlarge the CA's neighbourhood radius. For example, consider the binary state, radius 3 CA whose local rule f is defined by

$$f(0,1,0,1,0,1,0) = 0 \tag{9.13}$$

$$f(1,0,1,0,1,0,1) = 0 \tag{9.14}$$

$$f(0,0,0,1,1,1,0) = 0 \tag{9.15}$$

$$f(1,0,0,0,1,1,1) = 0 \tag{9.16}$$

$$\vdots$$

$$f(0,0,1,1,1,0,0) = 0 \tag{9.17}$$

$$f(x_{-3},\ldots,x_3) = 1 \quad \text{otherwise.} \tag{9.18}$$

Here the preimages of configuration 0^N are those configurations consisting of repetition of the string 01, *or* repetition of the string 000111. In terms of string lengths, there are two distinct cases: either both string lengths are 1, or both string lengths are 3. Thus the possible lengths of strings of 0s depend on the possible lengths of strings of 1s, and vice versa. This kind of interdependence never occurs among the ECAs.

The situation also becomes more complicated if we enlarge the CA's state set. Central to the results for ECAs is the fact that every heterogeneous configuration of a binary state CA can be written, modulo cyclic shift, as strings of 0s alternated with strings of 1s, as in Equation 9.1. A string of 0s must always be followed by a string of 1s, and vice versa. Furthermore, a heterogeneous configuration must contain, but not consist entirely of, a string of 0s. However, in a ternary state CA, a string of 0s may be followed by a string of 1s or a string of 2s, and indeed there is no guarantee that a heterogeneous configuration must contain a string of 0s at all.

9.2. De Bruijn matrices

The approach to counting preimages described in the previous section gives us an intuition for why the number of preimages is sometimes periodic with respect to N: if the possible string lengths are suitably constrained, then they can form a complete configuration only when N is a multiple of the appropriate value. However, the string length approach is somewhat laborious for producing numerical results, and becomes extremely complicated when applied to CAs beyond the ECAs.

This section describes an alternative approach, based on de Bruijn matrices (see Section 4.3). This approach can be considered as an application of *spectral graph theory* to de Bruijn graphs. Spectral graph theory is the study of graphs via the eigenvalues and eigenvectors of their adjacency matrices; see e.g. [CDS95].

The following is the main result for this section:

Theorem 9.5. *Let q be a CA state, and let D_q be the corresponding de Bruijn matrix. Suppose that the eigenvalues of D_q are $\lambda_1, \ldots, \lambda_k$. Then the homogeneous configuration q^N has exactly*

$$\lambda_1^N + \cdots + \lambda_k^N \tag{9.19}$$

preimages.

The proof of this result uses the following lemmas:

Lemma 9.6. *Suppose that matrices A and B share the eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$, with corresponding eigenvalues $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$ respectively. Then AB has eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$, with corresponding eigenvalues $a_1 b_1, \ldots, a_n b_n$.*

Proof. Follows immediately from the definition of eigenvalues and eigenvectors. □

Lemma 9.7. *Let A be a square matrix with eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ and corresponding eigenvalues $a_1, \ldots, a_n$. Then, for any positive integer k, the matrix A^k has eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ with corresponding eigenvalues $a_1^k, \ldots, a_n^k$.*

PROOF. By induction on k, using Lemma 9.6. □

PROOF OF THEOREM 9.5. By Theorem 4.7, q^N has $\operatorname{Tr} D_q^N$ preimages.

It is a well-known result that the trace of a matrix is the sum of its eigenvalues [Gan60, Chapter IV, Section 5]. Thus the number of preimages of q^N is the sum of the eigenvalues of the matrix D_q^N. By Lemma 9.7, the eigenvalues of D_q^N are the Nth powers of the eigenvalues of D_q, namely $\lambda_1^N, \ldots, \lambda_k^N$. Therefore q^N has

$$\lambda_1^N + \cdots + \lambda_k^N \tag{9.20}$$

preimages. □

This result means that, once the eigenvalues of D_q are known, finding the number of preimages of q^N for any value of N is as simple as raising the eigenvalues to the Nth power and summing the results. Furthermore, D_q is a $|S|^{2r}$ square matrix, so it has $|S|^{2r}$ eigenvalues; the size of the matrix and the number of eigenvalues do not depend on N. With respect to N, the time taken by this algorithm is proportional to the time required to raise a complex number to the Nth power.

9.2.1. For ECAs

Example 4.5 gives an expression for the de Bruijn matrices of an ECA. We can apply Theorem 9.5 to find numbers of preimages for homogeneous configurations of ECAs.

Example 9.8. The de Bruijn matrices for ECA rule 18 are

$$D_0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \quad \text{and} \quad D_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{9.21}$$

The eigenvalues of D_1 are all zero, so 1^N has no preimages. The eigenvalues of D_0 are

$$0, 1, \frac{1+\sqrt{5}}{2}, \frac{1-\sqrt{5}}{2}, \tag{9.22}$$

and so 0^N has

$$1 + \frac{1}{2^N}\left(\left(1+\sqrt{5}\right)^N + \left(1-\sqrt{5}\right)^N\right) \tag{9.23}$$

preimages.

These results are consistent with those given in Table F.1: the number of preimages of 1^N is indeed constant (in fact it is zero); for large N,

Equation 9.23 is dominated by

$$\left(\frac{1+\sqrt{5}}{2}\right)^N \approx 1.618^N \tag{9.24}$$

and so the number of preimages of 0^N grows exponentially with N. $\Diamond$

Recall that the eigenvalues of a matrix A are the solutions for λ in the equation

$$\det(A - \lambda I) = 0\,, \tag{9.25}$$

where I is the identity matrix. The expression $\det(A - \lambda I)$ is a polynomial in λ, called the *characteristic polynomial*. If the entries of A are real (as the entries of de Bruijn matrices always are), then the characteristic polynomial has real coefficients, and thus its roots (the eigenvalues of A) must be real or occur in complex conjugate pairs.

If two matrices have the same characteristic polynomial, they have the same eigenvalues. Among the de Bruijn matrices for all 88 essentially different ECAs, there are 23 distinct characteristic polynomials. These are enumerated in Table F.2, their roots are given in Table F.3, and the correspondence between characteristic polynomials and ECA rules is given in Table F.4. From these tables and Theorem 9.5 we can determine the number of preimages as a function of N.

Figure 9.1 plots the number of preimages of 1^N against N, for ECA rule 94 (corresponding to characteristic polynomial c_{10}). The overall trend is exponential, but some fluctuation is also apparent. What is the source of this fluctuation?

As N grows large, the expression

$$\lambda_1^N + \cdots + \lambda_k^N \tag{9.26}$$

is dominated by the terms corresponding to those λ_i whose modulus is maximal. More explicitly, let

$$|\lambda|_{\max} = \max\{|\lambda_i| \ : \ i = 1, \ldots, k\} \tag{9.27}$$

$$\Lambda = \{\lambda_i \ : \ i = 1, \ldots, k, |\lambda_i| = |\lambda|_{\max}\} \tag{9.28}$$

so that $|\lambda|_{\max}$ is the maximal modulus, and Λ is the set of λ_i on which this maximum is attained. Then, for large N,

$$\sum_{i=1}^{k} \lambda_i^N \approx \sum_{\lambda_i \in \Lambda} \lambda_i^N\,. \tag{9.29}$$

There are several possibilities:

(1) If $\Lambda = \{x\}$ for some real $x > 1$, then the sum grows exponentially with N in the manner of x^N.

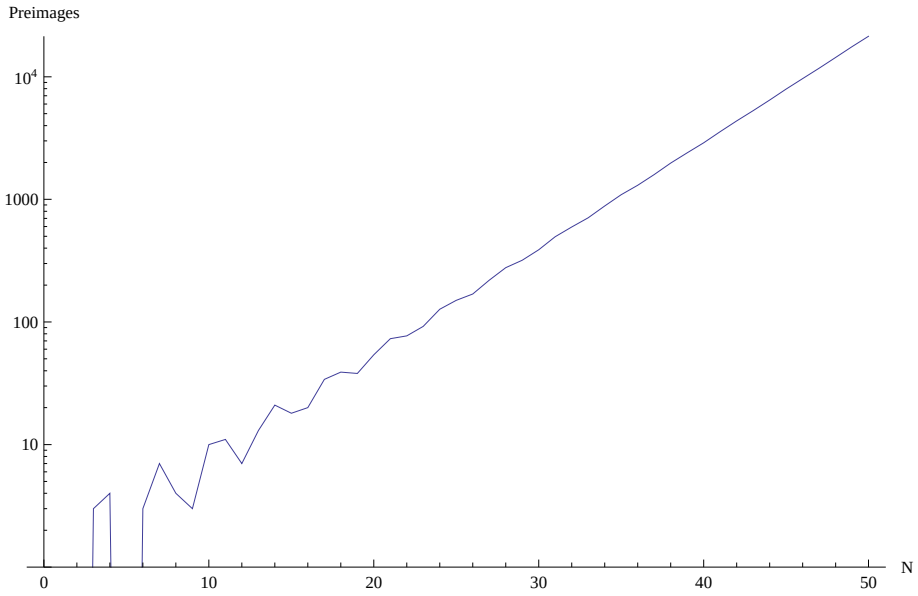


FIGURE 9.1. Plot of number of preimages in ECA rule 94 for the homogeneous configuration 1^N, against number of cells N. Note that the y-axis scale is logarithmic.

(2) If $\Lambda = \{x\}$ for some real $0 < x < 1$, then the sum decays exponentially with N in the manner of x^N.
(3) If $\Lambda = \{1\}$, then the sum approaches 1 as N grows large (the sum over Λ being 1 for all N).
(4) If $\Lambda = \{0\}$, then all of the λ_i must be zero and so the sum is zero.
(5) If $\Lambda = \{x\}$ for some real $x < 0$, the behaviour is analogous to the first three cases, but the sum oscillates between positive and negative. In other words, the behaviour for $\Lambda = \{x\}$ is the behaviour for $\Lambda = \{-x\}$ multiplied by $(-1)^N$.
(6) If $\Lambda = \{z, \bar{z}\}$ for some conjugate pair of complex numbers $z, \bar{z}$, then the sum has the form

$$2r^N \cos N\theta \,. \tag{9.30}$$

This follows immediately from writing $z^N + \bar{z}^N$ in polar form, where r and θ are the modulus and argument of z.

Ignoring the r^N term (which gives exponential growth or decay depending on whether r is greater or less than 1), this is almost periodic with respect to N; whether it is actually periodic depends on whether $\frac{2\pi}{\theta}$ (the ratio of the oscillation period to the "sampling" period corresponding to the integer values of N) is rational or irrational.

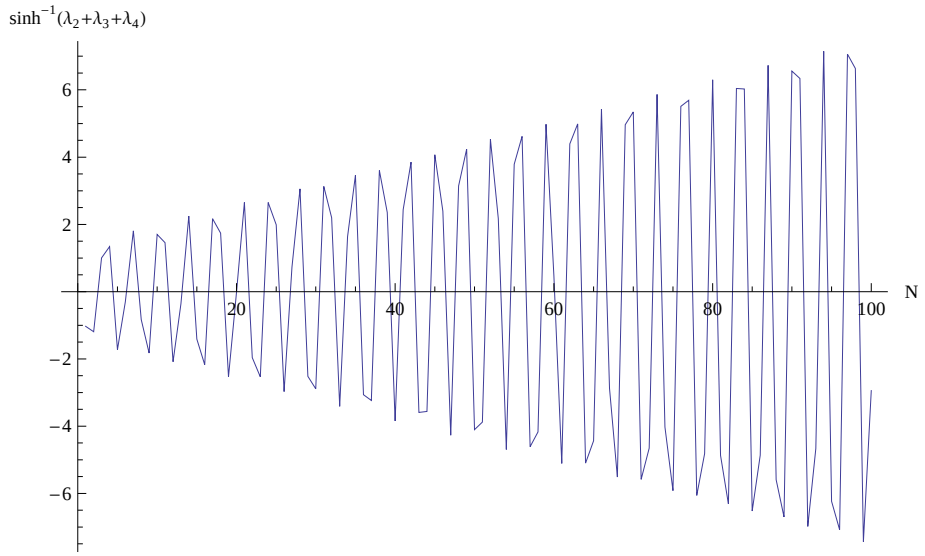


FIGURE 9.2. As Figure 9.1, but with the dominant exponential term subtracted. For this plot, we take the inverse hyperbolic sine of the data, to give the y-axis a "signed logarithmic" scale.

(7) If Λ is a union of two or more of these possibilities, then the overall behaviour is the sum of the corresponding individual behaviours.

For our sum $\lambda_1^N + \cdots + \lambda_k^N$, we can only have cases 1 to 4. This is because $\lambda_1^N + \cdots + \lambda_k^N$ is a number of preimages, and so it cannot be negative.

In each of the exponential cases (c_{10} to c_{23} inclusive) enumerated in Table F.3, we have case 1. Let λ_1 be the eigenvalue with largest magnitude, and omit it to consider the sum

$$\lambda_2^N + \cdots + \lambda_k^N \,. \tag{9.31}$$

An example of this is plotted in Figure 9.2. It is now apparent that the "fluctuations" noted in Figure 9.1 are in fact oscillations.

The sum without λ_1 can be negative, and so all of the cases enumerated above are possible. In many of the cases listed in Table F.3, we have case 6: a conjugate pair of complex numbers dominate, so the overall behaviour is sinusoidal oscillation.

From Equation 9.30, the magnitude of the oscillation is $2r^N$, so the oscillation decays as N tends to infinity if and only if $r < 1$. This is the case for all but three of the cases in Table F.2, the exceptions being c_{10}, c_{18} and c_{19}. Of these, the latter two have $r = 1$ (so the magnitude of the oscillation is constant with respect to N), so only c_{10} gives rise to oscillations whose magnitude grows with N. Thus, among the 88 essentially different ECAs, such growing oscillations only occur with ECA rules 94 and 122 for

homogeneous configuration 1^N (they also occur for those rules, not among the 88 essentially different rules, that are equivalent to rule 94 or rule 122). Although the oscillations in Figure 9.1 appear to decrease in magnitude, they do in fact grow exponentially, just not as rapidly as the λ_1 term.

We can continue to examine the sums in this way until no more terms remain, thus decomposing the overall behaviour into a sum of exponential growths, decays, and sinusoidal oscillations.

9.2.2. Beyond ECAs

De Bruijn matrices are equally applicable to more general 1-D CAs. In general, for a k-state CA with neighbourhood radius r, the de Bruijn matrices are square matrices with k^{2r} rows and columns. Furthermore, Theorem 9.5 applies to general 1-D CAs, the matrix in question having k^{2r} eigenvalues. When the number of preimages is written as a sum of Nth powers of eigenvalues, the cases enumerated in the previous section still apply, so the qualitative types of behaviour are the same as for ECAs.

9.3. Preimages of heterogeneous periodic configurations

Theorem 9.5 can be generalised to any situation where the matrix product

$$D_{c[0]} \times \cdots \times D_{c[N-1]} \tag{9.32}$$

can be expressed as a power of a matrix. For example, consider a configuration c with spatial period p, so that $c[i+p] = c[i]$ for all i. The spatial period p must be a factor of the number of cells N, so N/p is an integer. The corresponding product of de Bruijn matrices is

$$\left(D_{c[0]} \times \cdots \times D_{c[p-1]}\right)^{N/p} . \tag{9.33}$$

By a similar argument to the proof of Theorem 9.5, if the eigenvalues of $D_{c[0]} \times \cdots \times D_{c[p-1]}$ are $\lambda_1, \ldots, \lambda_k$, then the number of preimages is

$$\lambda_1^{N/p} + \cdots + \lambda_k^{N/p} . \tag{9.34}$$

As before, the λ_i are either real or occur in complex conjugate pairs, so the types of qualitative behaviour exhibited by this expression as a function of N are the same as those described above for preimages of homogeneous configurations for ECAs.

All configurations of a finite CA are periodic with period $p = N$. In this case, the powers of N/p disappear, and the above method reduces to simple application of Theorem 4.7. In other words, this method is useful only if $p < N$; it still works for $p = N$, but it does not reduce the amount of computation required.

Example 9.9. As observed in Section 2.7, a common phenomenon in ECA rule 110 is that of *gliders* moving through a spatially and temporally periodic *ether*. We may ask whether there is any configuration in which the gliders or other structures present are annihilated, leaving only ether. In other words, does an ether configuration have any preimages which are not themselves ether configurations?

Modulo cyclic shift, an ether configuration is composed of repetitions of the string $e = 00010011011111$. So an ether configuration can only exist when the number of cells N is a multiple of 14. Let

$$D_e = D_0^3 D_1 D_0^2 D_1^2 D_0 D_1^5 \,, \tag{9.35}$$

where D_0 and D_1 are the de Bruijn matrices for rule 110. Thus an ether configuration has

$$\operatorname{Tr} D_e^{N/14} \tag{9.36}$$

preimages ($N/14$ is the number of repeats of the ether pattern). Direct calculation shows that

$$D_e = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 3 & 2 \end{pmatrix} , \tag{9.37}$$

and that the eigenvalues of D_e are $0, 0, 0, 2$. Therefore the number of preimages is $2^{N/14}$.

In the $N = 14$ case, the preimages of e are

$$p = 11110001001101 \tag{9.38}$$

and

$$q = 11110001110101 \,. \tag{9.39}$$

Configuration p is simply e shifted cyclically by four cells to the right, and is thus also an ether configuration. Configuration q is not an ether configuration.

For larger N, the preimages are all possible sequences composed of repetitions of p and q. For example, for $N = 14 \times 3$, the eight preimages of eee are

$$ppp, ppq, pqp, pqq, qpp, qpq, qqp, qqq \,. \tag{9.40}$$

Precisely one of the preimages (namely $pp \ldots p$) is an ether configuration, leaving $2^{N/14} - 1$ non-ether preimages.

The de Bruijn matrices corresponding to p and q are

$$D_p = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad D_q = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} . \tag{9.41}$$

These matrices do not have the same eigenvectors, so Lemma 9.6 does not apply. However, direct calculation shows that

$$D_p^2 = 2D_p \tag{9.42}$$

$$D_q D_p = D_p \tag{9.43}$$

$$D_p D_q = 2D_q \tag{9.44}$$

$$D_q^2 = D_q \,, \tag{9.45}$$

and so any product involving D_p and D_q can be reduced to a scalar multiple of D_p or D_q, whichever appears last in the product. Furthermore, $\operatorname{Tr} D_p = 2$ and $\operatorname{Tr} D_q = 1$, so the trace of a product involving D_p and D_q is 2^c, where c is the number of times D_p occurs. For example, for $N = 14 \times 3$:

(1) $\operatorname{Tr}(D_p^3) = 2^3 = 8$, so *ppp* has eight preimages;
(2) $\operatorname{Tr}(D_p^2 D_q) = 2^2 = 4$, so *ppq* has four preimages;
(3) Similarly, *pqp* and *qpp* each have four preimages;
(4) *pqq*, *qpq* and *qqp* each have two preimages;
(5) *qqq* has one preimage.

The configuration q has one preimage, namely

$$s = 11011111011100 \,. \tag{9.46}$$

Thus the preimages of a configuration composed of p and q are precisely those configurations obtained by replacing each q with s, and each p with the cyclic shift by four cells to the right of either p or q. These configurations are "pre-preimages" of an ether configuration, or configurations from which an ether configuration is reached after two time steps.

Configuration s has five preimages, so this kind of analysis becomes rather more difficult at this point. In terms of transition graphs, we have reached a particularly "branchy" vertex in the tree. Further investigation of these transients is a subject for future work. ◊

9.4. Conclusion

We have given two different methods of counting preimages of homogeneous configurations. The second method is more useful in terms of producing numerical data; the first gives more insight into the causes of the data's

qualitative relationship with N, and into what the preimages actually are, but is rather cumbersome as a tool for calculation.

Both methods can explain why the number of preimages sometimes oscillates with respect to N. The first method shows that oscillations occur when the preimages have spatial periodicity, which can only occur when the number of cells is a multiple of the spatial period. The second method's explanation is less illuminating but more general: as a sum of powers of complex numbers occurring in conjugate pairs, the expression for the number of preimages often contains terms of the form $2r^N \cos N\theta$, and $\cos N\theta$ is periodic with respect to N (or "nearly periodic" if θ is not a rational multiple of π).

In studying the subset of initial configurations that eventually lead to a homogeneous configuration, we are effectively identifying the subspace of the CA's configuration space on which the CA is in Wolfram's class 1. Numbers of preimages for homogeneous configurations do not directly tell us the size of this region (as we do not count those initial configurations that yield homogeneous configurations after more than one time step), but they may serve as an indication.

CHAPTER 10

Distribution of transition distances

In a continuous dynamical system, we generally expect trajectories to be continuous: the state at time $t+\varepsilon$ (for small positive ε) is usually close to the state at time t, so that there are no discontinuous jumps in the trajectory. This is certainly true of classical physical systems. However, it is not true for discrete dynamical systems in general, nor for CAs in particular: for most reasonable geometric definitions of distance in the configuration space of a CA, there is no reason to expect the configuration at time $t+1$ to be close to the configuration at time t.

This chapter investigates transition distances in CAs: for a configuration c, what is the distance between c and $F(c)$? First, we must define what is meant by the "distance" between two configurations. In mathematics, distance is measured by a *metric* (Definition A.38), a function mapping pairs of elements to nonnegative real numbers, which satisfies the properties we might expect of a measure of distance.

One metric on the configuration space of a CA is the *Hamming distance*, which is defined as the number of cells at which the configurations differ. More formally, define a function $\bar{\delta} : S^2 \to \{0, 1\}$ by

$$\bar{\delta}(x, y) = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases} \tag{10.1}$$

(note that $\bar{\delta} = 1 - \delta$, where δ is *Kronecker's delta function*). Then the Hamming distance $d_h : S^{\mathbb{L}} \times S^{\mathbb{L}} \to \mathbb{R}$ is defined by

$$d_h(u, v) = \sum_{i \in \mathbb{L}} \bar{\delta}(u[i], v[i]) \,. \tag{10.2}$$

Geometrically, the configuration space of a binary state CA on the lattice $\mathbb{Z}_N$ is the vertex set of an N-dimensional hypercube, and the Hamming distance is the Manhattan distance (shortest distance along edges; see Example A.40) between vertices.

Section 10.1 gives numerical results for the transition Hamming distances for the 88 essentially different ECAs. These results suggest a conjecture: in the limit as N tends to infinity, that the Hamming distances are normally distributed. We do not prove or disprove this conjecture, but Section 10.2

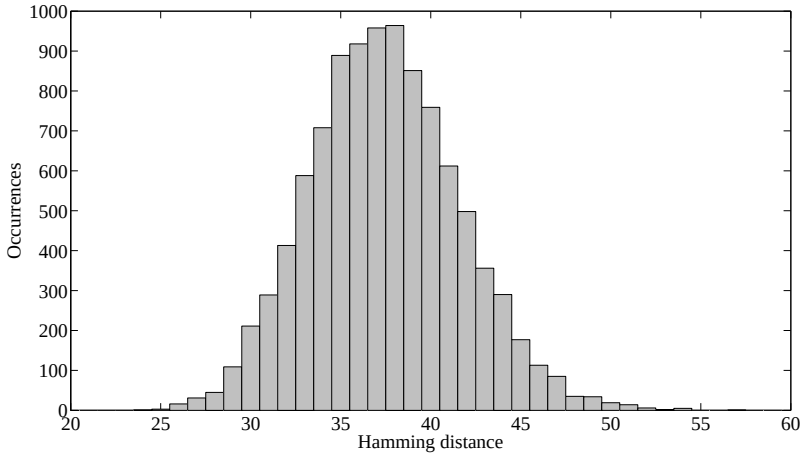


FIGURE 10.1. Histogram of Hamming distances between configurations and their successors under ECA rule 110. The number of cells is $N = 100$, and 10000 configurations were sampled at random, by assigning to each cell the state 0 or 1 with equal probability.

presents strong evidence in its favour by studying the moments of the distributions. Section 10.3 shows that transition Hamming distances are connected to numbers of preimages, so that the work in this chapter relates to the de Bruijn matrices introduced in Section 4.3.

The remainder of this chapter broadens the numerical study beyond that of Section 10.1, by studying subsets of the configuration space such as individual basins of attraction (Section 10.4), multiple transitions instead of single transitions (Section 10.5), and metrics other than the Hamming distance (Section 10.6). In the first two cases we see no deviation from the apparent normal distribution in the limit; for other metrics, we conjecture that the distribution of transition distances has the "same shape" as the underlying distribution of configurations' distances from the zero configuration. The normal distribution of Hamming distances is a corollary to this new conjecture, as here the underlying distribution is binomial and thus normal in the limit.

10.1. Numerical results

Figure 10.1 shows the distribution of Hamming distances between configurations and their successors under rule 110, and Figure 10.2 shows the distributions for all 88 essentially different ECAs.

Figure 10.2 suggests that the distribution always has something close to a "bell curve" shape. The only exceptions to this are rules 51 (the rule which exchanges states 0 and 1, so that the Hamming distance is always N)

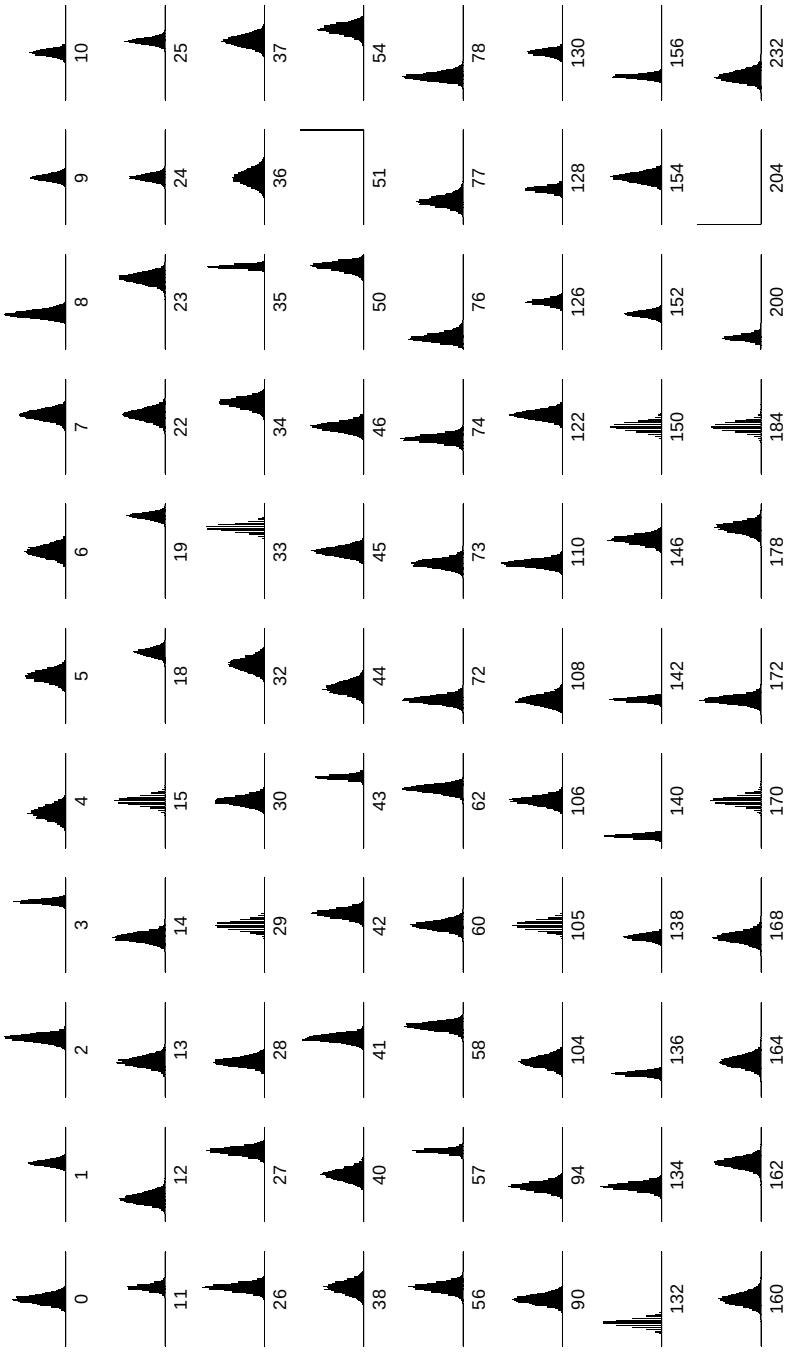


FIGURE 10.2. Histograms of Hamming distances for all 88 essentially different ECAs. As in Figure 10.1, the number of cells is $N = 100$ and the number of samples is 10000. In each case, the x-axis ranges from 0 to 100, and the y-axis is scaled to accommodate the data.

and 204 (the identity rule, for which the Hamming distance is always 0), although both of these cases can be considered as "bell curves" with zero width.

Rule 0 maps every configuration to the zero configuration, so the Hamming distance is precisely the number of 1s in the initial configuration. These numbers form a binomial distribution with $p = \frac{1}{2}$. However, it is easy to verify that the distribution is not binomial for all rules: in particular, there exist pairs of rules with the same mean but different variances.

We may ask how the Hamming distances are distributed as N becomes large. In the limit as N tends to infinity, a binomial distribution tends to a normal distribution, so this is certainly the case for rule 0. This, along with the bell curve shapes observed in Figure 10.2, suggests the following conjecture:

Conjecture 10.1. *In the limit as the number of cells N tends to infinity, the Hamming distances between configurations and their successors are normally distributed.*

We do not have a formal proof (or disproof) of this conjecture. However, the next section does present some compelling evidence in its favour.

10.2. Moments

10.2.1. Definitions

This section gives some standard definitions from probability theory, such as may be found in any textbook on the subject (e.g. [PP02]).

Consider a nonempty multiset X of numbers. The *kth raw moment* (or *kth moment about the origin*) of X is the mean of the kth powers of the elements of X:

$$m'_k = \frac{1}{|X|} \sum_{x \in X} x^k \,. \tag{10.3}$$

The first raw moment is the *mean* of X, denoted μ.

The *kth central moment* (or *kth moment about the mean*) is obtained by shifting X so that its mean is zero:

$$m_k = \frac{1}{|X|} \sum_{x \in X} (x - \mu)^k \,. \tag{10.4}$$

The second central moment is the *variance*, denoted σ^2. The positive square root of the variance is the *standard deviation*, denoted σ.

The kth central moment can be computed from the first k raw moments [PP02, Equation 5-71]:

$$m_k = \sum_{j=0}^{k} \binom{k}{j} (-1)^{k-j} \; m'_j \; \mu^{k-j} \tag{10.5}$$

(note that the zeroth raw moment is $m'_0 = 1$).

The *kth standardised moment* (or *normalised kth central moment*) is defined by

$$\gamma_{k-2} = \frac{m_k}{\sigma^k} \,. \tag{10.6}$$

By definition, the first and second standardised moments have values 0 and 1 respectively. The third standardised moment γ_1 is the *skewness*, and the fourth γ_2 is the *kurtosis*. Note that the standardised moments are undefined if the variance is zero, which occurs if and only if all the elements in X are equal.

The first few standardised moments for a normal distribution, starting with γ_1, are $0, 3, 0, 15, 0, 105, \ldots$. In general, a distribution is not uniquely determined by its moments. However, if the moments are finite and the sum

$$\phi(t) = \sum_{k=0}^{\infty} \frac{(it)^k m'_k}{k!} \tag{10.7}$$

converges absolutely near $t = 0$, then $\phi(t)$ is the *characteristic function* of the distribution, which does determine the distribution uniquely [PP02, Equation 5-105]. Thus, if the distributions are sufficiently well-behaved, having the same moments is a good indicator that two distributions are in fact the same.

In light of this, we recast Conjecture 10.1 in terms of moments:

Conjecture 10.2. *In the limit as the number of cells N tends to infinity, the distribution of Hamming distances between configurations and their successors has the same moments as a normal distribution.*

10.2.2. p-parameters

In this section, we define a family of parameters on the CA's local rule table. Subsequently, we derive expressions for the moments of the Hamming distance distributions in terms of these parameters.

Let $j_1 \ldots j_L$ be a sequence of nonnegative integers. Then the *p-parameter* $p_{j_1 \ldots j_L}$ is defined by

$$p_{j_1 \ldots j_L} = \frac{1}{|S|^{L+2r}} \sum_{x_{1-r} \in S} \cdots \sum_{x_{L+r} \in S} \bar{f}_x(1)^{j_1} \ldots \bar{f}_x(L)^{j_L} \tag{10.8}$$

where

$$\bar{f}_x(i) = \bar{\delta}(x_i, f(x_{i-r}, \ldots, x_{i+r})) \tag{10.9}$$

and adopting the convention that $0^0 = 1$, so that $\bar{f}_x(i)^0 = 1$ regardless of the value of $\bar{f}_x(i)$.

Note that, since $\bar{f}_x(i) \in \{0, 1\}$, we have $\bar{f}_x(i)^{j_i} = \bar{f}_x(i)$ whenever j_i is nonzero. Thus the p-parameter depends only on which of the j_is are zero and which are nonzero, independent of the actual values of the nonzero j_is.

Intuitively, $p_{j_1 \ldots j_L}$ is the proportion of partial configurations $x_{1-r} \ldots x_{L+r}$ in which all cells i with $j_i \neq 0$ have their states changed by application of the global map. This is because $\bar{f}_x(1)^{j_1} \ldots \bar{f}_x(L)^{j_L} = 1$ if and only if $\bar{f}_x(i)^{j_i} = 1$ for all i, otherwise the product is zero. If $j_i = 0$ then $\bar{f}_x(i)^{j_i} = 1$ always, otherwise $\bar{f}_x(i)^{j_i} = 1$ if and only if $\bar{f}_x(i) = 1$; that is, if and only if cell i has its state changed. This is illustrated by the following example:

Example 10.3. Let us compute p_{102} for ECA rule 110. By definition,

$$p_{102} = \frac{1}{2^5} \sum_{x_0 \in S} \cdots \sum_{x_4 \in S} \bar{f}_x(1)^1 \bar{f}_x(2)^0 \bar{f}_x(3)^2 \,. \tag{10.10}$$

Since $\bar{f}_x(i) \in \{0, 1\}$, this simplifies to

$$p_{102} = \frac{1}{32} \sum_{x_0 \in S} \cdots \sum_{x_4 \in S} \bar{f}_x(1) \bar{f}_x(3) \,. \tag{10.11}$$

The 32 cases for $x_0 \ldots x_4$ are enumerated in Table 10.1. As an example, consider the case where $x_0 \ldots x_4 = 01101$. Here we have

$$\begin{aligned} \bar{f}_x(1) &= \bar{\delta}(x_1, f(x_0, x_1, x_2)) && (10.12)\\ &= \bar{\delta}(1, f(0, 1, 1)) && (10.13)\\ &= \bar{\delta}(1, 1) = 0 && (10.14) \end{aligned}$$

and

$$\begin{aligned} \bar{f}_x(3) &= \bar{\delta}(x_3, f(x_2, x_3, x_4)) && (10.15)\\ &= \bar{\delta}(0, f(1, 0, 1)) && (10.16)\\ &= \bar{\delta}(0, 1) = 1 \,, && (10.17) \end{aligned}$$

and so $\bar{f}_x(1)\bar{f}_x(3) = 0 \times 1 = 0$.

The sum of the $\bar{f}_x(1)\bar{f}_x(3)$ column in Table 10.1 is 6, so we have $p_{102} = \frac{6}{32} = \frac{3}{16}$.

Intuitively, $\bar{f}_x(1)\bar{f}_x(3) = 1$ if and only if both cells 1 and 3 have their states changed by the global map, otherwise $\bar{f}_x(1)\bar{f}_x(3) = 0$. Thus the sum of $\bar{f}_x(1)\bar{f}_x(3)$ over all partial configurations $x_0 \ldots x_4$ is the number of those partial configurations for which both cells 1 and 3 change state, and dividing

$x_0 \dots x_4$	$\bar{f}_x(1)$	$\bar{f}_x(3)$	$\bar{f}_x(1)\bar{f}_x(3)$
00000	0	0	0
00001	0	1	0
00010	0	0	0
00011	0	0	0
00100	1	0	0
00101	1	1	1
00110	1	0	0
00111	1	1	1
01000	0	0	0
01001	0	1	0
01010	0	0	0
01011	0	0	0
01100	0	0	0
01101	0	1	0
01110	0	0	0
01111	0	1	0
10000	0	0	0
10001	0	1	0
10010	0	0	0
10011	0	0	0
10100	1	0	0
10101	1	1	1
10110	1	0	0
10111	1	1	1
11000	0	0	0
11001	0	1	0
11010	0	0	0
11011	0	0	0
11100	1	0	0
11101	1	1	1
11110	1	0	0
11111	1	1	1

TABLE 10.1. Enumeration of cases for computation of p_{102} for ECA rule 110 (see Example 10.3).

this by the total number of partial configurations (2^5) gives the proportion. ◊

Let $i_1, \dots, i_k$ be a sequence of integers, not necessarily different, between 0 and $N-1$ inclusive. (For now k is an arbitrary positive integer; when we apply these results in Section 10.2.3, it becomes the same k as in the definitions of Section 10.2.1.) Assuming that N is sufficiently large (specifically $N > 2kr$), the i_js can be shifted cyclically so that $\min(i_j) \geq r$ and $\max(i_j) \leq N - r$, ensuring that, if the i_js are interpreted as indices of cells

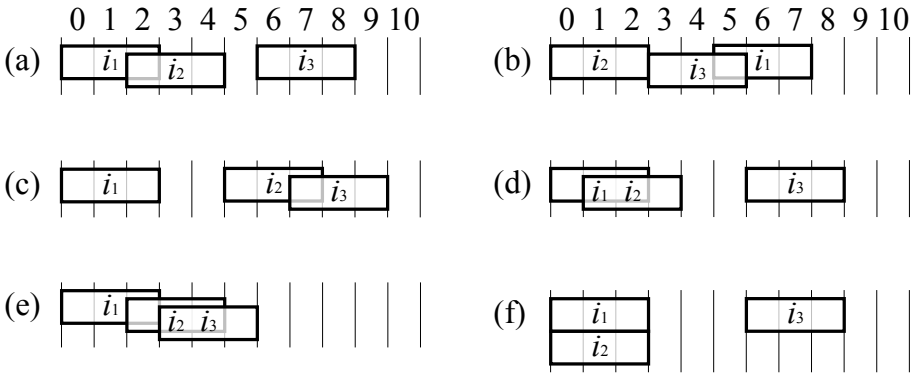


FIGURE 10.3. Examples of summation indices; see Example 10.4. The values of i_1, i_2, i_3 are the numbers of the cells thus labelled. The neighbourhoods of the indices are shown.

in a CA lattice of size N, then none of the cells' neighbourhoods cross the lattice's periodic boundary.

Partition the i_js according to overlapping neighbourhoods: two indices are in the same partition if their neighbourhoods overlap, or if the neighbourhood of the first overlaps with that of another member of the second's partition. More formally, define a binary relation $\sim$ by saying that $i_m \sim i_n$ if and only if $|i_m - i_n| \leq 2r$ (i.e. if and only if i_m's neighbourhood overlaps with i_n's), let $\sim'$ be the transitive closure of $\sim$, and partition the i_js into their $\sim'$-equivalence classes.

Consider a partition $i_1, \ldots, i_c$. Without loss of generality, assume that the i_js are in nondecreasing order, so that the smallest and largest elements in the partition are i_1 and i_c respectively. Associate with this partition a sequence of nonnegative integers $s_{i_1} s_{i_1+1} \ldots s_{i_c-1} s_{i_c}$, such that s_n is the number of i_js with value n.

Example 10.4. Some examples of possible choices of i_js are illustrated in Figure 10.3, for $k = 3$, $r = 1$ and $N = 11$.

In case (a), we have $\langle i_1, i_2, i_3 \rangle = \langle 1, 3, 7 \rangle$. The partitions are $\langle 1, 3 \rangle$ and $\langle 7 \rangle$. The sequences associated with these partitions are 101 and 1 respectively.

In cases (b) and (c), the i_js differ from case (a), but we still get two partitions with sequences 101 and 1.

In case (d), the partitions are $\langle 1, 2 \rangle$ and $\langle 7 \rangle$, which have sequences 11 and 1 respectively.

In case (e), we have a single partition $\langle 1, 3, 4 \rangle$, with sequence 1011.

In case (f), the partitions are $\langle 1, 1 \rangle$ and $\langle 7 \rangle$. The corresponding sequences are 2 and 1. $\Diamond$

The sequences associated with the partitions of the i_js describe the patterns of indices within the "blocks" (partitions) of indices whose neighbourhoods overlap, independent of the arrangement of those blocks on the lattice. There are some restrictions on the sequences which can be obtained this way: their first and last elements must be nonzero, and they cannot contain subsequences of $2r$ or more consecutive 0s.

Lemma 10.5. *Let $i_1, \ldots, i_k$ be integers between 0 and $N-1$, and assume $N > 2kr$. Let $\sigma_1, \ldots, \sigma_n$ be the sequences associated with the partitions of the i_js, as described above. Then*

$$\sum_{x_0 \in S} \cdots \sum_{x_{N-1} \in S} \bar{f}_x(i_1) \ldots \bar{f}_x(i_k) = |S|^N \, p_{\sigma_1} \ldots p_{\sigma_n} \,. \tag{10.18}$$

We do not give a formal proof of this result (although it would be possible to do so), but we justify it with an example:

Example 10.6. In Figure 10.3 (a), we have $i_1 = 1$, $i_2 = 3$ and $i_3 = 7$. Thus the left hand side of Equation 10.18 is

$$LHS = \sum_{x_0 \in S} \cdots \sum_{x_{10} \in S} \bar{f}_x(1) \bar{f}_x(3) \bar{f}_x(7) \,. \tag{10.19}$$

Now $\bar{f}_x(i)$ depends only on x_{i-1}, x_i and x_{i+1}, so $\bar{f}_x(1)\bar{f}_x(3)\bar{f}_x(7)$ does not depend on x_5, x_9 or x_{10}. Therefore these three summations can be eliminated, replacing each by a factor of $\sum_{x \in S} 1 = |S|$:

$$LHS = |S|^3 \sum_{x_0 \in S} \cdots \sum_{x_4 \in S} \sum_{x_6 \in S} \cdots \sum_{x_8 \in S} \bar{f}_x(1) \bar{f}_x(3) \bar{f}_x(7) \,. \tag{10.20}$$

Furthermore, $\bar{f}_x(1)\bar{f}_x(3)$ does not depend on x_6, x_7, x_8, nor $\bar{f}_x(7)$ on $x_0, \ldots, x_4$, so the sum can be split into two:

$$LHS = |S|^3 \left(\sum_{x_0 \in S} \cdots \sum_{x_4 \in S} \bar{f}_x(1) \bar{f}_x(3) \right) \left(\sum_{x_6 \in S} \cdots \sum_{x_8 \in S} \bar{f}_x(7) \right) . \tag{10.21}$$

As noted in Example 10.4, the case of Figure 10.3 (a) corresponds to blocks with sequences 101 and 1. Thus the right hand side of Equation 10.18 is

$$RHS = |S|^{11} \, p_{101} p_1 \,. \tag{10.22}$$

The definition of p-parameters gives

$$RHS = |S|^{11} \left(\frac{1}{|S|^5} \sum_{x_0 \in S} \cdots \sum_{x_4 \in S} \bar{f}_x(1) \bar{f}_x(3) \right) \left(\frac{1}{|S|^3} \sum_{x_0 \in S} \cdots \sum_{x_2 \in S} \bar{f}_x(1) \right) . \tag{10.23}$$

After collecting the $|S|$ terms and relabelling the x_js in the second sum, we have $LHS = RHS$, so Equation 10.18 holds in this case. $\Diamond$

A similar argument holds for all other cases.

10.2.3. Raw moments of Hamming distance distributions

By Equation 10.3, the kth raw moment of the Hamming distances between configurations and their successors is

$$m'_k = \frac{1}{|S|^N} \sum_{u \in S^N} d_h(u, F(u))^k \,. \tag{10.24}$$

Substituting the definitions of the Hamming distance d_h (Equation 10.2) and $\bar{f}$ (Equation 10.9), and splitting the sum over S^N into N sums over S, gives

$$m'_k = \frac{1}{|S|^N} \sum_{x_0 \in S} \cdots \sum_{x_{N-1} \in S} \left(\sum_{i=0}^{N-1} \bar{f}_x(i) \right)^k \,. \tag{10.25}$$

Raising a sum to the kth power yields the sum of all products of k-tuples of the original summands:

$$\left(\sum_{i=0}^{N-1} \bar{f}_x(i) \right)^k = \sum_{i_1=0}^{N-1} \cdots \sum_{i_k=0}^{N-1} \bar{f}_x(i_1) \ldots \bar{f}_x(i_k) \,. \tag{10.26}$$

Substituting this into Equation 10.25 and reordering the summations gives

$$m'_k = \frac{1}{|S|^N} \sum_{i_1=0}^{N-1} \cdots \sum_{i_k=0}^{N-1} \left(\sum_{x_0 \in S} \cdots \sum_{x_{N-1} \in S} \bar{f}_x(i_1) \ldots \bar{f}_x(i_k) \right) \,. \tag{10.27}$$

Applying Lemma 10.5 to the expression in parentheses, we have

$$m'_k = \sum_{i_1=0}^{N-1} \cdots \sum_{i_k=0}^{N-1} p_{\sigma_1} \ldots p_{\sigma_n} \,, \tag{10.28}$$

where $\sigma_1, \ldots, \sigma_n$ are the sequences corresponding to the indices $i_1, \ldots, i_k$, as in Example 10.4. For each term in this sum, the total of all the elements of the sequences $\sigma_1, \ldots, \sigma_n$ is k; assuming that N is sufficiently large ($N > 2kr$), each possible such choice of sequences appears at least once. Thus we can easily determine which products of p-parameters appear in the expression for m'_k, recalling that the valid choices for sequence σ_i have nonzero first and last elements and must not contain $2r$ or more consecutive 0s. Algorithm 10.1 shows how this can be done.

Example 10.7. For a CA with $r = 1$, referring to Algorithm 10.1, we have

$$\text{PSequences}(1, 1) = \{1\} \tag{10.29}$$

$$\text{PSequences}(2, 1) = \{2, 11, 101\} \tag{10.30}$$

$$\text{PSequences}(3, 1) = \{3, 21, 201, 12, 102, 111, 1011, 1101, 10101\} \tag{10.31}$$

```
1: procedure PSequences(k, r)
2:     output k
3:     for j = 1, ..., k − 1 do
4:         for u ∈ PSequences(j, r) do
5:             for z = 0, ..., 2r − 1 do
6:                 output (k − j)0^z u
7: procedure PProducts(k, r)
8:     for u ∈ PSequences(k, r) do
9:         output p_u
10:    for j = 1, ..., k − 1 do
11:        for u ∈ PSequences(j, r) do
12:            for q ∈ PProducts(k − j, r) do
13:                output p_u × q
```

Algorithm 10.1: Listing the products of p-parameters that appear in the expression for the raw moment m'_k.

and so

$$\text{PProducts}(1,1) = \{p_1\} \tag{10.32}$$

$$\text{PProducts}(2,1) = \{p_2, p_{11}, p_{101}, p_1p_1\} \tag{10.33}$$

$$\text{PProducts}(3,1) = \{p_3, p_{21}, p_{201}, p_{12}, p_{102}, p_{111}, p_{1011}, p_{1101}, p_{10101}, p_1p_2, p_1p_{11}, p_1p_{101}, p_1p_1p_1\}. \tag{10.34}$$

Therefore the expression for m'_3 involves the products p_3, p_{21}, ..., p_1p_{101}, and $p_1p_1p_1$.

Note that Algorithm 10.1 does not take into account that multiplication is commutative, and so it often outputs the same product of p-parameters multiple times (for example, PProducts$(3,1)$ outputs both p_1p_2 and p_2p_1). These duplicates are omitted in this example. ◊

The question remains of how many times each product of p-parameters appears.

10.2.4. Mean and variance

We can use the preceding results immediately to find expressions for the mean and variance.

Theorem 10.8. *For $N > 2r$, the mean Hamming distance is Np_1.*

Proof. From Equation 10.28, the mean is

$$\mu = m'_1 = \sum_{i_1=0}^{N-1} p_{\sigma_1} \dots p_{\sigma_n} . \tag{10.35}$$

There is only one way of choosing sequences $\sigma_1, \dots, \sigma_n$ whose elements sum to $k = 1$, namely the single "sequence" 1. Thus $p_{\sigma_1} \dots p_{\sigma_n} = p_1$ in all cases,

so we have

$$\mu = \sum_{i_1=0}^{N-1} p_1 = Np_1 \tag{10.36}$$

as required. □

For example, ECA rule 110 has $p_1 = \frac{3}{8}$, so the mean is $\mu = \frac{3N}{8}$. In Figure 10.1 we have $N = 100$, so the mean is $\frac{300}{8} = 37.5$.

Theorem 10.9. *For $N > 4r$, the variance of the Hamming distance is*

$$N\left(p_2 + 2p_{11} + 2p_{101} + \cdots + 2p_{10^{2r-1}1} - (4r+1)p_1^2\right), \tag{10.37}$$

where 0^{2r-1} denotes a string of $2r-1$ 0s.

Proof. From Equation 10.28, the second raw moment is

$$m_2' = \sum_{i_1=0}^{N-1} \sum_{i_2=0}^{N-1} p_{\sigma_1} \dots p_{\sigma_n}. \tag{10.38}$$

The possible products of p-parameters for $k = 2$ are $p_2, p_{11}, p_{101}, \dots, p_{10^{2r-1}1}$, and p_1p_1. Fix i_1, and consider how many choices of i_2 yield each of these products of p-parameters.

(1) To obtain p_2, both indices must be the same. There is only one choice of i_2 which works, namely $i_2 = i_1$.
(2) To obtain p_{10^z1} (for $0 \le z \le 2r-1$), the indices must be $z+1$ cells apart (i.e. there must be a gap of z cells between the indices). However, it does not matter whether i_2 is $z+1$ cells to the left of i_1, or $z+1$ cells to the right of i_1. Thus there are two choices of i_2, namely $i_2 = i_1 - (z+1)$ and $i_2 = i_1 + (z+1)$. These two cases are distinct for $z \le 2r-1$, given that N is sufficiently large.
(3) We have accounted for $1 + 4r$ of the possible choices for i_2. Thus by elimination, the remaining $N - 4r - 1$ choices must yield the product p_1p_1.

Thus

$$m_2' = \sum_{i_1=0}^{N-1} \left(p_2 + 2p_{11} + 2p_{101} + \cdots + 2p_{10^{2r-1}1} + (N-4r-1)p_1^2\right) \tag{10.39}$$

$$= N\left(p_2 + 2p_{11} + 2p_{101} + \cdots + 2p_{10^{2r-1}1} + (N-4r-1)p_1^2\right). \tag{10.40}$$

By Equation 10.5, the variance (the second central moment) is

$$\sigma^2 = m_2' - \mu^2 \tag{10.41}$$

where μ is the mean. Substituting Equation 10.40 for m_2', and invoking Theorem 10.8 to substitute for μ, gives the required expression. □

Rule	μ/N	σ^2/N	Rule	μ/N	σ^2/N	Rule	μ/N	σ^2/N
0	1/2	1/4	35	7/8	3/64	108	1/4	5/16
1	5/8	7/64	36	1/2	5/8	110	3/8	11/64
2	5/8	11/64	37	5/8	23/64	122	5/8	15/64
3	3/4	1/16	38	5/8	27/64	126	1/2	1/8
4	3/8	31/64	40	1/2	3/8	128	3/8	7/64
5	1/2	3/8	41	5/8	11/64	130	1/2	1/8
6	1/2	3/8	42	5/8	15/64	132	1/4	3/16
7	5/8	19/64	43	3/4	1/16	134	3/8	11/64
8	3/8	11/64	44	3/8	27/64	136	1/4	1/16
9	1/2	1/8	45	1/2	1/4	138	3/8	7/64
10	1/2	1/8	46	1/2	1/4	140	1/8	3/64
11	5/8	7/64	50	7/8	15/64	142	1/4	1/16
12	1/4	5/16	51	1	0	146	5/8	15/64
13	3/8	19/64	54	3/4	5/16	150	1/2	1/4
14	3/8	15/64	56	5/8	15/64	152	3/8	7/64
15	1/2	1/4	57	3/4	1/16	154	1/2	1/4
18	3/4	3/16	58	3/4	3/16	156	1/4	1/16
19	7/8	7/64	60	1/2	1/4	160	1/2	3/8
22	5/8	23/64	62	5/8	11/64	162	5/8	19/64
23	3/4	5/16	72	1/4	3/16	164	3/8	23/64
24	1/2	1/8	73	3/8	15/64	168	3/8	19/64
25	5/8	7/64	74	3/8	11/64	170	1/2	1/4
26	5/8	11/64	76	1/8	15/64	172	1/4	3/16
27	3/4	3/16	77	1/4	5/16	178	3/4	5/16
28	3/8	15/64	78	1/4	3/16	184	1/2	1/4
29	1/2	1/4	90	1/2	1/4	200	1/8	7/64
30	1/2	1/4	94	3/8	15/64	204	0	0
32	5/8	31/64	104	3/8	23/64	232	1/4	5/16
33	3/4	3/16	105	1/2	1/4			
34	3/4	5/16	106	1/2	1/4			

TABLE 10.2. Mean and variance of Hamming distances for the 88 essentially different ECAs, computed using Theorems 10.8 and 10.9.

Table 10.2 gives means and variances, computed using Theorems 10.8 and 10.9, for the 88 essentially different ECAs. Compare with the shapes of the bell curves depicted in Figure 10.2.

10.2.5. Higher moments

It is easy to find an expression, in terms of p-parameters, for a particular raw moment on a particular number of cells N, by iterating over all N^k choices of indices and keeping a tally of how often each product of p-parameters arises. In this section, we couple this approach with some basic

combinatorics to find an expression for a particular raw moment, for all (sufficiently large) values of N. In particular, we use this to investigate the limiting behaviour of the raw moments as N tends to infinity.

The key question is this: how many choices of indices $i_1, \ldots, i_k$ yield a given product of p-parameters? As shown in the proof of Theorem 10.9, this question is relatively easy to answer in simple cases; however, that kind of argument quickly becomes unworkable as k increases.

The following lemma is useful in counting choices of indices:

Lemma 10.10. *Let $S(a, b)$ be the number of distinct sequences of b non-negative integers that sum to a. Then*

$$S(a,b) = \frac{(a+b-1)!}{a!(b-1)!}\,. \tag{10.42}$$

Note that we consider two sequences to be distinct even if they consist of the same elements in a different order.

PROOF. Let $y_0, \ldots, y_b$ be a non-decreasing sequence of integers, with $y_0 = 0$ and $y_b = a$. It follows that $0 \leq y_i \leq a$ for all i. Note that we specify "non-decreasing" rather than "increasing", so consecutive y_is may be equal.

The number of ways of choosing such a sequence is the number of ways of choosing $b-1$ elements (the numbers $y_1, \ldots, y_{b-1}$) from a set of $a+1$ elements (the integers between 0 and a inclusive), regardless of order but with repetition (more succinctly, "$a+1$ choose $b-1$ with repetition"). This number is

$$\frac{((a+1)+(b-1)-1)!}{(b-1)!((a+1)-1)!} = \frac{(a+b-1)!}{a!(b-1)!}\,. \tag{10.43}$$

Define a sequence $x_1, \ldots, x_b$ by $x_i = y_i - y_{i-1}$. This sequence sums to $y_b - y_0 = a$, and the x_is are nonnegative since the sequence of y_is is non-decreasing. This mapping, between sequences $y_0, \ldots, y_b$ as defined above and sequences $x_1, \ldots, x_b$ of nonnegative integers that sum to a, is bijective, and so the numbers of distinct sequences in each case must be equal. The result thus follows from Equation 10.43. □

Example 10.11. The number of ways of choosing 3 numbers that sum to 5 is

$$S(5,3) = \frac{(5+3-1)!}{5!(3-1)!} = 21\,. \tag{10.44}$$

Specifically, the choices are

$$\begin{array}{llllll} 0,0,5 & 1,0,4 & 2,0,3 & 3,0,2 & 4,0,1 & 5,0,0 \\ 0,1,4 & 1,1,3 & 2,1,2 & 3,1,1 & 4,1,0 & \\ 0,2,3 & 1,2,2 & 2,2,1 & 3,2,0 & & \\ 0,3,2 & 1,3,1 & 2,3,0 & & & \\ 0,4,1 & 1,4,0 & & & & \\ 0,5,0 & & & & & \end{array} \tag{10.45}$$

The sequences of y_is corresponding to these x_is are

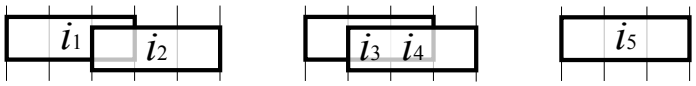
$$\begin{array}{llllll} 0,0,0,5 & 0,1,1,5 & 0,2,2,5 & 0,3,3,5 & 0,4,4,5 & 0,5,5,5 \\ 0,0,1,5 & 0,1,2,5 & 0,2,3,5 & 0,3,4,5 & 0,4,5,5 & \\ 0,0,2,5 & 0,1,3,5 & 0,2,4,5 & 0,3,5,5 & & \\ 0,0,3,5 & 0,1,4,5 & 0,2,5,5 & & & \\ 0,0,4,5 & 0,1,5,5 & & & & \\ 0,0,5,5 & & & & & \end{array} \tag{10.46}$$

$\Diamond$

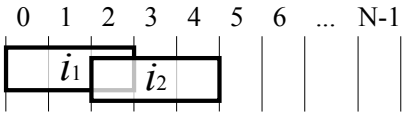
Example 10.12. For a CA with $r = 1$, how many choices of indices $i_1, \ldots, i_5$ yield the product of p-parameters $p_{101}p_{11}p_1$?

The first step is to distribute the five indices among the p-parameters. We assume that i_1 and i_2 correspond to the first and second 1s in p_{101} respectively, i_3 and i_4 to the first and second 1s in p_{11}, and i_5 to the 1 in p_1. This is one of 5! possible choices, but we lose no generality in considering this one specifically.

To find the actual values of the i_js, we must distribute the blocks



over the lattice so that they do not overlap. Without loss of generality, assume that the first block is positioned so that $i_1 = 1$ and $i_2 = 3$, i.e. covering cells 0 to 4 inclusive.



This is one of N possible choices for the position of this block. In any case, we are left with $N-5$ cells over which to distribute the remaining two blocks.

Assume that the second block is to the left of the third; the alternative is equivalent. We must now choose the number of cells between the right edge of the first block (i.e. cell 4) and the left edge of the second block, and similarly the number of cells between the second and third blocks. Both these

numbers must be nonnegative integers, and their sum must not exceed $N-12$ (the total number of uncovered cells once all three blocks are positioned). The number of choices is precisely the number of ways of choosing three nonnegative integers which sum to $N-12$ exactly, which by Lemma 10.10 is

$$\frac{(N-10)!}{2(N-12)!} = \frac{1}{2}(N-10)(N-11)\,. \tag{10.47}$$

Thus, multiplying by $5! \times N \times 2$ to account for previous assumptions, there are

$$120N(N-10)(N-11) \tag{10.48}$$

ways of choosing indices $i_1, \ldots, i_5$ which yield p-parameters $p_{101}p_{11}p_1$. ♢

In the general case, the situation may be more complex. For example, if any of the p-parameters are repeated, then we must be particularly careful not to count any of the choices of indices more than once. However, if we only want to find the number of choices in terms of N up to a constant multiple, the only relevant parts of this process are the choices of the gap lengths between consecutive blocks, and the multiplication by N corresponding to fixing the position of the first block. This leads to the following result:

Theorem 10.13. *Consider the product of p-parameters*

$$p_{j_{1,1}\ldots j_{1,L_1}} \cdots p_{j_{b,1}\ldots j_{b,L_b}}\,. \tag{10.49}$$

Denote by L the total number of cells covered by the corresponding blocks, so $L = L_1 + \cdots + L_b + 2rb$. The number of ways of choosing indices $i_1, \ldots, i_k$ which yield these p-parameters is

$$\frac{cN(N-L+b-1)!}{(N-L)!}\,, \tag{10.50}$$

where c is some constant (c depends on the chosen p-parameters, but is independent of N).

Corollary 10.14. *The number of ways of choosing indices which yield a product of b p-parameters is polynomial in N, with degree b and zero constant term. In particular, the number of choices yielding a single p-parameter is a multiple of N.*

The number of choices can be found computationally for fixed N, by iterating over all N^k choices of indices and counting how many of these choices yield the desired p-parameters. Doing this for a single sufficiently large N is sufficient to determine the constant c.

In principle it is possible to find a general expression for c by combinatorics, rather than finding particular values by computation. However, we have found the combinatorial approach to be prohibitively difficult, as it is

all too easy to arrive at a result which looks correct but breaks down in certain circumstances.

The following example illustrates how c may be found computationally:

Example 10.15. Fix $r = 1$, and let us find the second raw moment m'_2. (Recall that we have already done this by a different method in proving Theorem 10.9.) The relevant products of p-parameters are p_2, p_{11}, p_{101} and p_1^2.

For $N = 6$, we can easily enumerate the $N^k = 36$ possibilities for the indices i_1, i_2 and determine the p-parameter subscripts which correspond to each:

	$i_1 = 0$	1	2	3	4	5
$i_2 = 0$	2	11	101	1, 1	101	11
1	11	2	11	101	1, 1	101
2	101	11	2	11	101	1, 1
3	1, 1	101	11	2	11	101
4	101	1, 1	101	11	2	11
5	11	101	1, 1	101	11	2

(10.51)

Reading the coefficients from the number of times each sequence occurs in the table, the value of m'_2 for $N = 6$ is

$$6p_2 + 12p_{11} + 12p_{101} + 6p_1^2 \,. \tag{10.52}$$

By Corollary 10.14, the coefficients of p_2, p_{11} and p_{101} are multiples of N. Since the coefficients for $N = 6$ are 6, 12 and 12 respectively, the general coefficients are N, $2N$ and $2N$ respectively.

Consider the p_1^2 term. In the notation of Theorem 10.13, we have $L = 6$ and $b = 2$. Thus the coefficient is

$$\frac{cN(N - 6 + 2 - 1)!}{(N - 6)!} = cN(N - 5) \tag{10.53}$$

for some constant c. For $N = 6$ the coefficient is $c \times 6 \times (6 - 5) = 6$, so $c = 1$.

Putting this together, we have

$$m'_2 = N(p_2 + 2p_{11} + 2p_{101} + (N - 5)p_1^2) \,. \tag{10.54}$$

This is consistent with Equation 10.40. ◊

10.2.6. Skewness and kurtosis for radius 1 CAs

The coefficients in the third and fourth raw moments for $r = 1$ are enumerated in Tables 10.3 and 10.4. These allow us to find expressions for the skewness and kurtosis, and thus determine how they behave in the limit as N tends to infinity.

p-parameter subscripts	Coefficient
$1,1,1$	$N(N-7)(N-8)$
$1,101$	$6N(N-7)$
$1,11$	$6N(N-6)$
$1,2$	$3N(N-5)$
10101	$6N$
1011	$6N$
102	$3N$
1101	$6N$
111	$6N$
12	$3N$
201	$3N$
21	$3N$
3	N

TABLE 10.3. Table of coefficients for the third raw moment, for CAs with radius $r = 1$

Theorem 10.16. *As N tends to infinity, the skewness of the Hamming distances for ECAs tends to 0 (assuming that the variance is non-zero).*

PROOF. By Equations 10.5 and 10.6, the skewness (the third standardised moment) is given by

$$\gamma_1 = \frac{m'_3 - 3\mu m'_2 + 2\mu^3}{\sigma^3}\,. \tag{10.55}$$

We have already shown that σ^2 is a multiple of N, so it suffices to show that all terms in the numerator involving powers of N greater than $\frac{3}{2}$ vanish. The numerator is a polynomial in N of degree at most 3, so we need only consider the N^3 and N^2 terms.

From Theorems 10.8 and 10.9, we have

$$\mu^3 = N^3 p_1^3 \tag{10.56}$$

and

$$\mu m'_2 = Np_1 \times N(p_2 + 2p_{11} + 2p_{101} + (N-5)p_1^2) \tag{10.57}$$

$$= N^2 p_1(p_2 + 2p_{11} + 2p_{101} - 5p_1^2) + N^3 p_1^3\,. \tag{10.58}$$

From Table 10.3, the coefficient of N^3 in m'_3 is p_1^3. Thus the coefficient of N^3 in $m'_3 - 3\mu m'_2 + 2\mu^3$ is

$$p_1^3 - 3p_1^3 + 2p_1^3 = 0\,. \tag{10.59}$$

From Table 10.3, the coefficient of N^2 in m'_3 is

$$-15p_1^3 + 6p_1p_{101} + 6p_1p_{11} + 3p_1p_2\,. \tag{10.60}$$

p-parameter subscripts	Coefficient
$1,1,1,1$	$N(N-9)(N-10)(N-11)$
$1,1,101$	$12N(N-9)(N-10)$
$1,1,11$	$12N(N-8)(N-9)$
$1,1,2$	$6N(N-7)(N-8)$
$1,10101$	$24N(N-9)$
$1,1011$	$24N(N-8)$
$1,102$	$12N(N-7)$
$1,1101$	$24N(N-8)$
$1,111$	$24N(N-7)$
$1,12$	$12N(N-6)$
$1,201$	$12N(N-7)$
$1,21$	$12N(N-6)$
$1,3$	$4N(N-5)$
$101,101$	$12N(N-9)$
$101,11$	$24N(N-8)$
$101,2$	$12N(N-7)$
1010101	$24N$
101011	$24N$
10102	$12N$
101101	$24N$
10111	$24N$
1012	$12N$
10201	$12N$
1021	$12N$
103	$4N$
$11,11$	$12N(N-7)$
$11,2$	$12N(N-6)$
110101	$24N$
11011	$24N$
1102	$12N$
11101	$24N$
1111	$24N$
112	$12N$
1201	$12N$
121	$12N$
13	$4N$
$2,2$	$3N(N-5)$
20101	$12N$
2011	$12N$
202	$6N$
2101	$12N$
211	$12N$
22	$6N$
301	$4N$
31	$4N$
4	N

TABLE 10.4. Table of coefficients for the fourth raw moment, for CAs with radius $r = 1$

Thus the coefficient of N^2 in $m'_3 - 3\mu m'_2 + 2\mu^3$ is

$$-15p_1^3+6p_1p_{101}+6p_1p_{11}+3p_1p_2-3p_1(p_2+2p_{11}+2p_{101}-5p_1^2)+0 = 0\,. \quad (10.61)$$

Hence the result. □

Theorem 10.17. *As N tends to infinity, the kurtosis of the Hamming distances for ECAs tends to 3 (assuming that the variance is non-zero).*

Proof. By Equations 10.5 and 10.6, the kurtosis (the fourth standardised moment) is given by

$$\gamma_2 = \frac{m'_3 - 4\mu m'_3 + 6\mu^2 m'_2 - 3\mu^4}{\sigma^4}\,. \quad (10.62)$$

The denominator is

$$\left(\sigma^2\right)^2 = N^2\left(p_1 + 2p_{11} + 2p_{101} + (N-5)p_1^2\right)^2\,. \quad (10.63)$$

We now examine the numerator. From Tables 10.3 and 10.4, it can be shown that the coefficients of N^4 and N^3 are both zero, and the coefficient of N^2 is

$$3\left(p_1 + 2p_{11} + 2p_{101} + (N-5)p_1^2\right)^2\,. \quad (10.64)$$

so that the N^2 term in the numerator is three times the denominator. Thus

$$\gamma_2 = \frac{3aN^2 + bN + c}{aN^2} \quad (10.65)$$

for some a, b, c constant with respect to N, and so γ_2 tends to 3 as N tends to infinity. □

Note that these results only show that the moments have the stated values in the limit; for finite values of N, they may differ. This can be seen in Figure 10.2: the skewness is a measure of a distribution's symmetry about its mean, and some of the distributions in Figure 10.2 are asymmetric and thus have nonzero skewness.

10.2.7. Higher moments for radius 1 CAs

Finding the coefficients in the kth raw moment takes exponential time with respect to k, so we are only able to do so for relatively small values of k. The number of coefficients is also exponential with respect to k, so we do not reproduce tables of coefficients here. However, by using the methods described here to derive expressions for the moments and using symbolic computation (Mathematica) to find their limits, we have verified that the fifth and seventh standardised moments tend to 0 as N tends to infinity, and the sixth standardised moment tends to 15, all of which are consistent with the distribution being normal in the limit.

In summary, we have shown that the first seven standardised moments of the distribution of Hamming distances are consistent with the first seven standardised moments of a normal distribution, in the limit as the number of cells N tends to infinity. This does not prove that the distribution is normal in the limit; however, if we assume that the distributions are sufficiently well-behaved, it shows that any difference between the characteristic functions (Equation 10.7) must be at least of order t^8. This suggests that even if the distribution is not normal in the limit, it is "almost" normal at least.

10.3. Hamming distances and preimage counting

Consider a CA with state set S, and assume that $0, 1 \in S$ (this can always be arranged by renaming states if necessary). Define a new local rule $\bar{f}$ by

$$\bar{f}(x_{-r}, \ldots, x_r) = \begin{cases} 0 & \text{if } f(x_{-r}, \ldots, x_r) = x_0 \\ 1 & \text{if } f(x_{-r}, \ldots, x_r) \neq x_0 \,. \end{cases} \tag{10.66}$$

The corresponding global map $\overline{F}$ maps each cell to state 1 if that cell has its state changed by F, or to state 0 otherwise. In other words,

$$d_h(c, F(c)) = \sum_{i \in \mathbb{L}} \overline{F}(c)[i] \,. \tag{10.67}$$

For a configuration c to have $d_h(c, F(c)) = d$, the configuration $\overline{F}(c)$ must have exactly d cells in state 1 and the remainder in state 0. To put it another way, c must be a preimage under $\overline{F}$ of a configuration with d cells in state 1.

Let $P(d)$ denote the number of configurations c with $d_h(c, F(c)) = d$ ($P(d)/\left|S^{\mathbb{L}}\right|$ is the *probability mass function* for the distribution of Hamming distances). Then $P(d)$ is the sum of numbers of $\overline{F}$-preimages for all configurations with d cells in state 1 and the remainder in state 0. Section 4.3 gives a method for finding these numbers of preimages. Specifically, if $\overline{D}_0$ and $\overline{D}_1$ are the de Bruijn matrices (with respect to $\bar{f}$) for states 0 and 1 respectively, then $P(d)$ is the trace of the sum of all possible products of d copies of $\overline{D}_1$ and $N - d$ copies of $\overline{D}_0$:

$$P(d) = \operatorname{Tr} \left(\sum_{\substack{i_1, \ldots, i_N \in \{0,1\} \\ i_1 + \cdots + i_N = d}} \overline{D}_{i_1} \times \cdots \times \overline{D}_{i_N} \right) \tag{10.68}$$

In the limit as N tends to infinity, this expression involves an infinite sum of infinite products of matrices. However, if Conjecture 10.1 is true and the Hamming distances are normally distributed in the limit, then this expression (after appropriate translation and scaling by the mean and standard

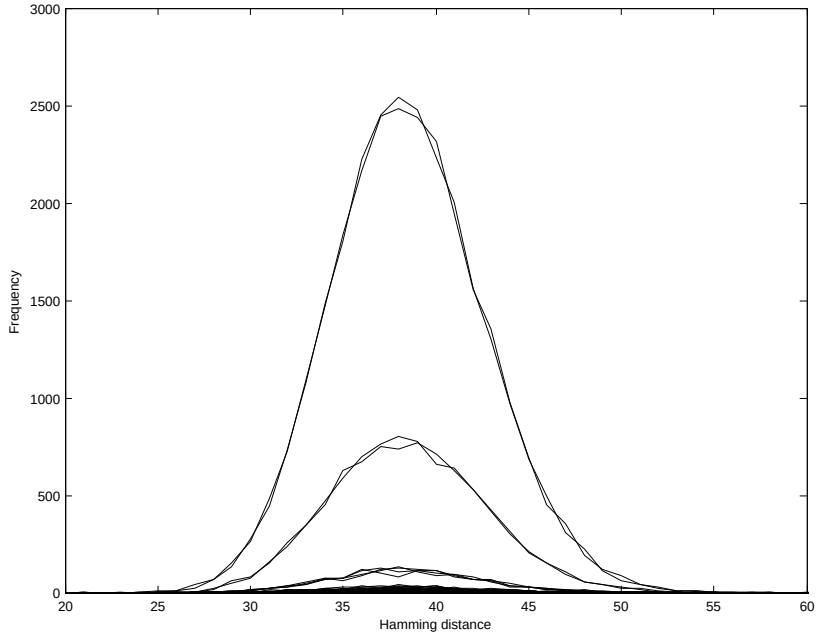


FIGURE 10.4. Histograms of transition Hamming distances for individual basins of attraction in ECA rule 110, with $N = 100$ cells and 100,000 samples.

deviation) must tend to the probability density function for a normal distribution.

As an aside, note that the parameter p_1 for f is precisely Langton's λ parameter (Section 5.1) for $\bar{f}$ with quiescent state 0.

10.4. Individual basins of attraction

That the Hamming distances always follow a normal (or almost normal) distribution may seem somewhat surprising: especially for Turing complete rules such as ECA rule 110, we might expect the distribution of transition distances to be more interesting than this. However, the distributions studied so far span the entire configuration space, so it is conceivable that any finer structure in particular regions of the configuration space (such as those in which the CA is actually emulating a universal computer) is obscured.

We can obtain numerical results for the distributions of Hamming distances in individual basins of attraction, by sampling over the configuration space, using the algorithm of Section 6.1 to determine which basin the sampled configuration is in, and maintaining a histogram for each basin encountered. Figure 10.4 shows a sample of the results.

These numerical results suggest that the Hamming distances within individual basins have the same distribution, appropriately scaled, as the overall configuration space.

Note that some basins in Figure 10.4 occur in pairs, with roughly the same distribution. These pairs of basins are isomorphic in the transition graph, and indeed the configurations in one basin are cyclic shifts by one cell of the configurations in the other. Note that it is not necessarily the case that isomorphic basins have the same distribution of transition distances, although they do have the same distribution if the configurations in one basin are cyclic shifts, or some other symmetry that preserves Hamming distances, of the other.

If interesting structure is being obscured by uninteresting regions of configuration space, then studying individual basins of attraction does not uncover this structure. Perhaps studying other subsets of the configuration space (attractor cycles, reachable configurations, configurations at a certain depth in the trees of the transition graph, individual trajectories) would yield better results, but this is left as a subject for future work.

10.5. Multiple transitions

The previous sections discuss the distribution of $d_h(c, F(c))$ as c ranges over the configuration space. Instead of considering a single transition, we may consider t transitions for any positive integer t. In other words, how is $d_h(c, F^t(c))$ distributed?

If F is the global map of a CA with neighbourhood radius r, then F^t is the global map of a CA with neighbourhood radius rt. In other words, multiple transitions of one CA can be thought of as a single transition of another, "larger" CA. In particular, if Conjecture 10.1 holds and single transition Hamming distances are normally distributed in the limit for all CAs, then multiple transition distances are also normally distributed in the limit.

How does the distribution of $d_h(c, F^t(c))$ vary with t? Some numerical results are shown in Figure 10.5. Two classes of behaviour are apparent, as t becomes large: either the distribution approaches the binomial distribution with $p = \frac{1}{2}$ (which can be seen by comparison with the distribution for rule 0), or it does not. If it does not, it may tend to a different bell curve distribution, or it may oscillate between two or more distributions.

Intuitively, there seem to be two cases in which we would expect the distribution to approach the binomial distribution in the long term:

(1) If a rule is in Wolfram's class 1, then $F^t(c)$ is a homogeneous configuration for large t. Thus the distribution of distances is precisely

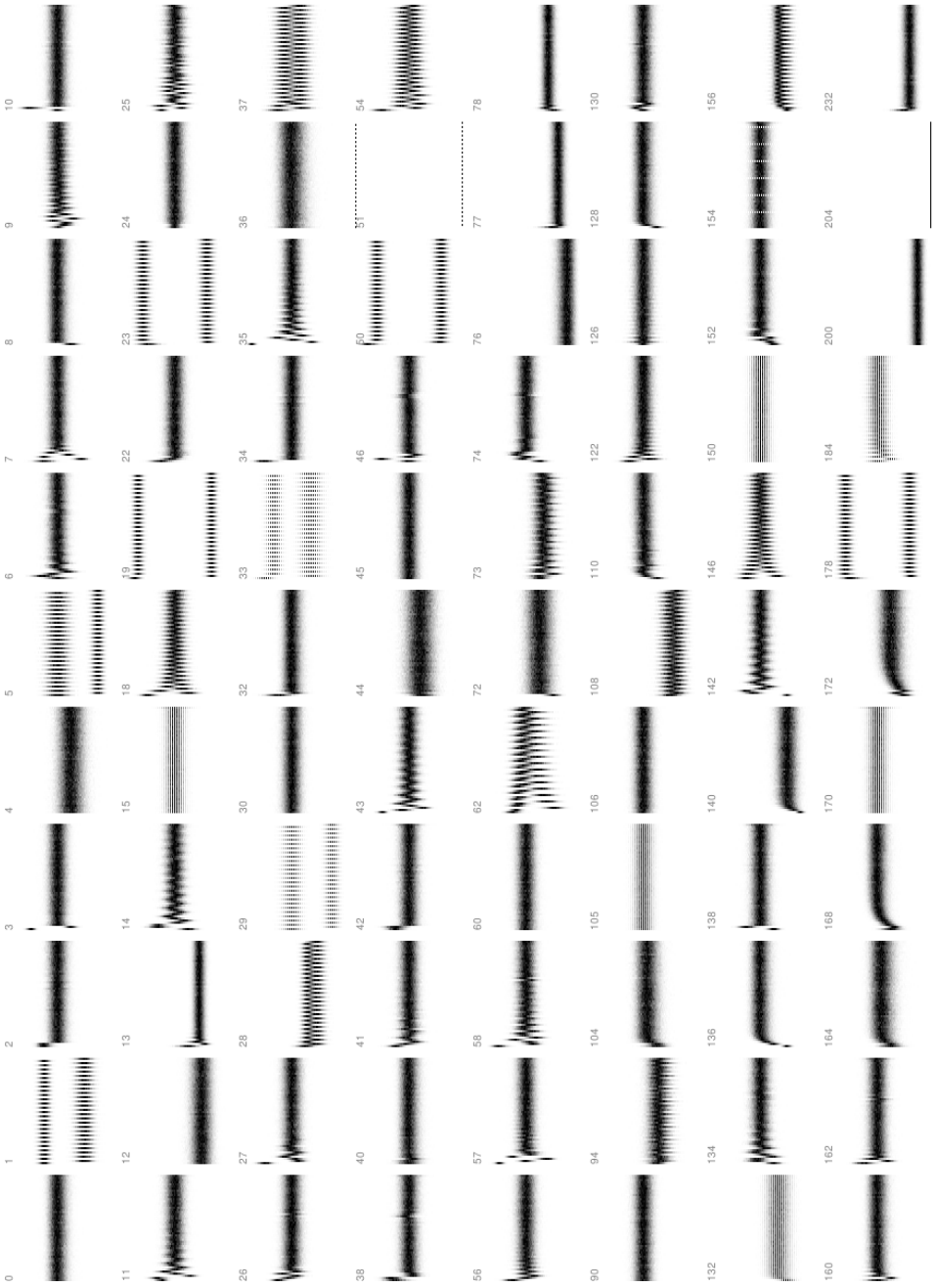


FIGURE 10.5. "Histograms" of multiple transition Hamming distances for all 88 essentially different ECAs. The number of cells is 100, and the sample size is 2500. In each plot, the horizontal axis denotes the number of transitions t, ranging from 1 to 50, and the vertical axis denotes Hamming distances from 0 to 100. The frequency of each Hamming distance is represented by a shade of grey, from white (zero) to black (maximum).

the "underlying" distribution of configurations' distances from a homogeneous configuration, which is binomial.

(2) If a rule is "unpredictable", i.e. in Wolfram's class 3 (or possibly 4), then we expect no apparent correlation between c and $F^t(c)$. Thus the distribution of distances is the "underlying" distribution of distances between any two configurations chosen at random, which is binomial.

This seems to suggest that only class 2 CAs should exhibit behaviour other than approaching the binomial distribution. Comparing the results in Figure 10.5 to the Wolfram classes listed in Appendix B, this certainly looks plausible, although many class 2 CAs also approach a binomial distribution.

10.6. Other metrics

There are numerous metrics, other than the Hamming distance, which may be defined on a set of strings. Chapman [Cha06] lists thirty, and by no means is this list exhaustive. This section gives some numerical results for two metrics.

10.6.1. The hypercube metric

Let d_k be the metric defined in Equation 5.12. Recall that the distance between two configurations with respect to this metric is defined as 2^{-k} where, roughly speaking, k is the shortest distance from the origin to a cell where the configurations differ.

Figure 10.6 shows the distribution of d_k distances for rule 110, and Figures 10.7 and 10.8 show the distribution for all 88 essentially different ECAs. As illustrated by the linear trends visible in Figure 10.8, the distribution for $d_k \leq \frac{1}{2}$ (i.e. $\log_2 d_k \leq -1$) seems to obey a power law. The point $d_k = 1$ is seemingly independent of this, and the frequency of $d_k = 1$ may be higher or lower than that of $d_k = \frac{1}{2}$.

Under this metric, what is the underlying distribution of configurations' distances from the all-zeroes configuration? Let $p(k)$ denote the probability that, for a given configuration c, $d_k(c, 0) = 2^{-k}$. First, note that $d_k = 2^0$ precisely when cell 0 has state 1; thus $p(0) = \frac{1}{2}$. For $k > 0$, $p(k)$ is the probability that all cells $N - k + 1, \ldots, N - 1, 0, \ldots, k - 1$ have state 0, but either cell $N - k$ or cell k, or both, have state 1. Thus

$$p(k) = \frac{1}{2^{2k-1}} \times \frac{3}{4} = \frac{3}{2^{2k+1}} = \frac{3}{2}\left(2^{-k}\right)^2 \ . \tag{10.69}$$

Notice that this does not depend on N; however, the value of N does place an upper bound (namely $\frac{N}{2}$) on the values of k for which this argument works.

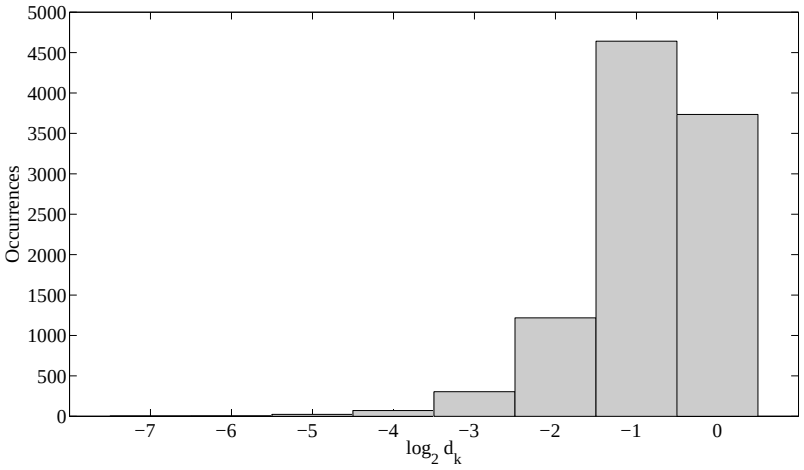


FIGURE 10.6. Histogram of distances with respect to the metric d_k between configurations and their successors under ECA rule 110. The number of cells is 100, and 10000 configurations were chosen at random.

Also notice that, as observed in relation to Figure 10.8, this probability does indeed obey a power law with respect to $d_k = 2^{-k}$.

10.6.2. The Levenshtein distance

The *Levenshtein distance* d_l between two strings is the minimum number of edit operations required to transform one string into the other, where each edit operation is an insertion, deletion or substitution of a single character. The Hamming distance can be thought of as the number of substitution operations required to transform one string into the other, so the Levenshtein distance is always less than or equal to the Hamming distance.

A cyclic shift by one cell yields a Levenshtein distance of at most two, as it is equivalent to deleting a character from one end of the string and inserting it at the other. In contrast, the Hamming distance obtained by a cyclic shift depends on the precise cell states, and is binomially distributed if the cell states are random (the distribution being related to the probability that adjacent cells have the same state).

Some numerical results for the Levenshtein distance for a single transition are plotted in Figure 10.9. Like the Hamming distances, the distributions are bell curves without exception. Also like the Hamming distances, the underlying distribution of configurations' distances from the zero configuration is binomial; indeed, it is easy to see that $d_l(c, 0) = d_h(c, 0)$ for all configurations c (intuitively, there is no faster way to edit the zero configuration into c than to apply a substitution at each nonzero cell).

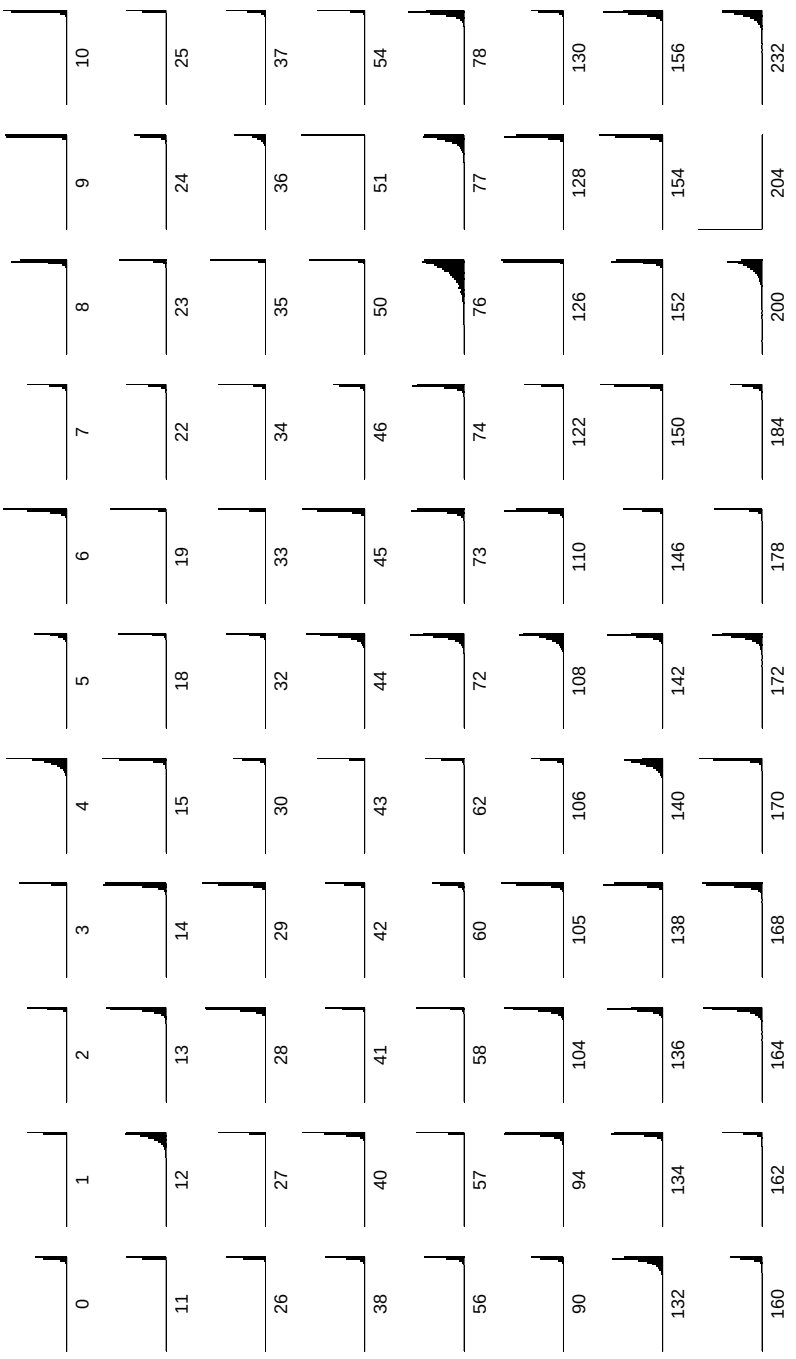


FIGURE 10.7. Histograms of d_k distances for all 88 essentially different ECAs. As in Figure 10.6, the number of cells is 100 and the number of samples is 10000. In each case, the logarithmic x-axis ranges from 2^{-49} to 2^0, with $d_k = 0$ plotted as 2^{-50}. The y-axes are scaled to accommodate the data in each plot.

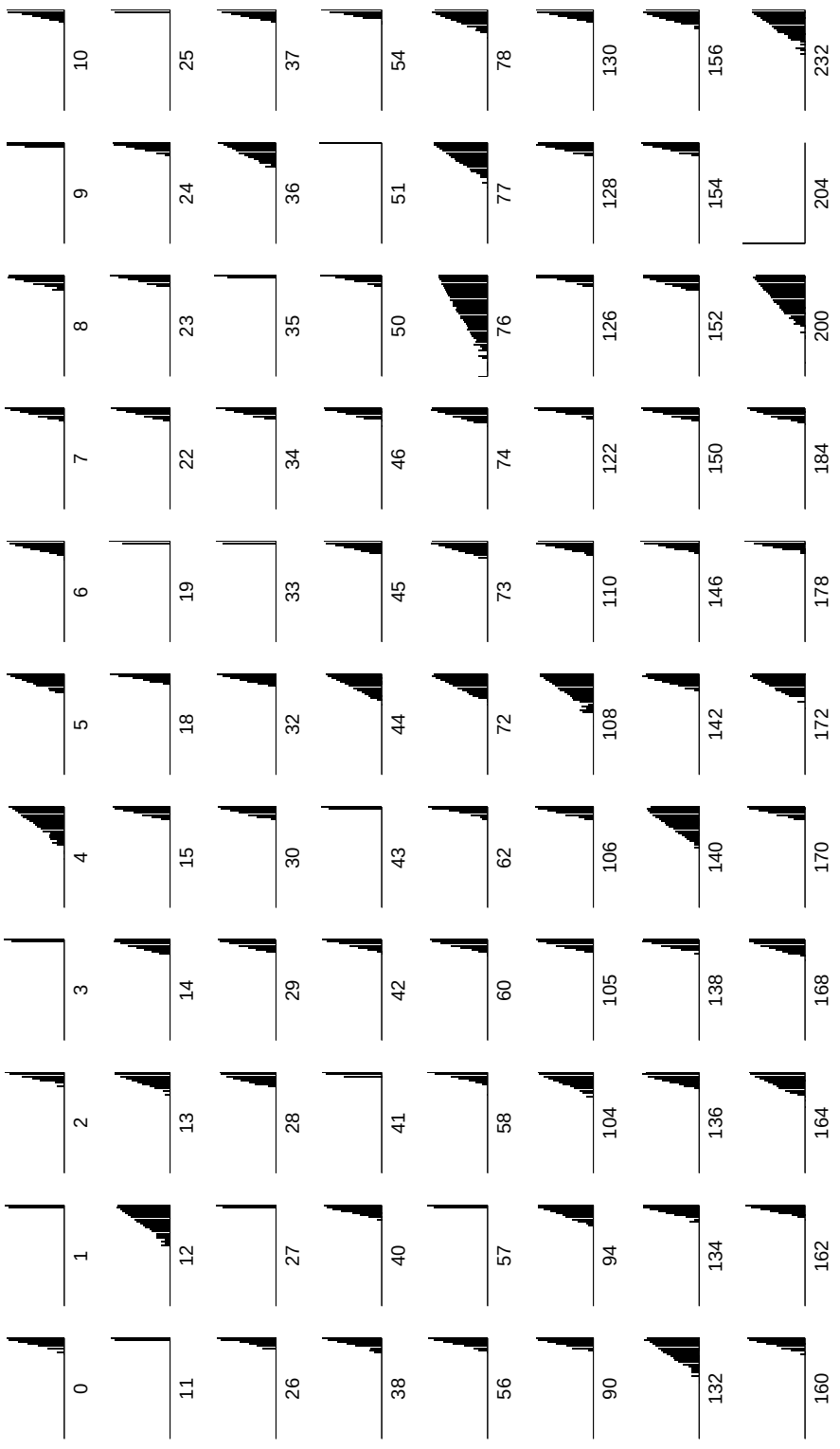


FIGURE 10.8. As Figure 10.7, but with logarithmic y-axes.

10.6.3. Discussion

The more complex definitions of these metrics as compared to the Hamming distance makes their theoretical study rather more difficult. Figure 10.9 seems to show "bell curve" distributions for the Levenshtein distances, but the methods used to derive expressions for the moments of the Hamming distance distributions, which rely on the fact that the Hamming distance can be defined as a sum over the cells in the lattice, do not seem easy to apply here.

The results presented in this section hint at a more general form of Conjecture 10.1:

Conjecture 10.18. *Let d be any metric on the configuration space of a CA. In the limit as the number of cells tends to infinity, the distribution of distances between configurations and their successors has the same shape (for some reasonable definition of "shape") as the underlying distribution of the configuration space from the zero configuration.*

10.7. Conclusion

We have empirical evidence that the distribution of transition distances for CAs depends not on the CA itself, but on the underlying distribution of configurations' distances from the zero configuration. It is not clear whether this is simply a result of "averaging" over the entire configuration space, or something inherent in the nature of CAs. One can imagine that most properties of a physical system would appear uninteresting if averaged over the entire phase space. The numerical investigation of Section 10.4 is an unsuccessful attempt to address this issue; the similar experiments suggested in that section may be more fruitful.

There is no apparent correlation between transition distances and Wolfram classes. A normal distribution is parameterised by its mean and variance; as shown in Section 10.2.4, the mean and variance for the Hamming distances can be expressed in terms of p-parameters. The p-parameters are simple parameters on the local rule, somewhat reminiscent of Langton's λ parameter (Section 5.1), and so are unlikely to give a good indication of an ECA's Wolfram class.

It is not so surprising that transition distances should be a poor indicator of Wolfram class: Wolfram classes explicitly relate to the long-term dynamics of the CA, whereas a single transition is as short-term as can be. Indeed, the distributions of Hamming distances for multiple transitions (Section 10.5), particularly in the limit as the number of transitions grows large, seems more promising from a classification perspective.

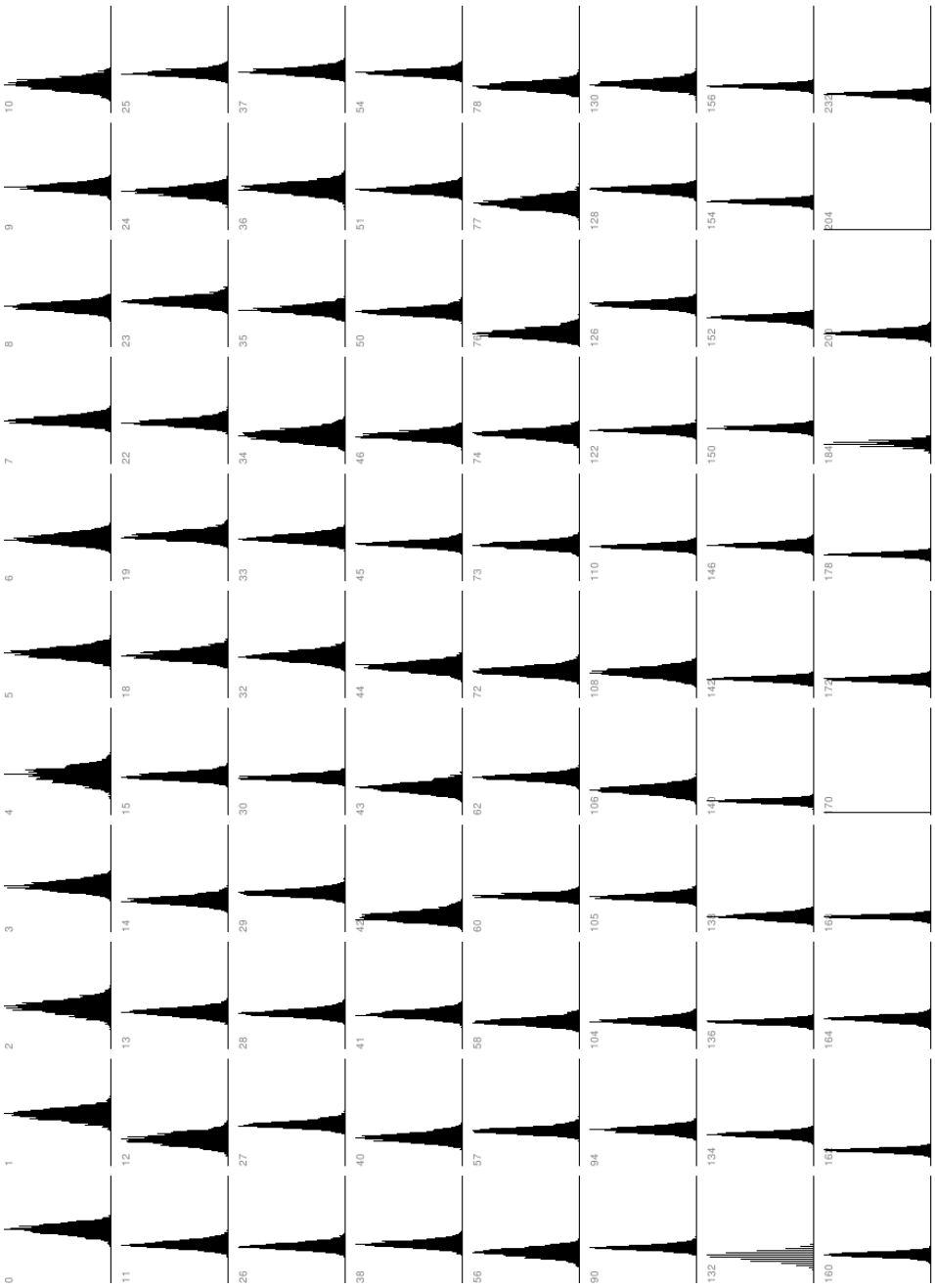


FIGURE 10.9. Histograms of transition Levenshtein distances for the 88 essentially different ECAs, on 100 cells, from 2500 samples.

CHAPTER 11

Discussion

We have explored several global properties of cellular automata, including numbers of automorphisms of transition graphs, numbers of preimages for homogeneous configurations, and distances between successive configurations. We have focused almost exclusively on the elementary CAs, although many of the methods and results should generalise readily to CAs with larger state sets, neighbourhoods, and/or dimensionality.

The aim of this thesis, as stated in Chapter 1, was to investigate what conditions might be necessary and/or sufficient for a CA to fall into each of Wolfram's classes, the idea being that this may give insight into why those classes of behaviour occur. The results in Chapter 7 suggest that chaotic (class 3) behaviour of the type that creates, rather than merely preserves, randomness arises from a lack of symmetry, whereas simpler (class 1 and 2) behaviour arises from an abundance of symmetry. The relationship between symmetry and class 4 behaviour remains unclear.

Wolfram's classes are defined in terms of long-term behaviour, whereas preimages and transition distances are distinctly short-term phenomena. In light of this, perhaps it is not so surprising that those properties are not good indicators of Wolfram class. This seems to suggest that classification requires not just *spatial* globality (considering the entire lattice or the entire configuration space), but also *temporal* globality (considering entire trajectories rather than single transitions). Properties of transition graphs and distances of multiple transitions (Section 10.5) are spatially and temporally global, and seem to be the most promising indicators of Wolfram class. Properties such as the entropy measures of Section 5.3 are temporally global but spatially local, whereas Langton's λ parameter (Section 5.1) is both spatially and temporally local.

A recurring difficulty in studying global properties of the CA is the fact that the size of the configuration space $S^{\mathbb{L}}$ grows exponentially with the size of the lattice $\mathbb{L}$. As a result, we are only able to compute for relatively small lattices, and even an orders-of-magnitude increase in computing resources only allows an incremental increase in lattice size. However, this thesis demonstrates some ways to circumvent this limitation:

(1) De Bruijn matrices (Section 4.3) allow the preimages of a configuration to be counted in linear time.
(2) The reverse algorithm (Section 4.1) lists preimages of a configuration. The time complexity is exponential in the worst case, but this worst case is relatively rare.
(3) The distribution of transition distances can be approximated by sampling over the configuration space (Section 10.1). In the case of Hamming distances, the moments of the distribution can be found by a combination of computational and algebraic methods (Section 10.2), allowing us to find their limits as the number of cells tends to infinity.

There are other possibilities in the special case of linear CAs:

(1) A modified version of the reverse algorithm (Section 4.2) lists preimages of a configuration in linear time, even in the worst case.
(2) The transition graphs can often be described completely without constructing them (Section 6.4).
(3) Following from the above, the automorphisms of the transition graph can be counted much more efficiently than in the general case (Section 8.1).

There seems to be a common theme to these optimisations: they either only work for "simple" CAs (such as linear CAs or class 2 CAs), or they allow the computation of "simple" properties that have little correlation with Wolfram class. It seems logical that complex (class 3 or 4) behaviour should be irreducible, and equally logical that CAs with such behaviour should not admit reductions in the amount of computation required for their analysis. Indeed, if this notion could be given a more rigorous foundation, it may itself form the basis of a classification.

Obtaining a complete formal analogue of Wolfram's classification remains an open problem. Such a classification would be useful, not so much for the ability to answer the question of which class a given CA is in, but to answer *why* the CA is in that class. For this reason, approaches to classification such as evolutionary computation or pattern recognition are not as attractive as they may appear at first. If such a classification can be found, it may give us insight into why some systems have more complex emergent behaviour than others, and perhaps how to engineer systems that exhibit such behaviour.

In summary, the main contributions of this thesis are as follows:

(1) A method for counting the automorphisms of a CA's transition graph (Section 7.3). Indeed, this method can be applied to any functional graph.
(2) A more efficient method for counting automorphisms for ECA rule 90 (Theorem 8.5), which may be applicable to other linear CAs.
(3) The suggestion of links between symmetry and complexity:
    (a) Certain chaotic CAs exhibit fewer automorphisms, and a slower increase in number of automorphisms as the number of cells is increased, than other CAs.
    (b) The number of automorphisms for linear CAs is more sensitive to changes in the number of cells than that for nonlinear CAs.
(4) Two methods for counting preimages of homogeneous configurations of 1-dimensional CAs (Chapter 9). One of these methods can also be applied to heterogeneous but spatially periodic configurations (Section 9.3).
(5) A method for calculating moments of distributions of transition Hamming distances for 1-dimensional CAs (Section 10.2).
(6) A conjecture that, in the limit as the number of cells approaches infinity, the overall distribution of transition distances has the same shape as the underlying distribution of the configuration space (Conjecture 10.18).

Appendix

APPENDIX A

Mathematical prerequisites

A.1. Magmas, semigroups, monoids and groups

Two excellent references on this material are Fraleigh [Fra03] and Rotman [Rot00].

Definition A.1. Let X be a set. A *binary operation* $*$ on X is a function from X^2 to X. We write $*$ using infix notation; i.e. we write $x*y$ for $*(x, y)$.

Definition A.2. Let X be a set, and let $*$ be a binary operation on X. Then the pair $(X, *)$ is a *magma* or X is a *magma under* $*$. Where the operation $*$ is clear from context, we simply refer to the magma as X. The *order* of the magma is the number of elements in X.

Definition A.3. Let $(S, *)$ be a magma. We say that $(S, *)$ is a *semigroup* if it satisfies the *associative property*:

$$a * (b * c) = (a * b) * c \text{ for all } a, b, c \in S. \tag{A.1}$$

Example A.4. Let X be a set. The set of all functions from X to itself forms a semigroup under composition of functions. ◊

Definition A.5. Let $(M, *)$ be a semigroup. We say that $(M, *)$ is a *monoid* if it has an *identity element*: an element $e \in M$ such that

$$e * a = a = a * e \text{ for all } a \in M. \tag{A.2}$$

Example A.6. Let A be a non-empty set. The set of all words (in the sense of formal language theory) over A forms a monoid under concatenation; the identity element is the empty word. ◊

Definition A.7. Let $(G, *)$ be a monoid with identity element e. We say that $(G, *)$ is a *group* if every element has an *inverse*: for every $a \in G$, there exists $a^{-1} \in G$ such that

$$a * a^{-1} = e = a^{-1} * a. \tag{A.3}$$

Example A.8. The non-zero real numbers form a group under multiplication. The identity element is 1, and the inverse of x is $\frac{1}{x}$. The same is true for the non-zero rationals and complex numbers.

The real numbers form a group under addition. The identity element is 0, and the inverse of x is $-x$. The same is true for the integers, rationals and complex numbers.

The $n \times n$ matrices with real entries and non-zero determinant form a group under multiplication. ◊

Definition A.9. Let $(X, *)$ be a magma, semigroup, monoid or group. We say that $(X, *)$ is *commutative* if

$$a * b = b * a \text{ for all } a, b \in X. \tag{A.4}$$

In the case of a group, we use the word *abelian* instead of commutative.

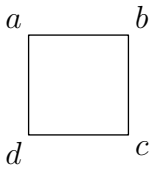
Example A.10. The groups described in Example A.8 (1) and (2) are abelian, but the group described in Example A.8 (3) is not. ◊

Definition A.11. Let G be a group under $*$, and let X be a subset of G. We say that X is a set of *generators* for G if, for every $g \in G$, we have

$$g = x_1^{\lambda_1} * x_2^{\lambda_2} * \cdots * x_m^{\lambda_m} \tag{A.5}$$

for some $x_i \in X$ and $\lambda_i = \pm 1$. In other words, every element of G can be written as a combination of elements of X and their inverses.

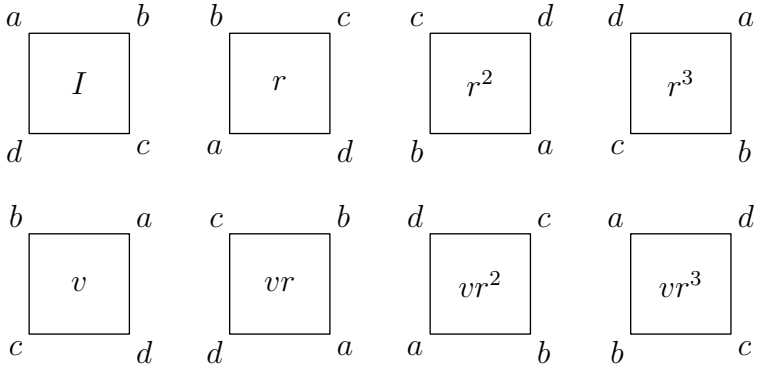
Example A.12 (Symmetries of the square)**.** Consider the square



A *symmetry* of the square is a geometrical transformation which maps the square onto itself, leaving the shape and position unchanged. There are eight such symmetries:

$$I, r, r^2, r^3, v, vr, vr^2, vr^3 \tag{A.6}$$

where I is the identity, r is an anticlockwise rotation through $90°$ about the centre of the square, and v is a reflection in the vertical axis through the centre of the square.



It is easy to see that

$$\mathcal{D}_4 = \left\{I, r, r^2, r^3, v, vr, vr^2, vr^3\right\} \tag{A.7}$$

is a group under composition of functions. A set of generators for $\mathcal{D}_4$ is $\{r, v\}$. ◊

A.2. Relations

Definition A.13. Let X be a set. A *relation* $\sim$ on X is a subset of X^2. We generally write $x \sim y$ to denote that (x, y) is an element of this subset.

Example A.14. The symbols $=, <, >, \leq, \geq, \neq, \approx$ all denote relations on sets of numbers. ◊

Definition A.15. An *equivalence relation* on a set X is a relation $\sim$ with the following properties:

Reflexivity: $x \sim x$ for all $x \in X$;
Symmetry: If $x \sim y$, then $y \sim x$;
Transitivity: If $x \sim y$ and $y \sim z$, then $x \sim z$.

Example A.16. The motivating example of an equivalence relation is $=$ (equality). Another example on the integers is congruence modulo n. ◊

Definition A.17. Let $\sim$ be an equivalence relation on a set X, and let $x \in X$. The *$\sim$-class* of x is the set of all elements related to x by $\sim$:

$$\{y \in X \ : \ x \sim y\}. \tag{A.8}$$

Example A.18. In the integers, the "congruence modulo 4"-class of 3 is

$$\{\ldots, -5, -1, 3, 7, 11, \ldots\} \tag{A.9}$$

◊

Remark A.19. Two $\sim$-classes are either disjoint or equal. Thus the distinct $\sim$-classes of X give a *partitioning* of X; that is, a family of subsets such that every element of X is in exactly one subset.

A.3. Group actions and orbit counting

Definition A.20. Let G be a group, and let X be a set. Define a function $\cdot : G \times X \to X$. Say that G *acts on* X if the following two conditions hold:

(1) $e \cdot x = x$ for all $x \in X$, where e is the identity in G;
(2) $(g * h) \cdot x = g \cdot (h \cdot x)$ for all $x \in X$ and all $g, h \in G$.

Say that g *fixes* x, or x is *fixed by* g, if $g \cdot x = x$.

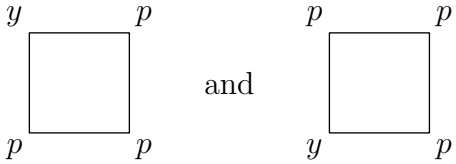
Example A.21. Let G be a group of functions from X to X, under composition of functions. Then G acts on X, with the operation of function application. ◊

Example A.22. Affix a purple or yellow bead to each vertex of a square. The set of all configurations of beads is $\{p, y\}^4$. The group $\mathcal{D}_4$ of symmetries of the square acts on this set, with $\cdot$ defined in the natural way. ◊

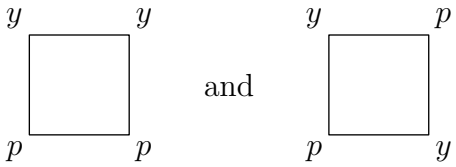
Definition A.23. Let group G act on set X. Two elements $x, y \in X$ are *G-equivalent*, denoted $x \sim_G y$, if $y = g \cdot x$ for some $g \in G$.

Remark A.24. It follows from the definition of a group that $\sim_G$ is an equivalence relation.

Example A.25. Following on from Example A.22, the configurations



are G-equivalent, as the second is obtained from the first by $r \in \mathcal{D}_4$. However, the configurations



are not G-equivalent, as there is no element of $\mathcal{D}_4$ that transforms the first into the second. ◊

Definition A.26. Let group G act on set X. The *orbits* of X are the $\sim_G$-classes of X.

Theorem A.27 (The orbit counting theorem)**.** *Let group G act on set X. The number of distinct orbits among the elements of X is*

$$\frac{1}{|G|} \sum_{g \in G} \chi(g)\,, \tag{A.10}$$

where

$$\chi(g) = |\{x \in X \ : \ g \cdot x = x\}| \tag{A.11}$$

is the number of elements of X fixed by g.

This result is often incorrectly attributed to Burnside; as a result, some authors refer to it as "the lemma that is not Burnside's". It is more correctly attributed to Cauchy or Frobenius.

Example A.28. Following on from Example A.22, how many essentially different configurations of beads are there?

All elements of X are fixed by the identity transformation, so $\chi(I) = 16$.

For a configuration to be fixed by r, the bead at a must be the same colour as the bead at b, which must be the same colour as the bead at c, and so on. In other words, all beads must be the same colour. Thus $\chi(r) = 2$. A similar argument shows that $\chi(r^3) = 2$.

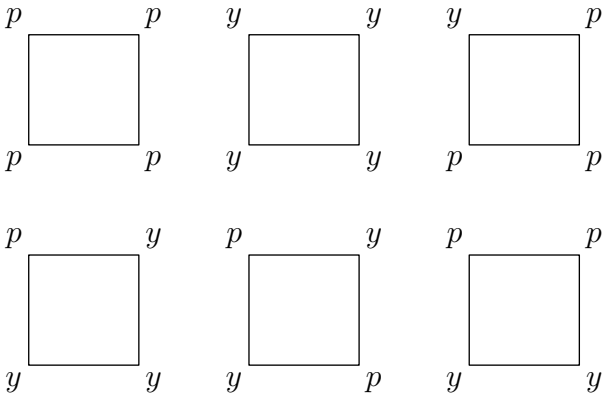
For a configuration to be fixed by r^2, bead a must be the same colour as bead c, and bead b the same colour as bead d. Thus $\chi(r^2) = 4$. A similar argument shows that $\chi(v) = \chi(vr^2) = 4$.

For a configuration to be fixed by vr, bead a must be the same colour as bead c; there are no restrictions on beads b and d. Thus $\chi(vr) = 8$. A similar argument shows that $\chi(vr^3) = 8$.

Substituting into Equation A.11 gives

$$\frac{16 + 2 \times 2 + 3 \times 4 + 2 \times 8}{8} = 6 \tag{A.12}$$

different configurations. In this example, it is easy to see that these configurations are



◊

A.4. Rings and fields

Definition A.29. Let R be a set, and let $+$ and $\times$ be binary operations on R. We say R is a *commutative ring* if:

(1) $(R, +)$ is an *abelian group* (Definitions A.7 and A.9):
   (a) $a + (b + c) = (a + b) + c$ for all $a, b, c \in R$;
   (b) $a + b = b + a$ for all $a, b \in R$;
   (c) There exists $0 \in R$ with $a + 0 = a$ for all $a \in R$;
   (d) For each $a \in R$ there exists $-a \in R$ such that $a + (-a) = 0$.

(2) $(R, \times)$ is a *commutative monoid* (Definitions A.5 and A.9):
   (a) $a \times (b \times c) = (a \times b) \times c$ for all $a, b, c \in R$;
   (b) $a \times b = b \times a$ for all $a, b \in R$;
   (c) There exists $1 \in R$ with $a \times 1 = a$ for all $a \in R$.

(3) $\times$ *distributes* over $+$: $a \times (b + c) = (a \times b) + (a \times c)$ for all $a, b, c \in R$.

Definition A.30. We say that a ring R is a *field* if every element in $R \setminus \{0\}$ has a multiplicative inverse; that is, for each $a \in R \setminus \{0\}$ there exists $a^{-1} \in R \setminus \{0\}$ such that $a \times a^{-1} = 1$.

Example A.31. The motivating example of a commutative ring is the set of integers under the usual operations of addition and multiplication; examples of fields include the rationals and reals under addition and multiplication.

An example of a finite commutative ring is the set $\mathbb{Z}_k$ of integers modulo k, under the usual operations of addition and multiplication modulo k. Furthermore, $\mathbb{Z}_p$ is a field if p is prime. ◊

A.5. Finite rings of polynomials

This appendix describes the algebraic construction of the commutative rings of polynomials used in Section 3.2.

Definition A.32. Let R be a commutative ring, and let $a \in R$. The *principal ideal* generated by a, denoted $\langle a \rangle$, is

$$\langle a \rangle = aR = \{ar \ : \ r \in R\}. \tag{A.13}$$

Definition A.33. Let R be a commutative ring, and let $\langle a \rangle$ be a principal ideal of R. The *factor ring* of R by $\langle a \rangle$, denoted $R/\langle a \rangle$, is

$$R/\langle a \rangle = \{r + \langle a \rangle \ : \ r \in R\}, \tag{A.14}$$

where

$$r + \langle a \rangle = \{r + s \ : \ s \in \langle a \rangle\}. \tag{A.15}$$

Definition A.34. Let F be a field. We denote by $F[x]$ the set of all polynomials with coefficients in F:

$$F[x] = \left\{ a_0 + a_1 x + \cdots + a_k x^k \ : \ k \geq 0, a_i \in F \right\}. \tag{A.16}$$

Let F be a field. It can be shown that $F[x]$ is a commutative ring. Choose a positive integer n, and consider the factor ring

$$R_n^F = F[x]/\langle x^n - 1\rangle. \tag{A.17}$$

An element of R_n^F has the form

$$f(x) + \langle x^n - 1\rangle = \{f(x) + h(x) \ : \ h(x) \in \langle x^n - 1\rangle\} \tag{A.18}$$

We shall now use the following theorem:

Theorem A.35. *Let F be a field, and let $f(x), g(x) \in F[x]$ with $g(x) \neq 0$. Then there exist unique polynomials $q(x), r(x) \in F[x]$, with $r = 0$ or $\deg r < \deg g$, such that*

$$f(x) = q(x)g(x) + r(x). \tag{A.19}$$

The polynomials $q(x)$ and $r(x)$ are the *quotient* and *remainder* resulting from division of $f(x)$ by $g(x)$.

Let us apply Theorem A.35 to $f(x)$ in (A.18), setting $g(x) = x^n - 1$. We have

$$f(x) = (x^n - 1)q(x) + r(x), \tag{A.20}$$

where $r(x) = 0$ or $\deg r < \deg x^n - 1 = n$. But $(x^n - 1)q(x) \in \langle x^n - 1\rangle$, so we have

$$f(x) + \langle x^n - 1\rangle = r(x) + \langle x^n - 1\rangle. \tag{A.21}$$

This shows that the factor ring R_n^F has the form

$$\{r(x) + \langle x^n - 1\rangle \ : \ r(x) \in F[x], r(x) = 0 \text{ or } \deg r < n\} \tag{A.22}$$

So the factor ring is isomorphic to the set of all polynomials of degree less than n, and thus is finite if F is finite. Addition and multiplication of polynomials is carried out in the usual way, but when polynomials of degree n or more appear they are replaced with their remainder under division by $\langle x^n - 1\rangle$. It turns out that this is exactly the same as simply replacing all powers of x with their residue modulo n (i.e. replacing x^{an+b} with x^b):

Lemma A.36. *Let a and b be nonnegative integers, with $b < n$. Then*

$$x^{an+b} + \langle x^n - 1\rangle = x^b + \langle x^n - 1\rangle. \tag{A.23}$$

Proof. First, note that

$$x^{an+b} = ((x^n - 1) + 1)^a \, x^b, \tag{A.24}$$

Expanding the binomial $((x^n - 1) + 1)^a$ term gives

$$x^{an+b} = ((x^n - 1)q(x) + 1) \, x^b \tag{A.25}$$

for some polynomial $q(x)$. So we have

$$x^{an+b} = (x^n - 1)q'(x) + x^b \tag{A.26}$$

for some polynomial $q'(x) = x^b q(x)$. Now $(x^n - 1)q'(x) \in \langle x^n - 1 \rangle$, so the result (A.23) follows. □

Example A.37. Set $F = \mathbb{Z}_2$ and $n = 4$, and consider the polynomials used in Example 3.3:

$$f(x) = x + x^2 + x^3 \tag{A.27}$$

$$g(x) = 1 + x^2. \tag{A.28}$$

Working in $\mathbb{Z}_2[x]$, we have

$$f(x) \times g(x) = (x + x^2 + x^3) \times (1 + x^2) \tag{A.29}$$

$$= (x + x^2 + x^3) + (x^3 + x^4 + x^5) \tag{A.30}$$

$$= x + x^2 + x^4 + x^5. \tag{A.31}$$

Now consider the corresponding elements in $R_4^{\mathbb{Z}_2}$:

$$(f(x) + \langle x^4 - 1 \rangle) \times (g(x) + \langle x^4 - 1 \rangle) \tag{A.32}$$

$$= [f(x) \times g(x)] + \langle x^4 - 1 \rangle \tag{A.33}$$

$$= [x + x^2 + x^4 + x^5] + \langle x^4 - 1 \rangle \tag{A.34}$$

$$= [1 + x^2 + 1 + x + x^4 + x^5] + \langle x^4 - 1 \rangle \tag{A.35}$$

$$= [1 + x^2 + (1 + x)(1 + x^4)] + \langle x^4 - 1 \rangle \tag{A.36}$$

$$= [1 + x^2] + \langle x^4 - 1 \rangle \tag{A.37}$$

since $(1 + x)(1 + x^4) = (1 + x)(x^4 - 1) \in \langle x^4 - 1 \rangle$. This is the same as the answer obtained in Example 3.3. ♢

A.6. Metric spaces and topology

Definition A.38. Let M be a set. A function $d : M^2 \to \mathbb{R}$ is a *metric* on M if:

(1) $d(x, y) \geq 0$ for all $x, y \in M$;
(2) $d(x, y) = 0$ if and only if $x = y$;
(3) $d(x, y) = d(y, x)$ for all $x, y \in M$;
(4) $d(x, z) \leq d(x, y) + d(y, z)$ for all $x, y, z \in M$ (the *triangle inequality*).

If d is a metric on M, we say that (M, d) is a *metric space*.

A metric is a measure of "distance" between any two points in a set, satisfying the properties we might expect of such a measure.

Example A.39. The *Euclidean metric* d_e on $\mathbb{R}^D$ is defined by

$$d_e((x_1,\ldots,x_D),(y_1,\ldots,y_D)) = \sqrt{(y_1-x_1)^2+\cdots+(y_D-x_D)^2}\,. \quad \text{(A.38)}$$

This is simply the familiar measure of distance between points in D-dimensional space, i.e. the D-dimensional analogue of Pythagoras' Theorem. ◊

Example A.40. The *Manhattan metric* d_m on $\mathbb{R}^D$ is defined by

$$d_m((x_1,\ldots,x_D),(y_1,\ldots,y_D)) = |y_1-x_1|+\cdots+|y_D-x_D|\,. \quad \text{(A.39)}$$

The Manhattan metric measures the shortest distance between two points when travelling parallel to the coordinate axes. The name arises from the fact that the $D=2$ case measures the distance between two places in a city whose roads follow a grid pattern. ◊

Example A.41. For a finite nonempty set X, the *Hamming distance* d_h is a metric on X^D defined by

$$d_h(x_1\ldots x_D, y_1\ldots y_D) = |\{i\in\{1,\ldots,D\} \;:\; x_i\neq y_i\}|\,. \quad \text{(A.40)}$$

In other words, the Hamming distance is the number of positions at which the strings $x_1\ldots x_D$ and $y_1\ldots y_D$ differ. ◊

Remark A.42. If $L\subseteq M$ and d is a metric on M, then d is also a metric on L. In particular, the Euclidean and Manhattan metrics are also metrics on $\mathbb{Z}^D$, and on subsets (finite or infinite) thereof.

Definition A.43. Let (M,d) be a metric space, and let $x_1,x_2,x_3,\ldots$ be an infinite sequence of elements of M. Say that x is the *limit* of the sequence if x_i grows arbitrarily close to x as i tends to infinity: for all $\varepsilon>0$, there exists n such that $d(x_i,x)<\varepsilon$ for all $i\geq n$.

Definition A.44. Let M be a metric space, and $X\subseteq M$. Say X is *dense* in M if every point in M is the limit of a sequence in X.

Example A.45. The set of rational numbers $\mathbb{Q}$ is dense in $\mathbb{R}$, since every real number is the limit of a sequence of rationals. ◊

Definition A.46. Let (M,d) be a metric space, let $x\in M$ and $r\in\mathbb{R}$. The *open ball* centred at x of radius r, denoted $B(x,r)$, is the set of points in M a distance less than r from x:

$$B(x,r) = \{y\in M \;:\; d(x,y)<r\} \quad \text{(A.41)}$$

Definition A.47. Let (M,d) be a metric space, and let $X\subseteq M$. Denote by X^c the *complement* of X, defined as the set of those elements of M that are not in X.

(1) A point $x \in M$ is an *interior point* of X if there exists $r > 0$ such that $B(x, r) \subseteq X$.
(2) A point $x \in M$ is an *exterior point* of X if there exists $r > 0$ such that $B(x, r) \subseteq X^c$.
(3) A point $x \in M$ is a *boundary point* of X if it is neither an interior nor an exterior point of X.

Note that X contains all its interior points, but none of its exterior points. It may contain none, some or all of its boundary points.

Definition A.48. Let (M, d) be a metric space, and let $X \subseteq M$.

(1) Say that X is *open* if it does not contain any of its own boundary points; that is, if every point in X is an interior point.
(2) Say that X is *closed* if it contains all of its own boundary points.
(3) Say that X is *clopen* if it is both closed and open, i.e. if it has no boundary points.

Example A.49. Consider the metric space $\mathbb{R}$ with the Euclidean metric.

(1) The subset $\{x \in \mathbb{R} \ : \ 0 < x < 1\}$ is open;
(2) The subset $\{x \in \mathbb{R} \ : \ 0 \leq x \leq 1\}$ is closed;
(3) The subset $\{x \in \mathbb{R} \ : \ 0 \leq x < 1\}$ is neither open nor closed;
(4) The "subsets" $\emptyset$ and $\mathbb{R}$ are clopen. (In general, for any metric space M, the "subsets" $\emptyset$ and M are clopen.)

In each of the first three cases, the set of interior points is $\{x \in \mathbb{R} \ : \ 0 < x < 1\}$ and the set of boundary points is $\{0, 1\}$. ◊

Theorem A.50. *In a metric space, the union of any (finite or infinite) collection of open sets is also an open set. The intersection of any* finite *collection of open sets is an open set; the intersection of an infinite collection of open sets may or may not be open.*

Definition A.51. Let X be a set, and let T be a collection of subsets of X. Then X is a *topological space*, with *topology* T, if the following conditions hold:

(1) $\emptyset, X \in T$;
(2) The union of any subset (finite or infinite) of T is also in T;
(3) The intersection of any *finite* subset of T is also in T.

The sets in T are called the *open sets* of X.

Example A.52. Every metric space is a topological space, with open sets as in Definition A.48. ◊

Definition A.53. The *discrete topology* on a set X is defined by letting every subset of X be open, so that the collection T consists of all subsets of X.

Note that the discrete topology can be defined for *any* set X.

A.7. Graph theory

Definition A.54. A *(directed) graph* or *digraph* is a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of *vertices* or *nodes*, and $\mathcal{E} \subseteq \mathcal{V}^2$ is the set of *edges*. An edge directed from vertex x to vertex y is represented by an element $(x, y) \in \mathcal{E}$, and is often denoted $x \to y$.

Definition A.55. A *bipartite graph* is a graph whose vertices can be partitioned into two sets L and R, such that the edge set is a subset of $L \times R$ (i.e. all edges are directed from a vertex in L to a vertex in R).

Definition A.56. Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V} = \{v_1, \ldots, v_n\}$. The *adjacency matrix* A of $\mathcal{G}$ is the $n \times n$ matrix defined by

$$A_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in \mathcal{E} \\ 0 & \text{otherwise,} \end{cases} \tag{A.42}$$

where A_{ij} denotes the entry in row i, column j of A.

Definition A.57. If $\mathcal{G} = (L \cup R, \mathcal{E})$ is a bipartite graph with $L = \{l_1, \ldots, l_n\}$ and $R = \{r_1, \ldots, r_m\}$, the adjacency matrix A is alternatively defined as the $n \times m$ matrix with

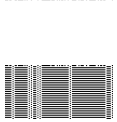
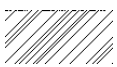
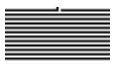
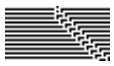
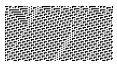
$$A_{ij} = \begin{cases} 1 & \text{if } (l_i, r_j) \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases} \tag{A.43}$$

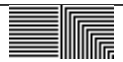
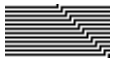
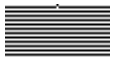
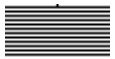
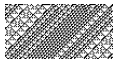
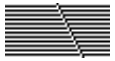
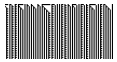
APPENDIX B

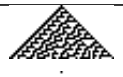
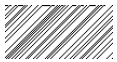
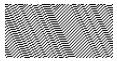
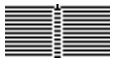
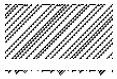
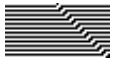
Table of elementary cellular automata

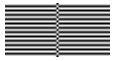
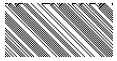
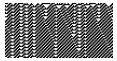
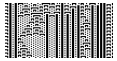
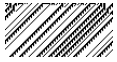
This table lists the 88 essentially different ECAs, choosing the rule with the smallest number from each equivalence class. The table shows the following information:

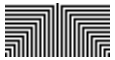
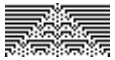
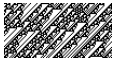
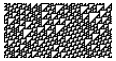
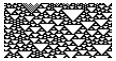
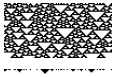
(1) The rule number in decimal notation.
(2) The rule number in binary notation.
(3) The rule's image under the reflection transformation (Equation 2.11). If the rule is its own reflection (so that reflection is a symmetry in the sense of Definition 7.3), a box is drawn around the rule number.
(4) As (3), but for the conjugation transformation (Equation 2.10).
(5) As (3), but for the composition of the reflection and conjugation transformations.
(6) If the rule is linear (Chapter 3), or linear plus a constant term, the "Linear" column gives an expression for $f(x, y, z)$. If the rule is nonlinear, this column is blank.
(7) Space-time diagram showing 25 steps of the evolution from an initial configuration assigning state 1 to a single cell.
(8) Space-time diagram showing 50 steps of the evolution from a randomly generated initial configuration, on the periodic lattice $\mathbb{Z}_{100}$. The same random initial configuration is used for each rule.
(9) The value of Langton's λ parameter (Section 5.1) obtained by choosing 0 as the quiescent state. If choosing 0 yields the value λ, choosing 1 yields $1 - \lambda$.
(10) The value of Wuensche's Z parameter (Section 5.2).
(11) The CA's Wolfram class, determined as described in Section 2.5. Due to the subjective nature of Wolfram's classification, this should not be considered as authoritative, but merely as an indication.
(12) Any remarks or references into the text.

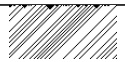
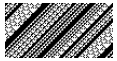
Rule number		Equivalent rules			Linear	Space-time diagrams		Parameters		Wolfram	
Dec.	Binary	Refl.	Conj.	Both	$f(x,y,z)$	Single	Random	λ	Z	class	Remarks
0	00000000	0	255	255	0			0.0	0.0	1	Zero rule
1	00000001	1	127	127				0.125	0.25	2	
2	00000010	16	191	247				0.125	0.25	2	
3	00000011	17	63	119				0.25	0.5	2	
4	00000100	4	223	223				0.125	0.25	2	
5	00000101	5	95	95				0.25	0.5	2	
6	00000110	20	159	215				0.25	0.5	2	
7	00000111	21	31	87				0.375	0.75	2	
8	00001000	64	239	253				0.125	0.25	1	
9	00001001	65	111	125				0.25	0.5	2	
10	00001010	80	175	245				0.25	0.5	2	
11	00001011	81	47	117				0.375	0.75	2	
12	00001100	68	207	221				0.25	0.5	2	

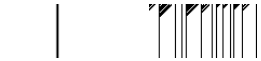
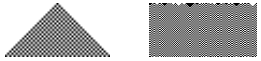
Rule number		Equivalent rules			Linear	Space-time diagrams		Parameters		Wolfram	
Dec.	Binary	Refl.	Conj.	Both	$f(x,y,z)$	Single	Random	λ	Z	class	Remarks
13	00001101	69	79	93				0.375	0.75	2	
14	00001110	84	143	213				0.375	0.75	2	
15	00001111	85	[15]	85	$x+1$			0.5	1.0	2	In $\mathcal{Z}$ (§7.4)
18	00010010	[18]	183	183				0.25	0.5	3	
19	00010011	[19]	55	55				0.375	0.625	2	
22	00010110	[22]	151	151				0.375	0.75	3	
23	00010111	[23]	[23]	[23]				0.5	0.5	2	
24	00011000	66	231	189				0.25	0.5	2	
25	00011001	67	103	61				0.375	0.75	2	
26	00011010	82	167	181				0.375	0.75	2	
27	00011011	83	39	53				0.5	0.75	2	
28	00011100	70	199	157				0.375	0.75	2	
29	00011101	71	71	[29]				0.5	0.5	2	

Rule number		Equivalent rules			Linear	Space-time diagrams		Parameters		Wolfram	
Dec.	Binary	Refl.	Conj.	Both	$f(x,y,z)$	Single	Random	λ	Z	class	Remarks
30	00011110	86	135	149				0.5	1.0	3	In $\mathcal{C}$ (§7.4); chaotic
32	00100000	32	251	251				0.125	0.25	1	
33	00100001	33	123	123				0.25	0.5	2	
34	00100010	48	187	243				0.25	0.5	2	
35	00100011	49	59	115				0.375	0.625	2	
36	00100100	36	219	219				0.25	0.5	2	
37	00100101	37	91	91				0.375	0.75	2	
38	00100110	52	155	211				0.375	0.75	2	
40	00101000	96	235	249				0.25	0.5	1	
41	00101001	97	107	121				0.375	0.75	2	
42	00101010	112	171	241				0.375	0.75	2	
43	00101011	113	43	113				0.5	0.5	2	
44	00101100	100	203	217				0.375	0.75	2	

Rule number		Equivalent rules			Linear	Space-time diagrams		Parameters		Wolfram	
Dec.	Binary	Refl.	Conj.	Both	$f(x,y,z)$	Single	Random	λ	Z	class	Remarks
45	00101101	101	75	89				0.5	1.0	3	In $\mathcal{C}$ (§7.4); chaotic
46	00101110	116	139	209				0.5	0.5	2	
50	00110010	50	179	179				0.375	0.625	2	
51	00110011	51	51	51	$y+1$			0.5	1.0	2	Exchanges 0 and 1
54	00110110	54	147	147				0.5	0.75	4	
56	00111000	98	227	185				0.375	0.75	2	
57	00111001	99	99	57				0.5	0.75	2	
58	00111010	114	163	177				0.5	0.75	2	
60	00111100	102	195	153	$x+y$			0.5	1.0	3	In $\mathcal{Z}$ (§7.4)
62	00111110	118	131	145				0.625	0.75	2	
72	01001000	72	237	237				0.25	0.5	2	
73	01001001	73	109	109				0.375	0.75	3	
74	01001010	88	173	229				0.375	0.75	2	

Rule number		Equivalent rules			Linear	Space-time diagrams		Parameters		Wolfram	
Dec.	Binary	Refl.	Conj.	Both	$f(x,y,z)$	Single	Random	λ	Z	class	Remarks
76	01001100	76	205	205				0.375	0.625	2	
77	01001101	77	77	77				0.5	0.5	2	
78	01001110	92	141	197				0.5	0.75	2	
90	01011010	90	165	165	$x+z$			0.5	1.0	3	In $\mathcal{Z}$ (§7.4); also §3.4, §8.2
94	01011110	94	133	133				0.625	0.75	2	
104	01101000	104	233	233				0.375	0.75	2	
105	01101001	105	105	105	$x+y+z+1$			0.5	1.0	3	In $\mathcal{Z}$ (§7.4)
106	01101010	120	169	225				0.5	1.0	3	In $\mathcal{C}$ (§7.4)
108	01101100	108	201	201				0.5	0.75	2	
110	01101110	124	137	193				0.625	0.75	4	Turing complete (§2.8)
122	01111010	122	161	161				0.625	0.75	3	
126	01111110	126	129	129				0.75	0.5	3	
128	10000000	128	254	254				0.125	0.25	1	

Rule number		Equivalent rules			Linear	Space-time diagrams		Parameters		Wolfram	
Dec.	Binary	Refl.	Conj.	Both	$f(x,y,z)$	Single	Random	λ	Z	class	Remarks
130	10000010	144	190	246				0.25	0.5	2	
132	10000100	132	222	222				0.25	0.5	2	
134	10000110	148	158	214				0.375	0.75	2	
136	10001000	192	238	252				0.25	0.5	1	
138	10001010	208	174	244				0.375	0.75	2	
140	10001100	196	206	220				0.375	0.625	2	
142	10001110	212	142	212				0.5	0.5	2	
146	10010010	146	182	182				0.375	0.75	3	
150	10010110	150	150	150	$x+y+z$			0.5	1.0	3	In $\mathcal{Z}$ (§7.4)
152	10011000	194	230	188				0.375	0.75	2	
154	10011010	210	166	180				0.5	1.0	2	In $\mathcal{Z}$ (§7.4)
156	10011100	198	198	156				0.5	0.75	2	
160	10100000	160	250	250				0.25	0.5	1	

Rule number		Equivalent rules			Linear	Space-time diagrams		Parameters		Wolfram	
Dec.	Binary	Refl.	Conj.	Both	$f(x,y,z)$	Single	Random	λ	Z	class	Remarks
162	10100010	176	186	242				0.375	0.75	2	
164	10100100	164	218	218				0.375	0.75	2	
168	10101000	224	234	248				0.375	0.75	1	
170	10101010	240	170	240	z			0.5	1.0	2	Left shift rule
172	10101100	228	202	216				0.5	0.75	2	
178	10110010	178	178	178				0.5	0.5	2	
184	10111000	226	226	184				0.5	0.5	2	
200	11001000	200	236	236				0.375	0.625	2	
204	11001100	204	204	204	y			0.5	1.0	2	Identity rule
232	11101000	232	232	232				0.5	0.5	2	

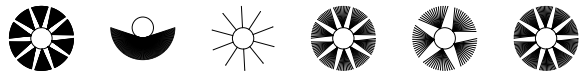
APPENDIX C

Transition graphs for ECAs on $\mathbb{Z}_{10}$

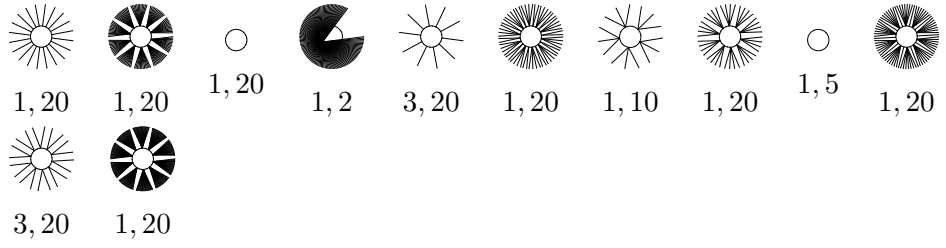
This appendix shows transition graphs for the 88 essentially different ECAs, excluding rule 0 and the identity rule 204, on the periodic lattice $\mathbb{Z}_{10}$. For each rule, the Wolfram class (determined as described in Section 2.5) and the longest transient length (maximum distance of a configuration from its cycle) are given. Below each basin is a pair m, p, where m is the number of isomorphic copies of this basin and p is the length of the cycle.

Rule 1 (class 2, max transient 1)

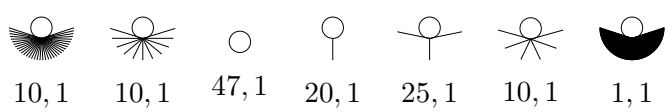
Rule 2 (class 2, max transient 1)



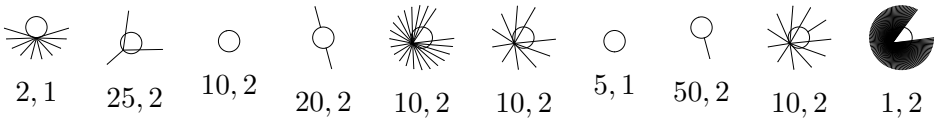
Rule 3 (class 2, max transient 1)



Rule 4 (class 2, max transient 1)

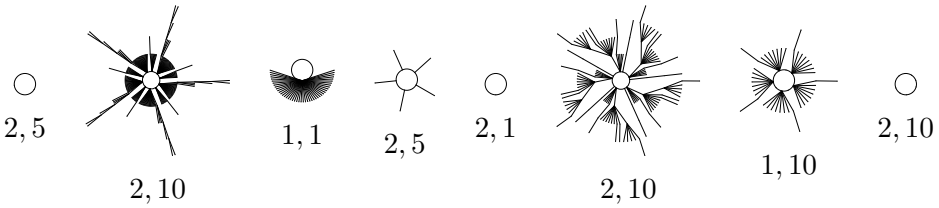


Rule 5 (class 2, max transient 1)

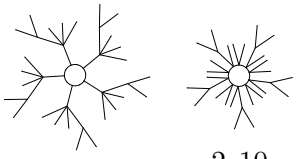


$2,1$ $25,2$ $10,2$ $20,2$ $10,2$ $10,2$ $5,1$ $50,2$ $10,2$ $1,2$ $10,1$

Rule 6 (class 2, max transient 4)

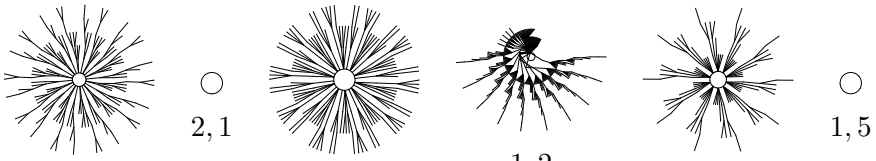


$2,5$ $2,10$ $1,1$ $2,5$ $2,1$ $2,10$ $1,10$ $2,10$



$2,10$ $2,10$

Rule 7 (class 2, max transient 9)

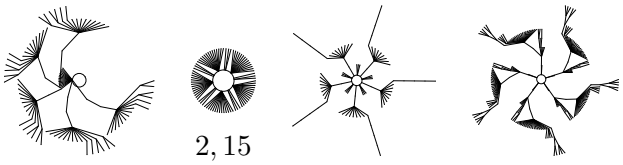


$1,20$ $2,1$ $1,20$ $1,2$ $1,20$ $1,5$

Rule 8 (class 1, max transient 2)

$1,1$

Rule 9 (class 2, max transient 8)

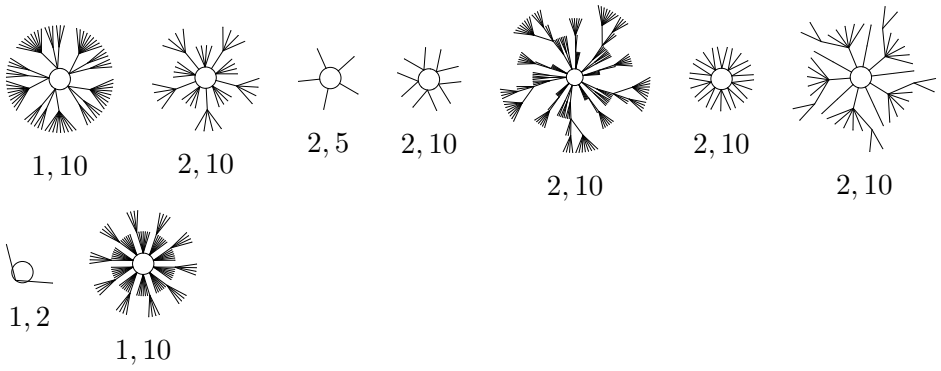


$1,2$ $2,15$ $2,10$ $2,5$

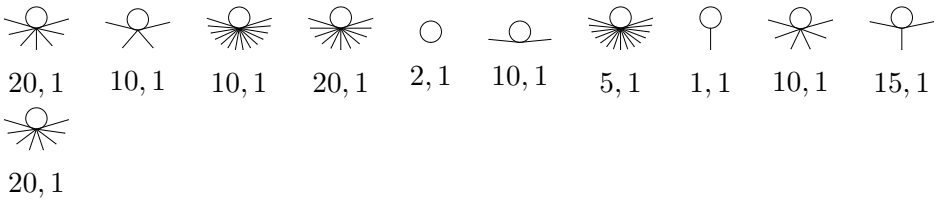
Rule 10 (class 2, max transient 1)

$1,5$ $2,10$ $1,1$ $6,10$ $1,5$ $3,10$

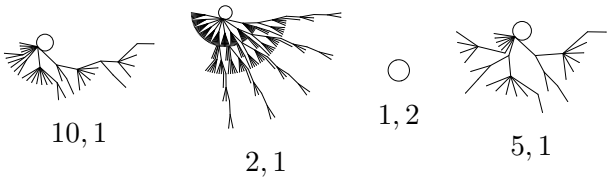
Rule 11 (class 2, max transient 4)



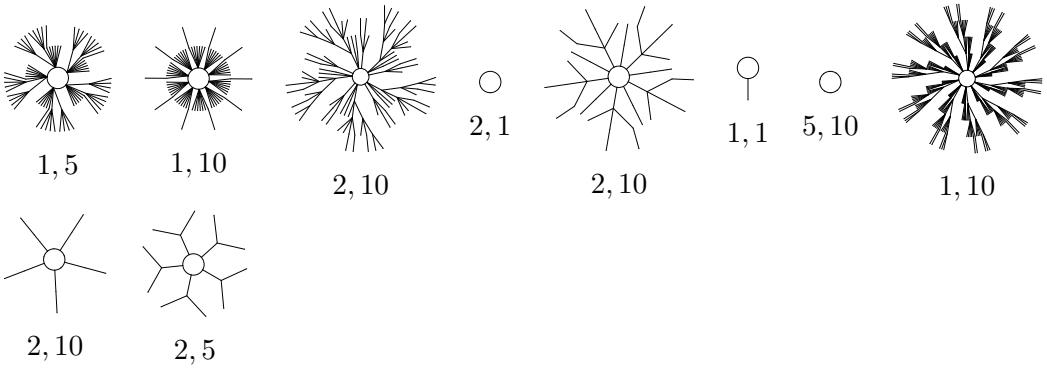
Rule 12 (class 2, max transient 1)



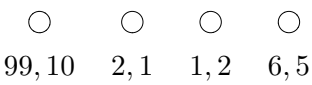
Rule 13 (class 2, max transient 8)



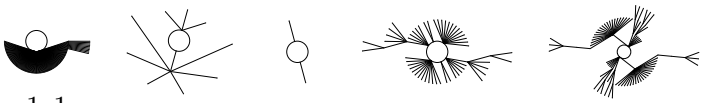
Rule 14 (class 2, max transient 4)



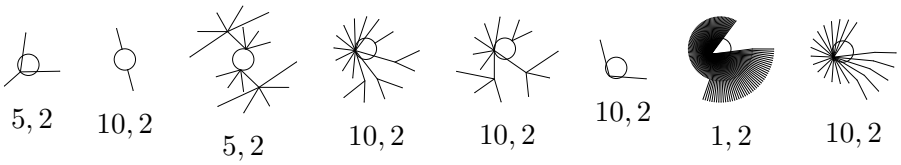
Rule 15 (class 2, max transient 0)



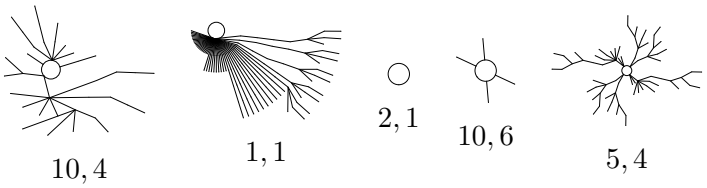
Rule 18 (class 3, max transient 5)



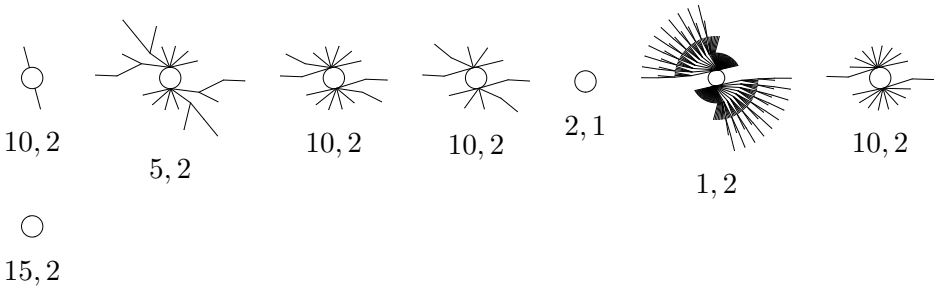
Rule 19 (class 2, max transient 2)



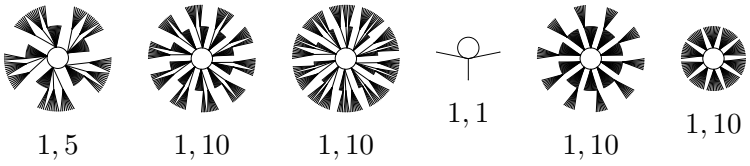
Rule 22 (class 3, max transient 7)



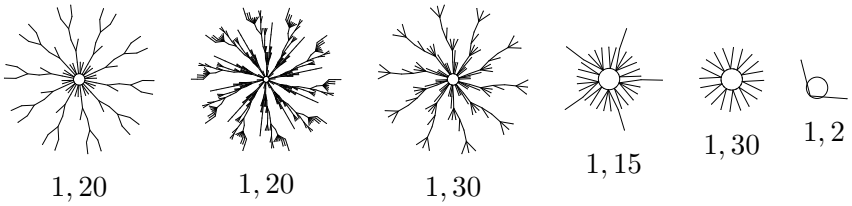
Rule 23 (class 2, max transient 4)



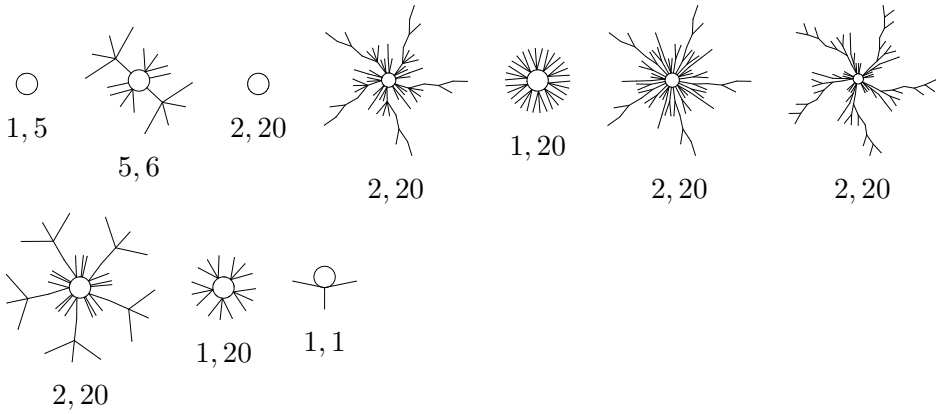
Rule 24 (class 2, max transient 2)



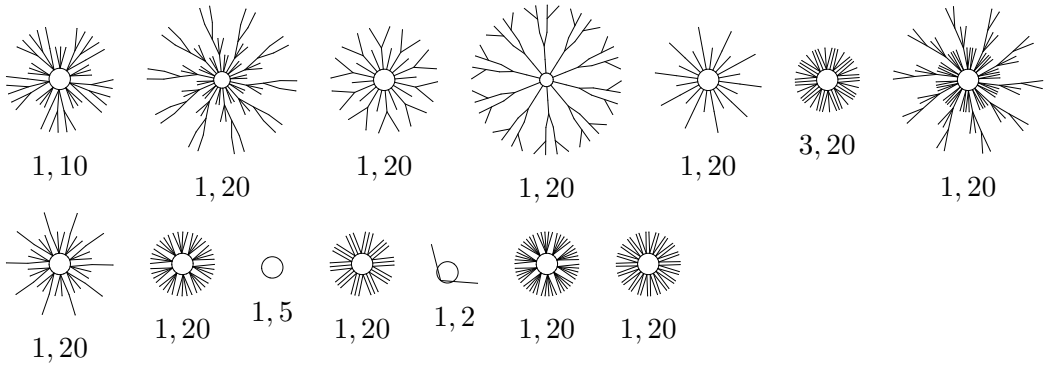
Rule 25 (class 2, max transient 12)



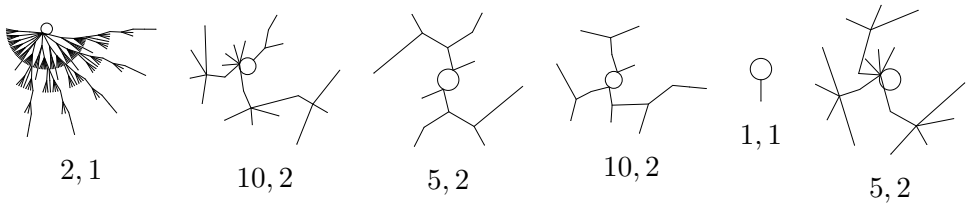
Rule 26 (class 2, max transient 7)



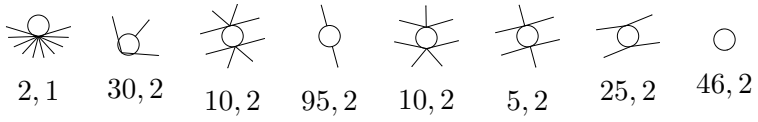
Rule 27 (class 2, max transient 5)



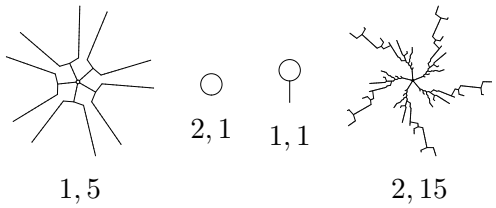
Rule 28 (class 2, max transient 9)



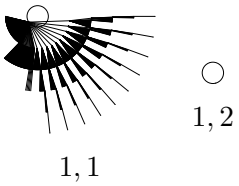
Rule 29 (class 2, max transient 1)



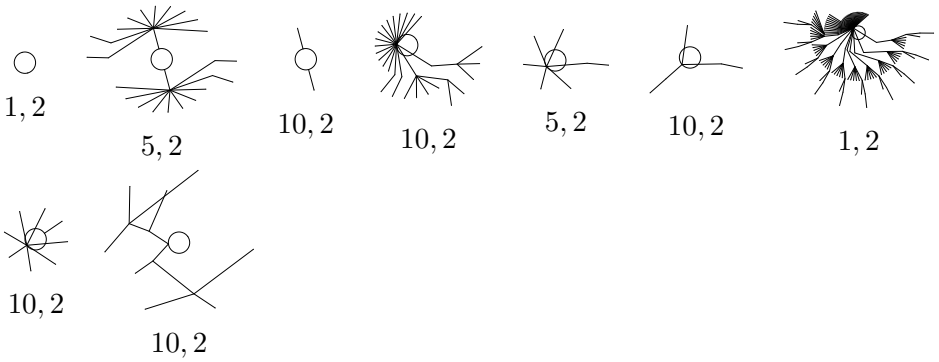
Rule 30 (class 3, max transient 46)



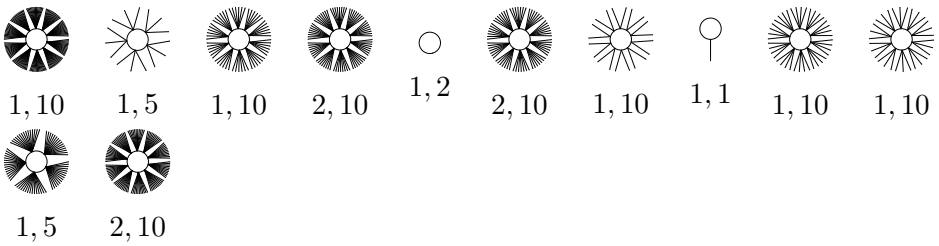
Rule 32 (class 1, max transient 5)



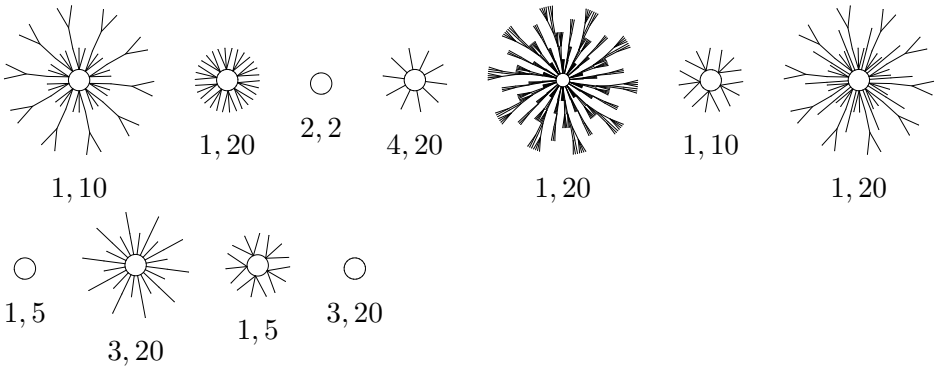
Rule 33 (class 2, max transient 5)



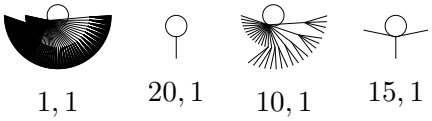
Rule 34 (class 2, max transient 1)



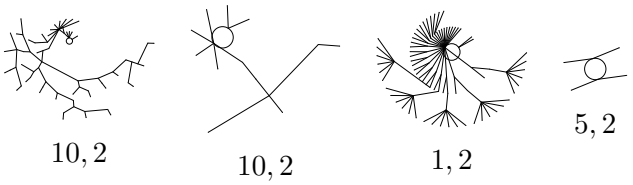
Rule 35 (class 2, max transient 5)



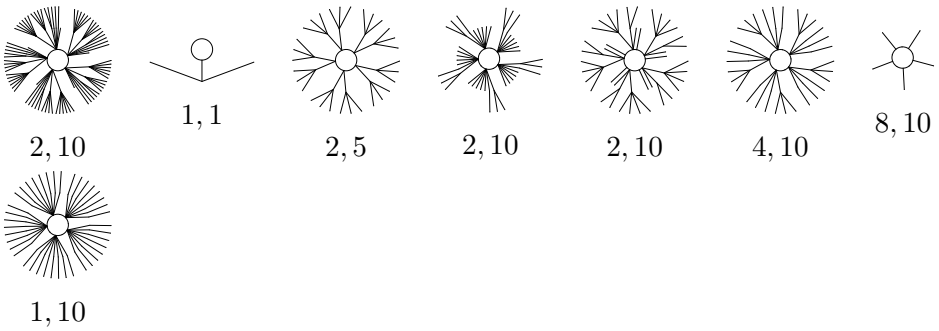
Rule 36 (class 2, max transient 2)



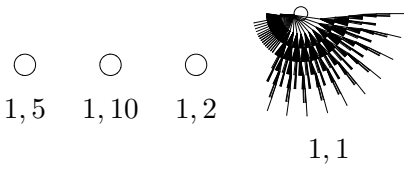
Rule 37 (class 2, max transient 13)



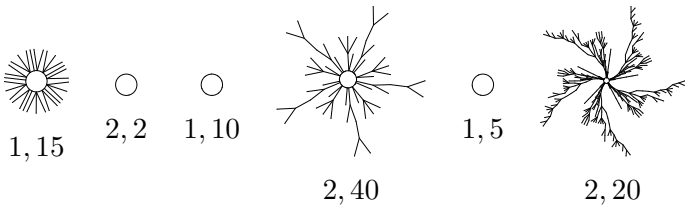
Rule 38 (class 2, max transient 2)



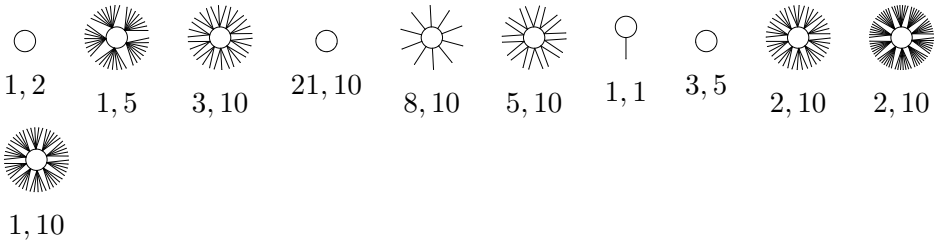
Rule 40 (class 1, max transient 7)



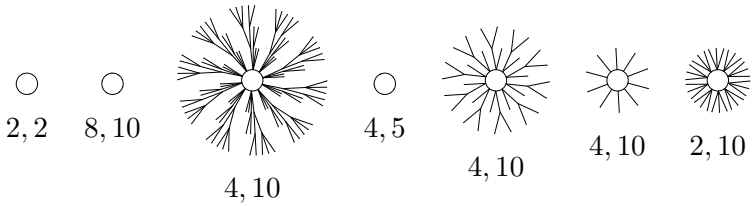
Rule 41 (class 2, max transient 13)



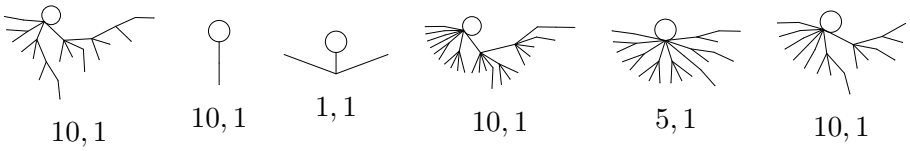
Rule 42 (class 2, max transient 1)



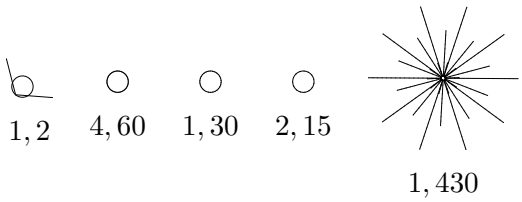
Rule 43 (class 2, max transient 3)



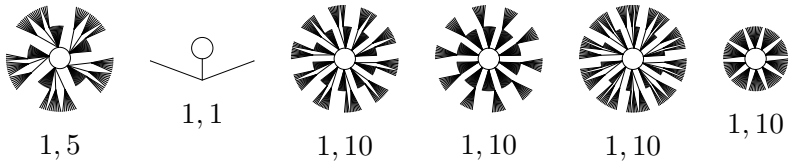
Rule 44 (class 2, max transient 5)



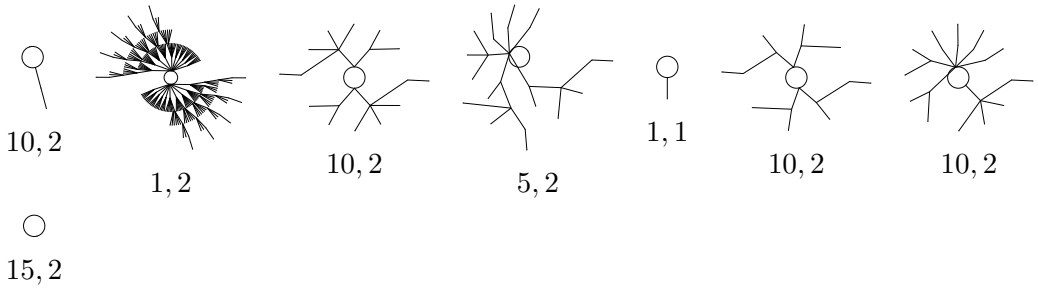
Rule 45 (class 3, max transient 16)



Rule 46 (class 2, max transient 2)

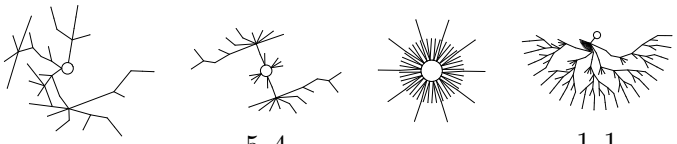


Rule 50 (class 2, max transient 5)



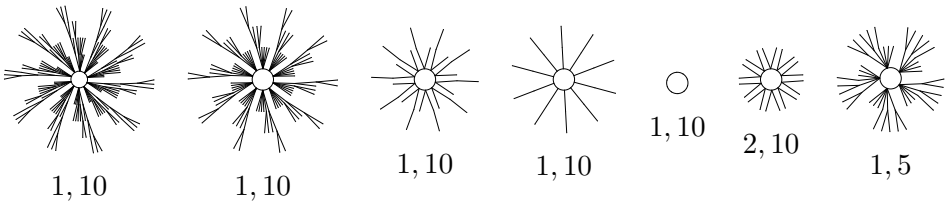
Rule 51 (class 2, max transient 0)

Rule 54 (class 4, max transient 10)

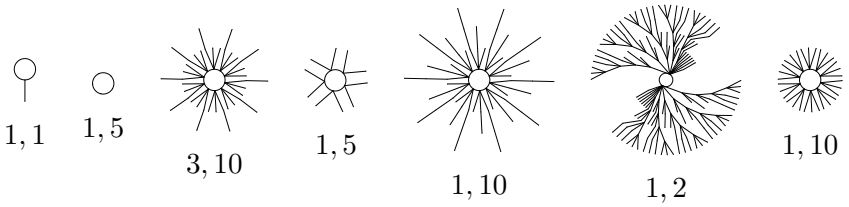


$10, 4$ $5, 4$ $2, 30$ $1, 1$

Rule 56 (class 2, max transient 5)

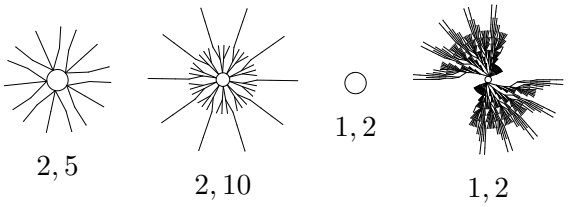


$1, 10$ $1, 10$ $1, 10$ $1, 10$ $1, 10$ $2, 10$ $1, 5$



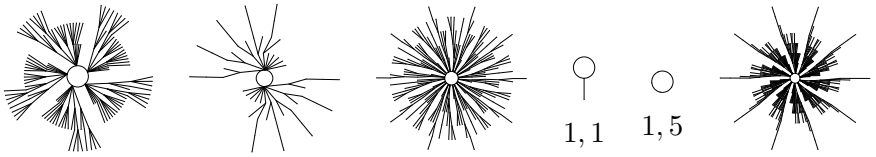
$1, 1$ $1, 5$ $3, 10$ $1, 5$ $1, 10$ $1, 2$ $1, 10$

Rule 57 (class 2, max transient 11)



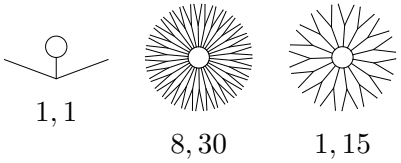
$2, 5$ $2, 10$ $1, 2$ $1, 2$

Rule 58 (class 2, max transient 7)



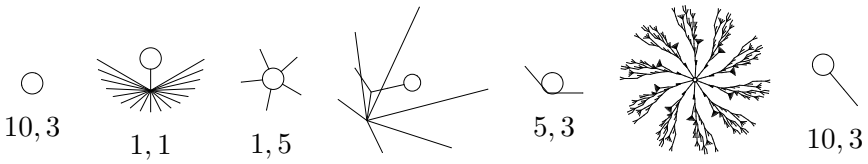
$1, 5$ $1, 2$ $1, 20$ $1, 1$ $1, 5$ $1, 10$

Rule 60 (class 3, max transient 2)

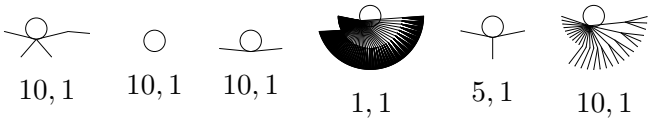


$1, 1$ $8, 30$ $1, 15$

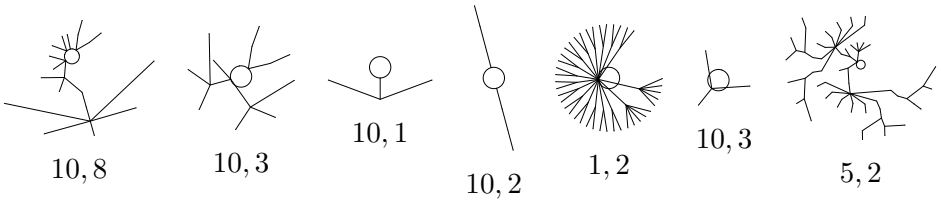
Rule 62 (class 2, max transient 17)



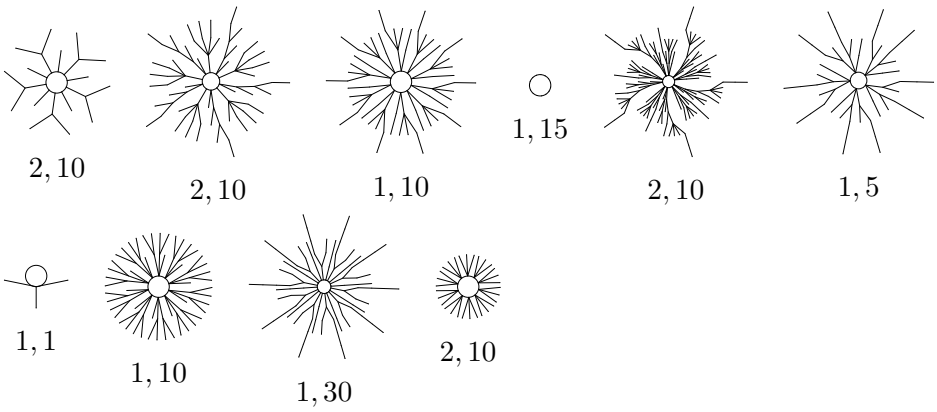
Rule 72 (class 2, max transient 2)



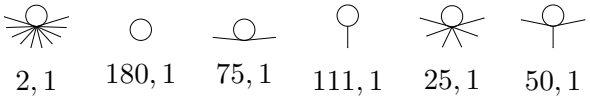
Rule 73 (class 3, max transient 8)



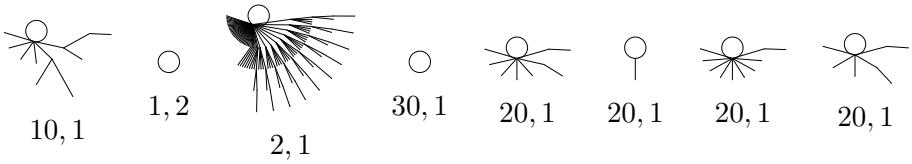
Rule 74 (class 2, max transient 6)



Rule 76 (class 2, max transient 1)

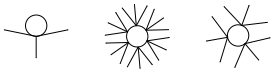


Rule 77 (class 2, max transient 4)

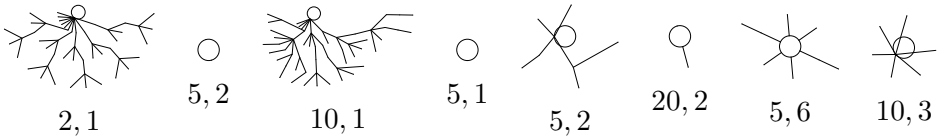


Rule 78 (class 2, max transient 9)

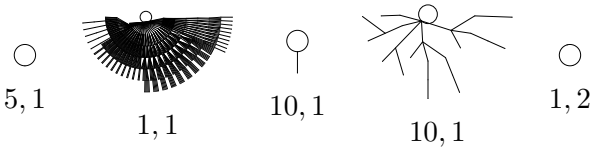
Rule 90 (class 3, max transient 1)



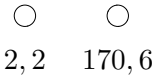
Rule 94 (class 2, max transient 7)



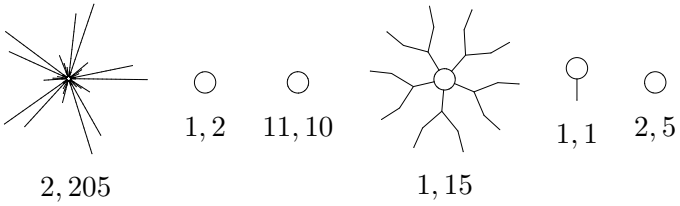
Rule 104 (class 2, max transient 7)



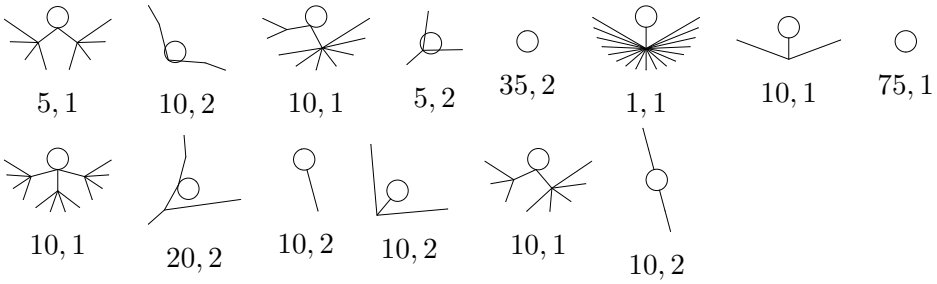
Rule 105 (class 3, max transient 0)



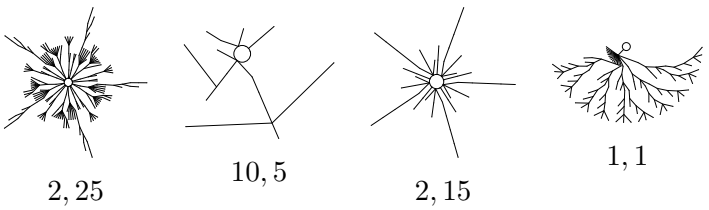
Rule 106 (class 3, max transient 17)



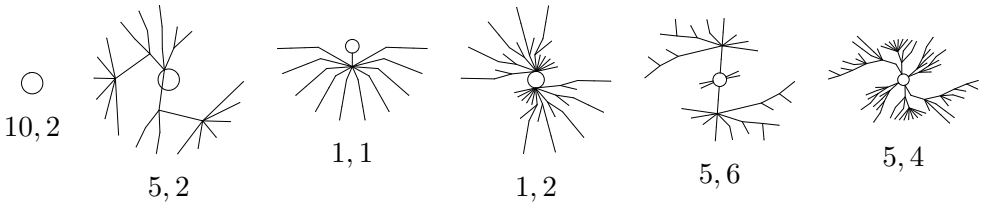
Rule 108 (class 2, max transient 2)



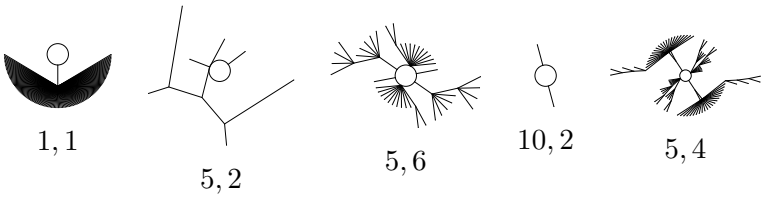
Rule 110 (class 4, max transient 9)



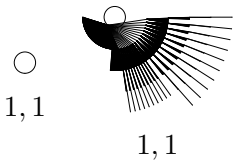
Rule 122 (class 3, max transient 6)



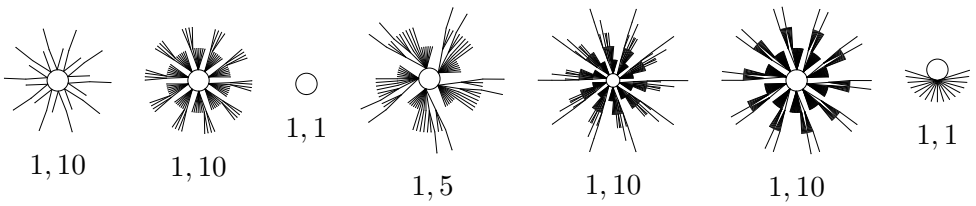
Rule 126 (class 3, max transient 6)



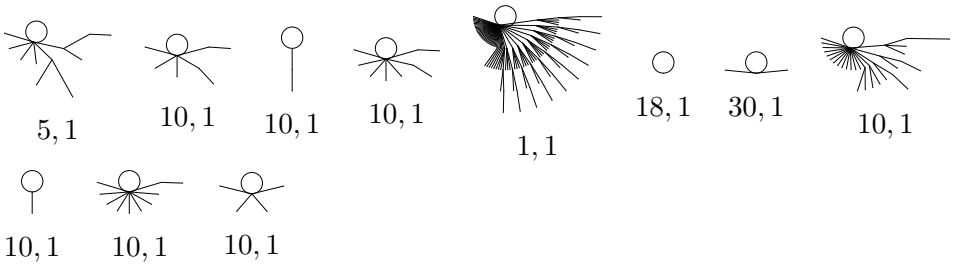
Rule 128 (class 1, max transient 5)



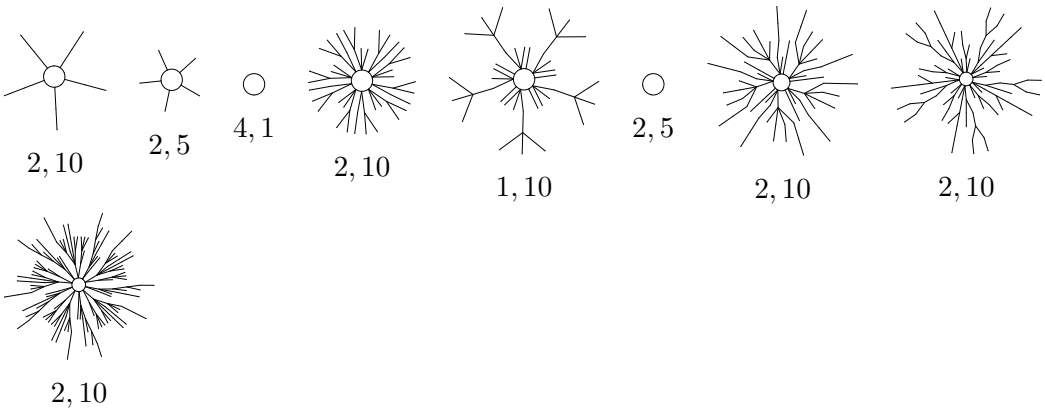
Rule 130 (class 2, max transient 5)



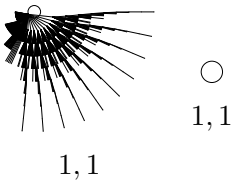
Rule 132 (class 2, max transient 4)



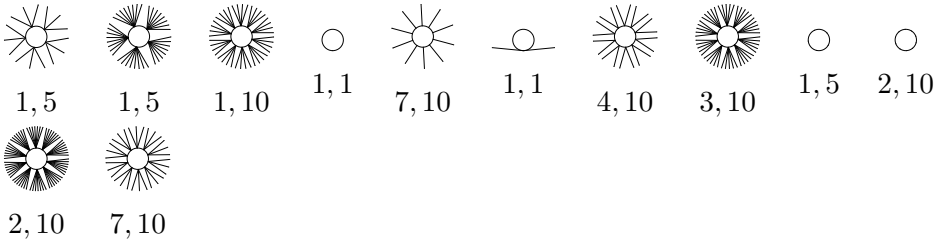
Rule 134 (class 2, max transient 5)



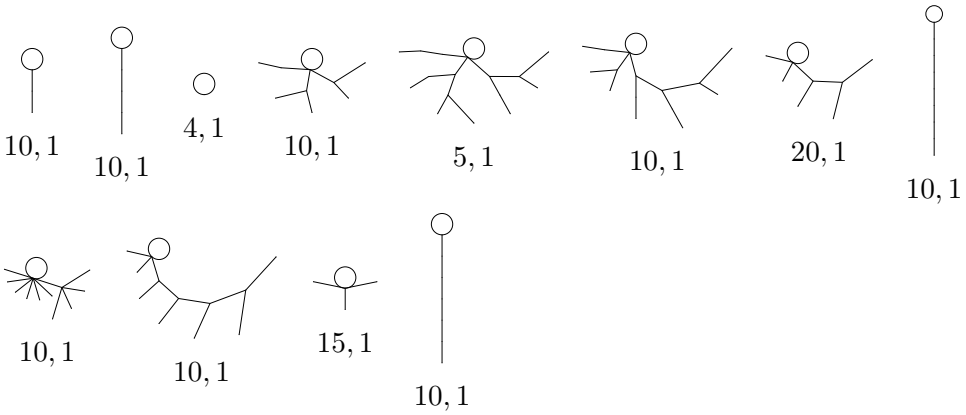
Rule 136 (class 1, max transient 9)



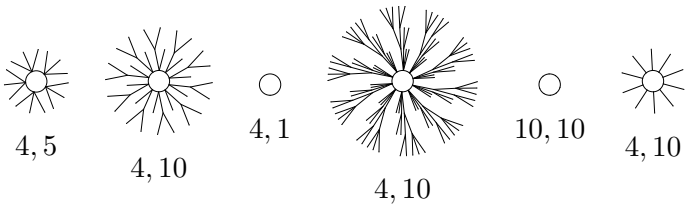
Rule 138 (class 2, max transient 1)



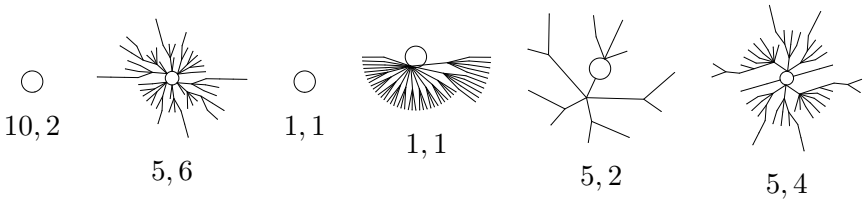
Rule 140 (class 2, max transient 8)



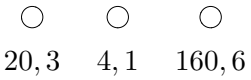
Rule 142 (class 2, max transient 3)



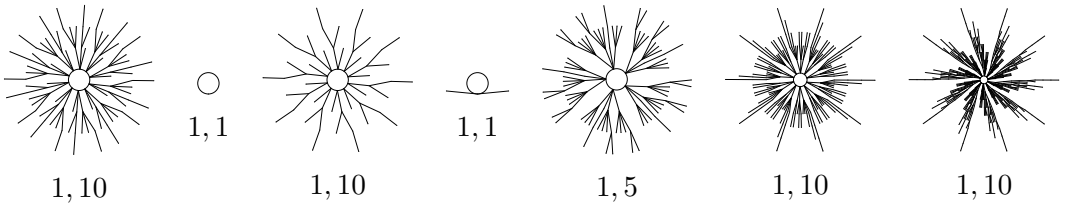
Rule 146 (class 3, max transient 5)



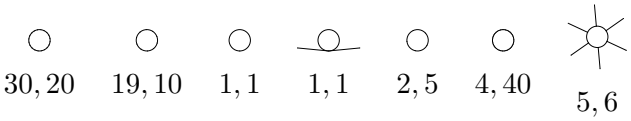
Rule 150 (class 3, max transient 0)



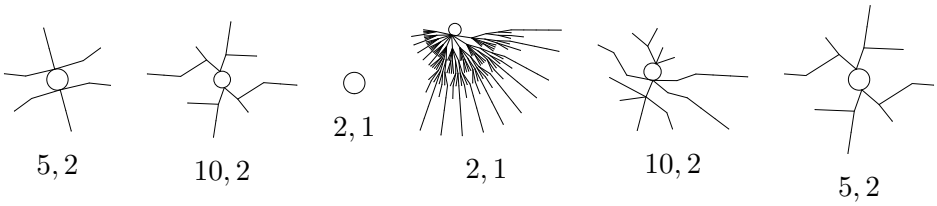
Rule 152 (class 2, max transient 9)



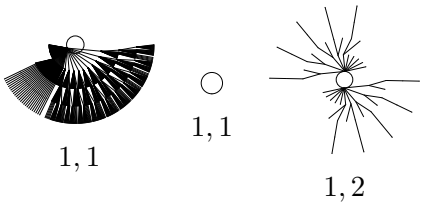
Rule 154 (class 2, max transient 1)



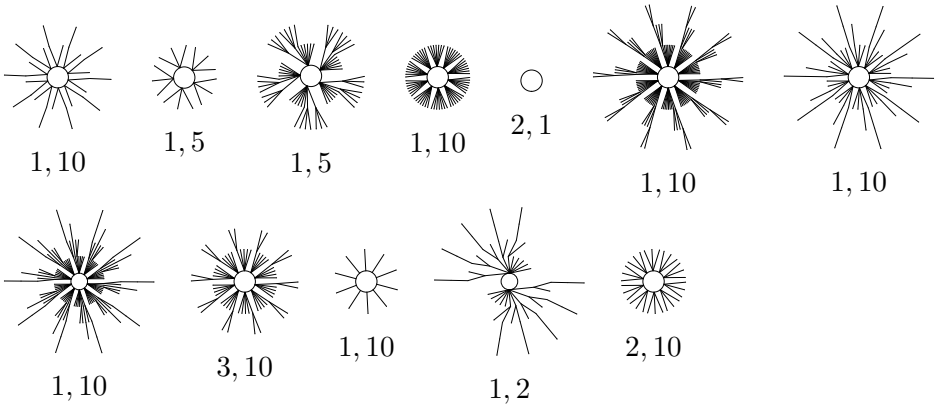
Rule 156 (class 2, max transient 8)



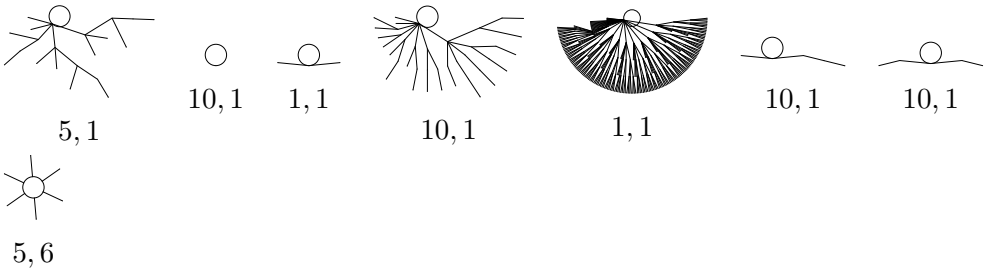
Rule 160 (class 1, max transient 4)



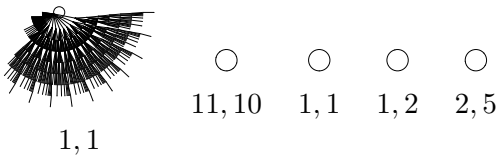
Rule 162 (class 2, max transient 4)



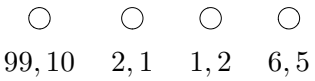
Rule 164 (class 2, max transient 4)



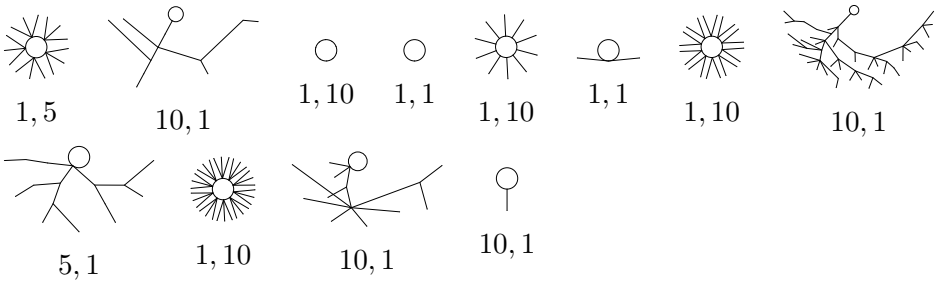
Rule 168 (class 1, max transient 8)



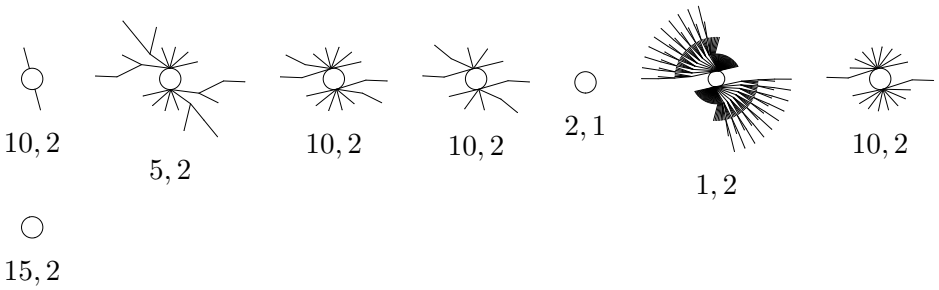
Rule 170 (class 2, max transient 0)



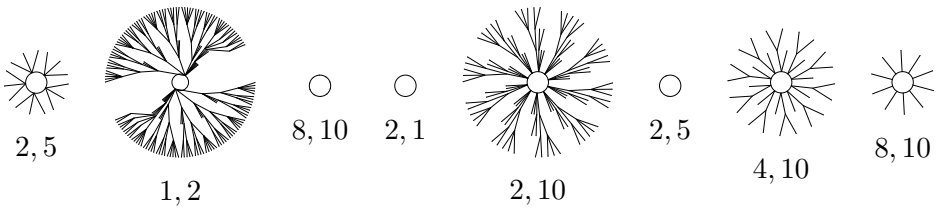
Rule 172 (class 2, max transient 8)



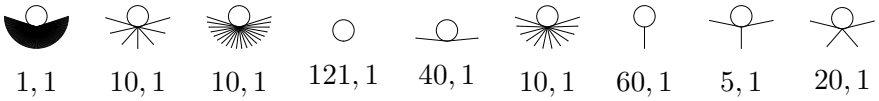
Rule 178 (class 2, max transient 4)



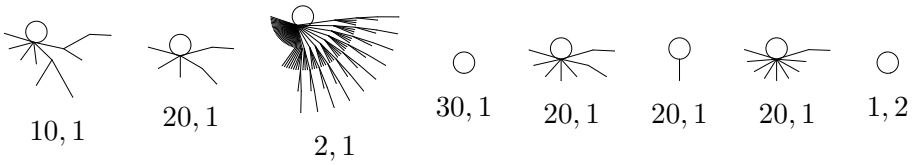
Rule 184 (class 2, max transient 4)



Rule 200 (class 2, max transient 1)



Rule 232 (class 2, max transient 4)

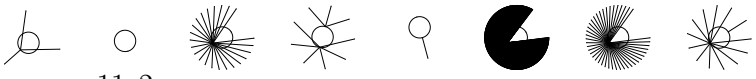


APPENDIX D

Transition graphs for ECAs on $\mathbb{Z}_{11}$

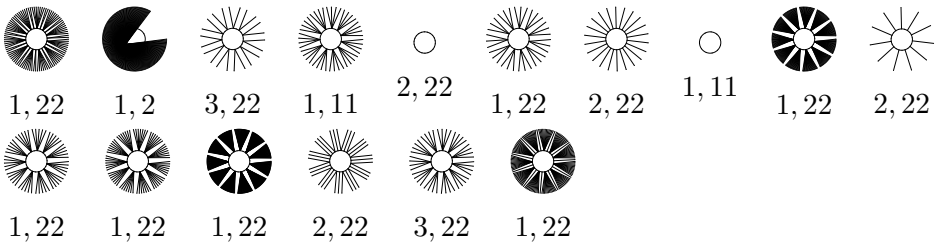
This appendix shows transition graphs for the 88 essentially different ECAs, excluding rule 0 and the identity rule 204, on the periodic lattice $\mathbb{Z}_{11}$. Please refer to the notes at the beginning of Appendix C.

Rule 1 (class 2, max transient 1)



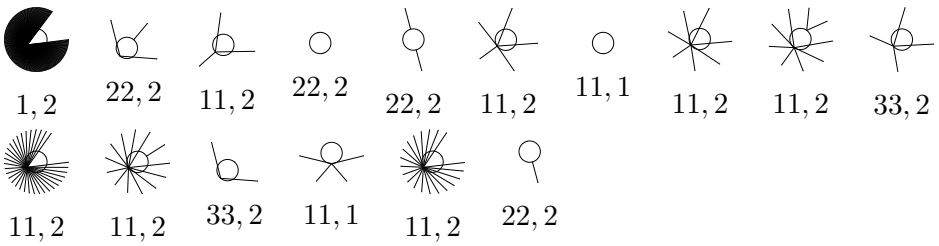
Rule 2 (class 2, max transient 1)

Rule 3 (class 2, max transient 1)

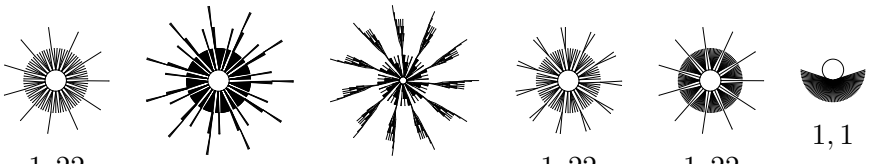


Rule 4 (class 2, max transient 1)

Rule 5 (class 2, max transient 1)

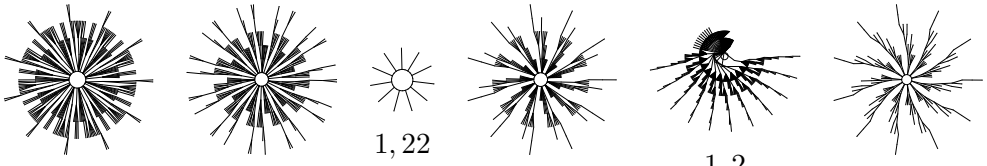


Rule 6 (class 2, max transient 10)



$1,22$ $1,22$ $1,22$ $1,22$ $1,22$ $1,1$

Rule 7 (class 2, max transient 10)

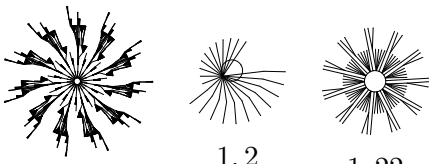


$1,22$ $1,22$ $1,22$ $1,22$ $1,2$ $1,11$

Rule 8 (class 1, max transient 2)

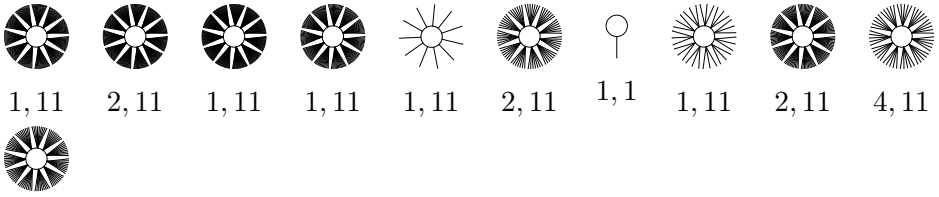
$1,1$

Rule 9 (class 2, max transient 11)



$1,22$ $1,2$ $1,22$

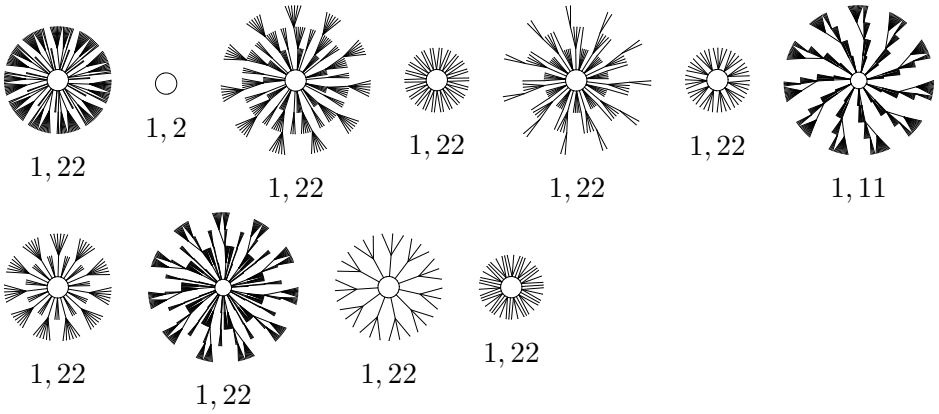
Rule 10 (class 2, max transient 1)



$1,11$ $2,11$ $1,11$ $1,11$ $1,11$ $2,11$ $1,1$ $1,11$ $2,11$ $4,11$

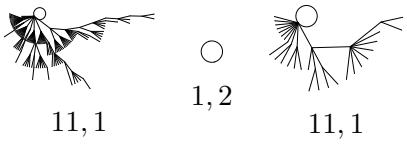
$3,11$

Rule 11 (class 2, max transient 4)

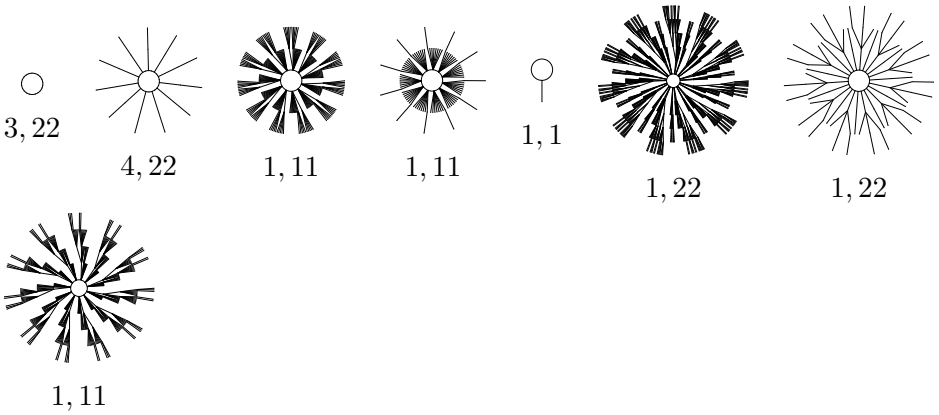


Rule 12 (class 2, max transient 1)

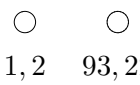
Rule 13 (class 2, max transient 9)



Rule 14 (class 2, max transient 5)



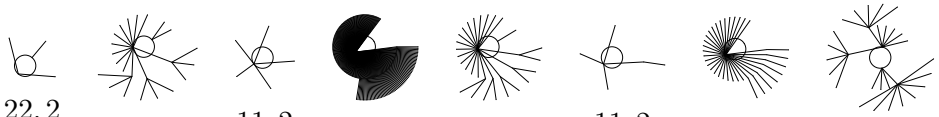
Rule 15 (class 2, max transient 0)



Rule 18 (class 3, max transient 6)

$1,1$ $2,11$ $11,4$ $11,2$

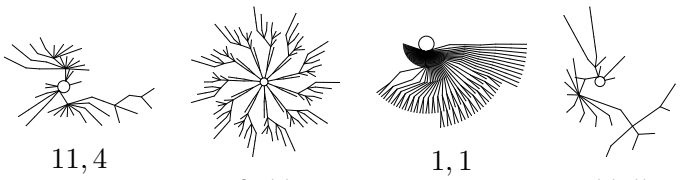
Rule 19 (class 2, max transient 2)



$22,2$ $11,2$ $11,2$ $1,2$ $11,2$ $11,2$ $11,2$ $11,2$

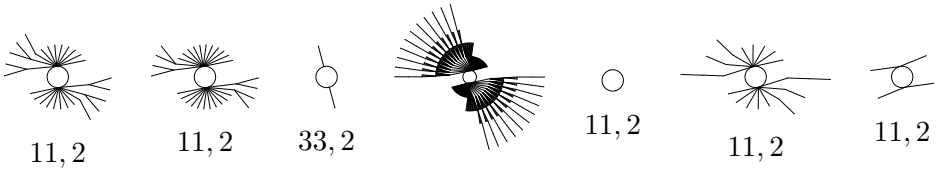
$11,2$

Rule 22 (class 3, max transient 9)

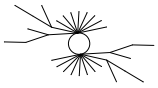


$11,4$ $2,11$ $1,1$ $11,5$

Rule 23 (class 2, max transient 5)



$11,2$ $11,2$ $33,2$ $1,2$ $11,2$ $11,2$ $11,2$

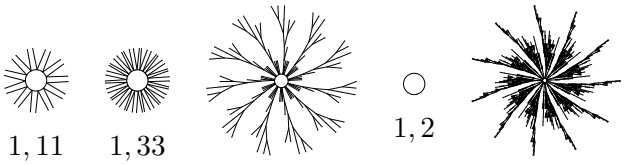


$11,2$

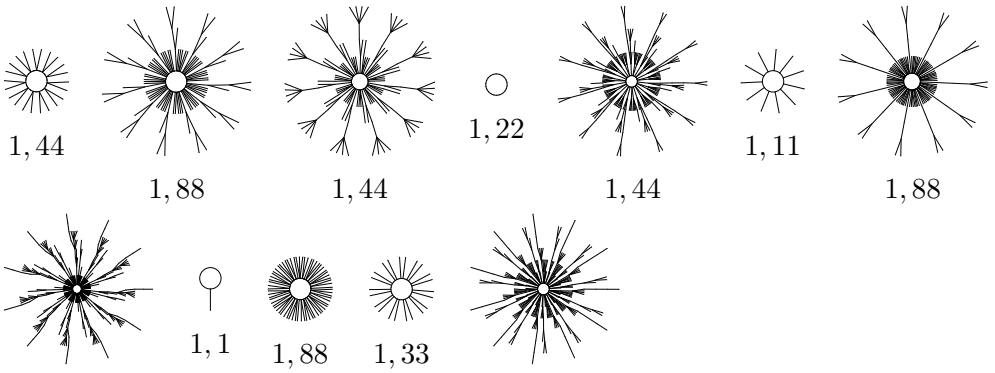
Rule 24 (class 2, max transient 2)

$1,11$ $1,11$ $1,11$ $1,11$ $1,11$ $1,1$ $1,11$

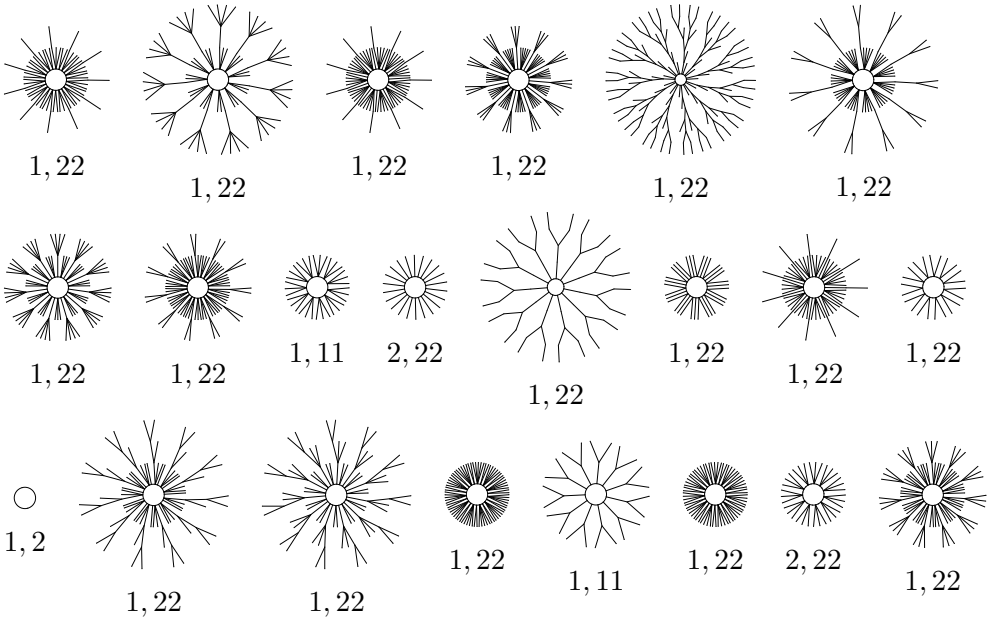
Rule 25 (class 2, max transient 17)



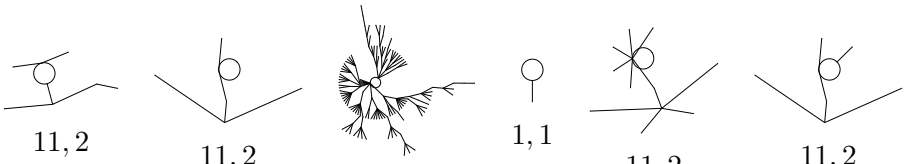
Rule 26 (class 2, max transient 8)



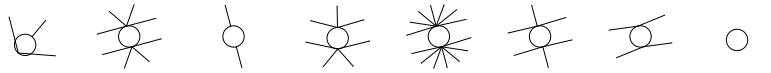
Rule 27 (class 2, max transient 6)



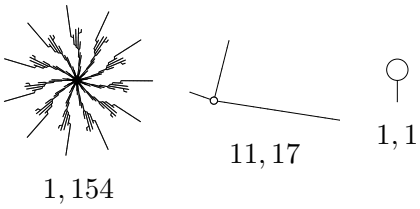
Rule 28 (class 2, max transient 9)



Rule 29 (class 2, max transient 1)

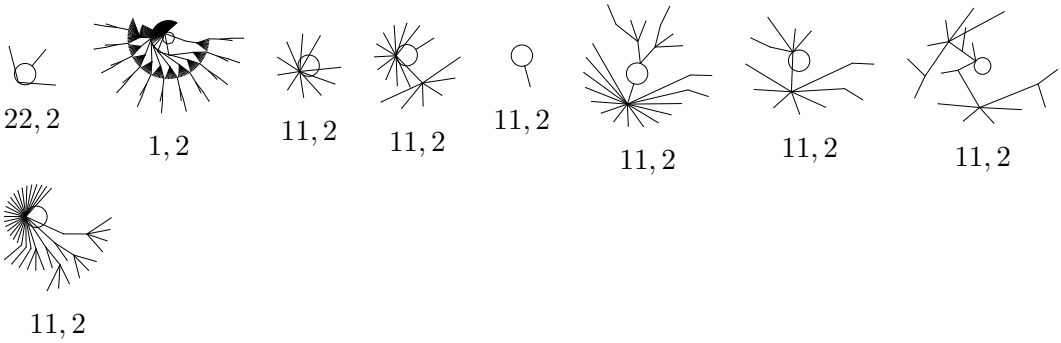


Rule 30 (class 3, max transient 55)

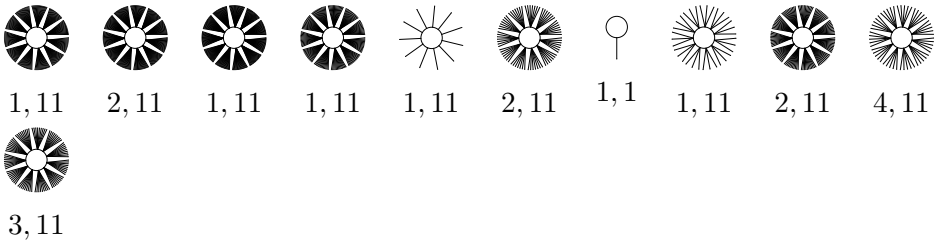


Rule 32 (class 1, max transient 6)

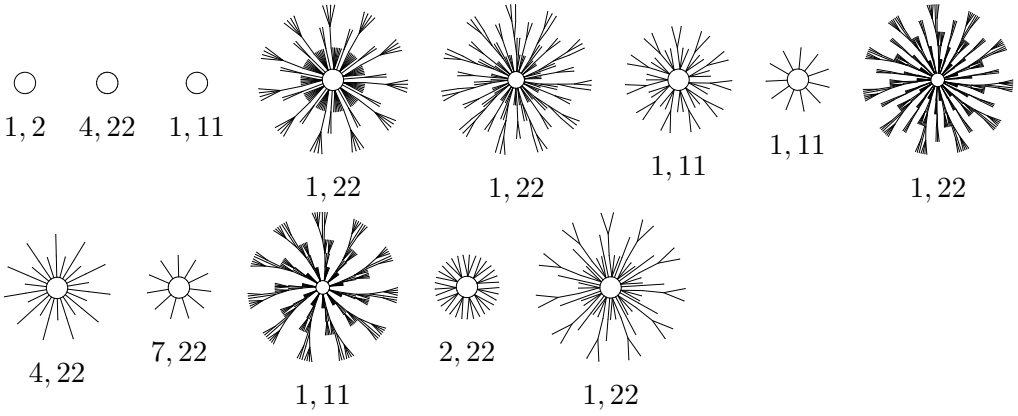
Rule 33 (class 2, max transient 6)



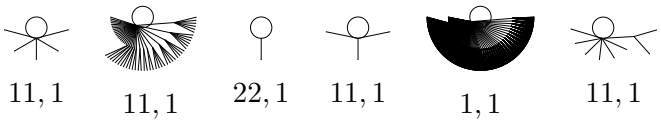
Rule 34 (class 2, max transient 1)



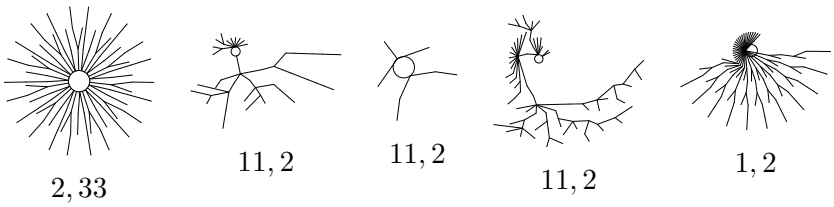
Rule 35 (class 2, max transient 5)



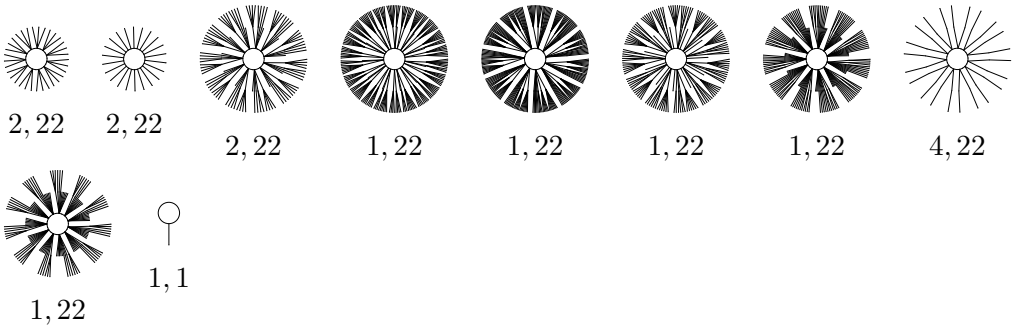
Rule 36 (class 2, max transient 2)



Rule 37 (class 2, max transient 12)



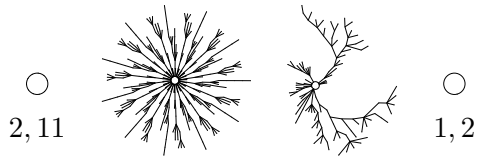
Rule 38 (class 2, max transient 2)



Rule 40 (class 1, max transient 7)

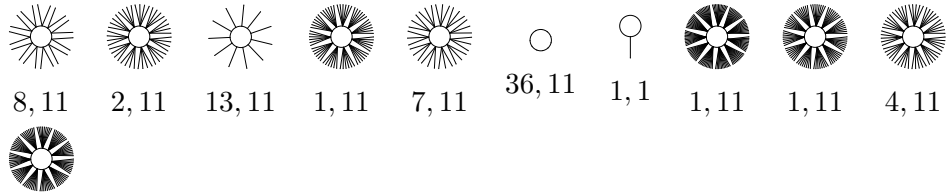
$2, 11$ $1, 1$

Rule 41 (class 2, max transient 10)



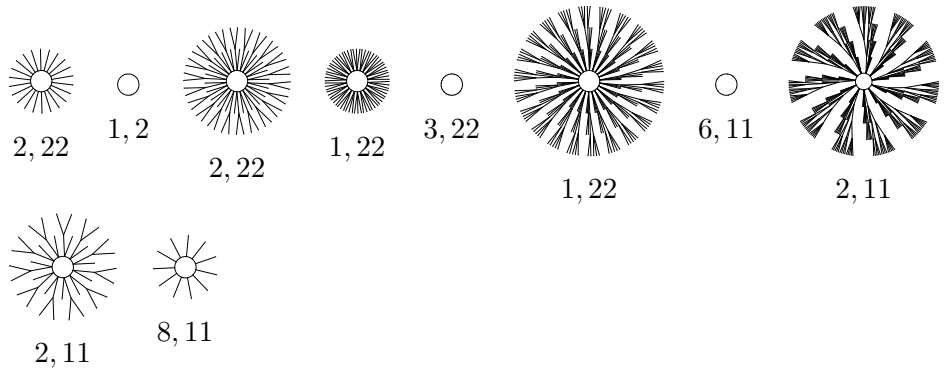
$2, 11$ $1, 44$ $11, 11$ $1, 2$

Rule 42 (class 2, max transient 1)



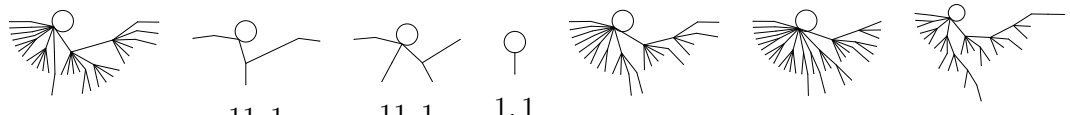
$8, 11$ $2, 11$ $13, 11$ $1, 11$ $7, 11$ $36, 11$ $1, 1$ $1, 11$ $1, 11$ $4, 11$ $1, 11$

Rule 43 (class 2, max transient 4)



$2, 22$ $1, 2$ $2, 22$ $1, 22$ $3, 22$ $1, 22$ $6, 11$ $2, 11$ $2, 11$ $8, 11$

Rule 44 (class 2, max transient 6)



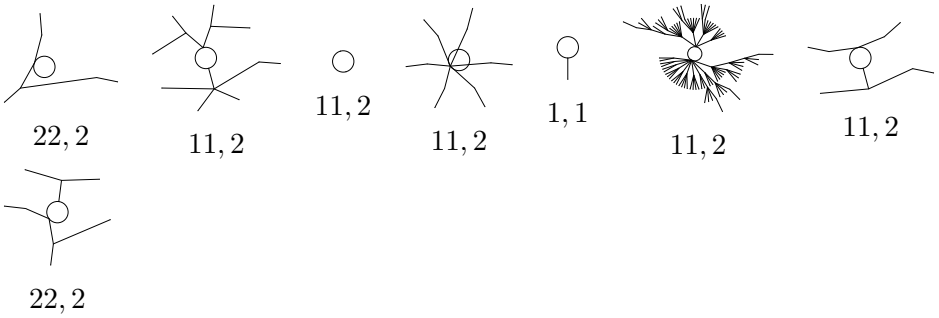
$11, 1$ $11, 1$ $11, 1$ $1, 1$ $11, 1$ $11, 1$ $11, 1$

Rule 45 (class 3, max transient 0)

$1, 2$ $1, 11$ $1, 935$ $11, 5$ $1, 979$ $1, 66$

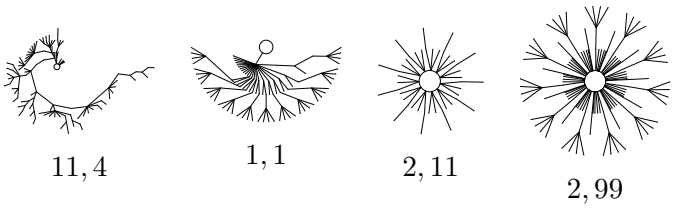
Rule 46 (class 2, max transient 2)

Rule 50 (class 2, max transient 5)

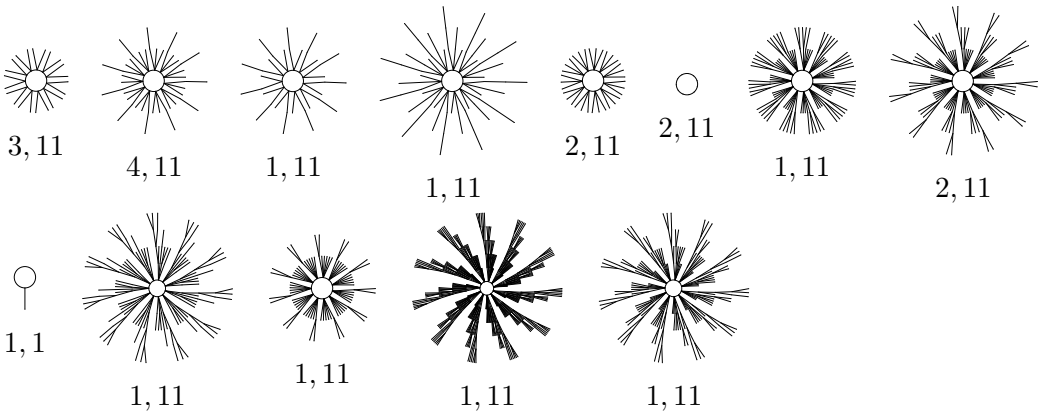


Rule 51 (class 2, max transient 0)

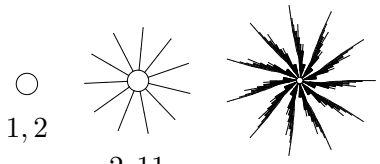
Rule 54 (class 4, max transient 16)



Rule 56 (class 2, max transient 5)

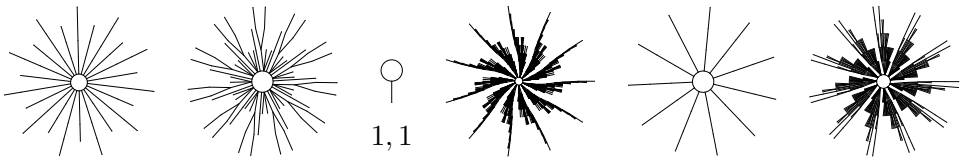


Rule 57 (class 2, max transient 11)



$1, 2$ $2, 11$ $2, 11$

Rule 58 (class 2, max transient 9)

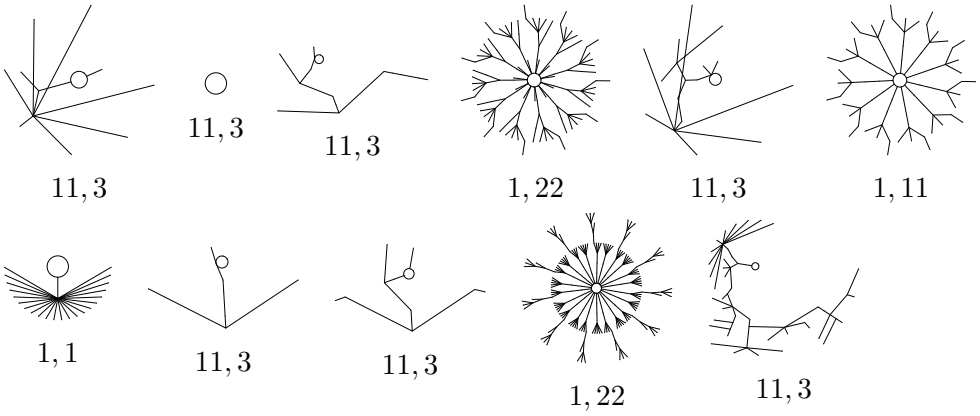


$1, 22$ $1, 22$ $1, 1$ $1, 11$ $1, 11$ $1, 11$

Rule 60 (class 3, max transient 1)

$3, 341$ $1, 1$

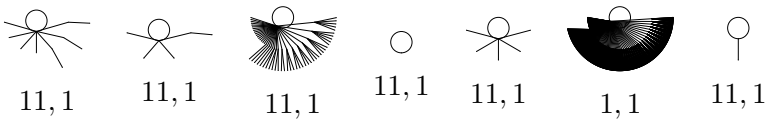
Rule 62 (class 2, max transient 14)



$11, 3$ $11, 3$ $11, 3$ $1, 22$ $11, 3$ $1, 11$

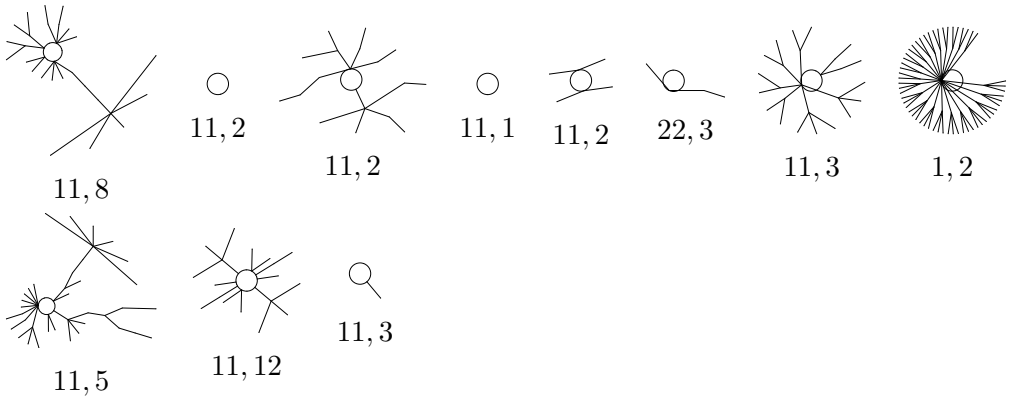
$1, 1$ $11, 3$ $11, 3$ $1, 22$ $11, 3$

Rule 72 (class 2, max transient 2)

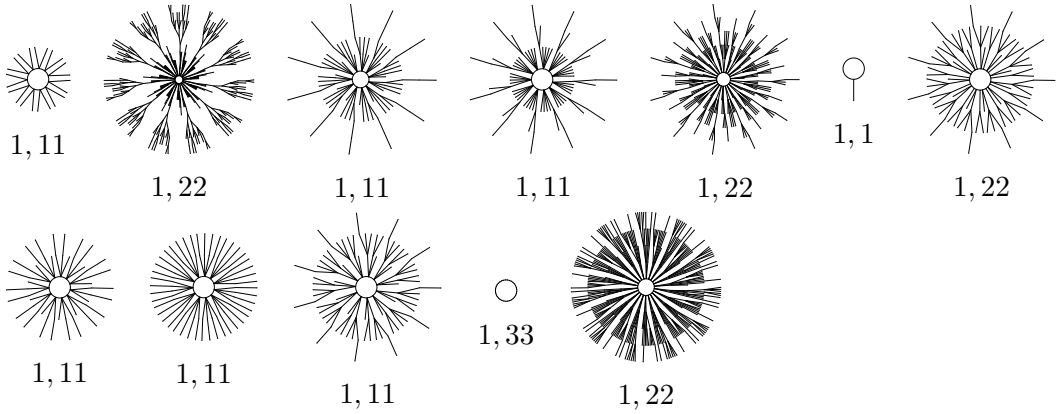


$11, 1$ $11, 1$ $11, 1$ $11, 1$ $11, 1$ $1, 1$ $11, 1$

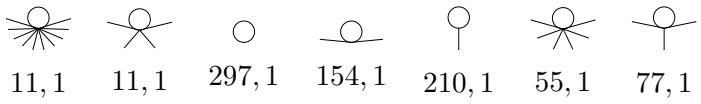
Rule 73 (class 3, max transient 6)



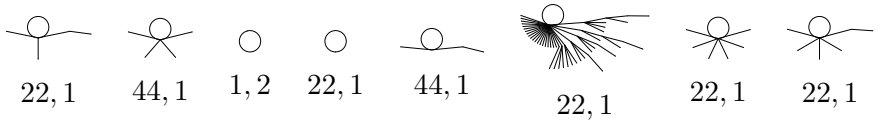
Rule 74 (class 2, max transient 9)



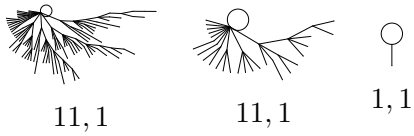
Rule 76 (class 2, max transient 1)



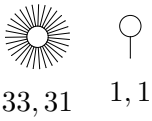
Rule 77 (class 2, max transient 4)



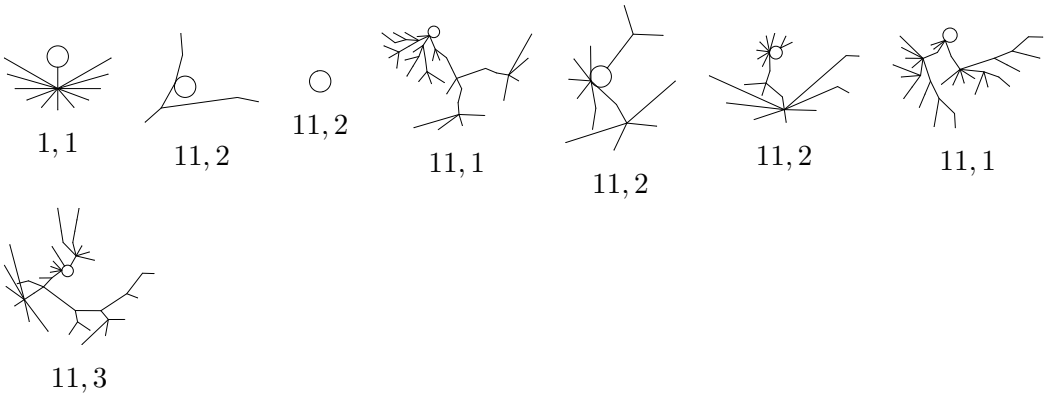
Rule 78 (class 2, max transient 9)



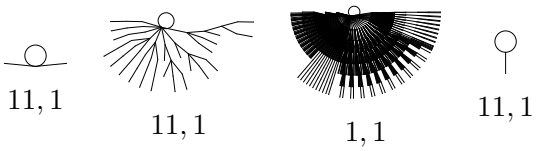
Rule 90 (class 3, max transient 1)



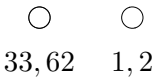
Rule 94 (class 2, max transient 8)



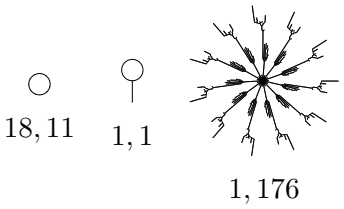
Rule 104 (class 2, max transient 7)



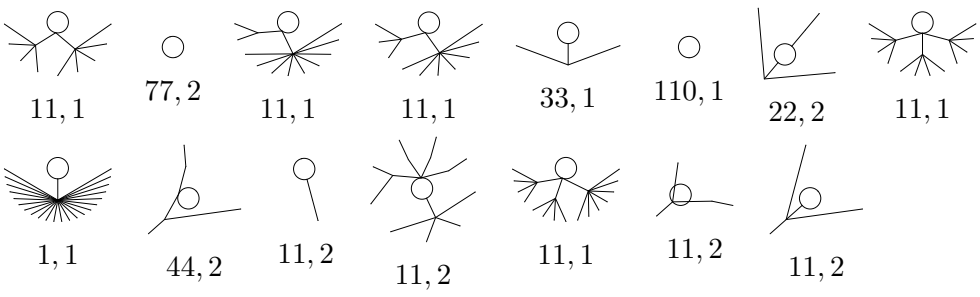
Rule 105 (class 3, max transient 0)



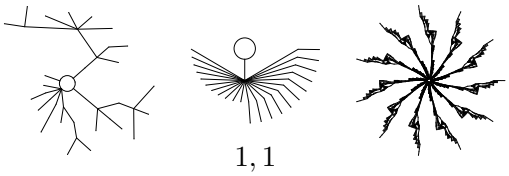
Rule 106 (class 3, max transient 73)



Rule 108 (class 2, max transient 2)

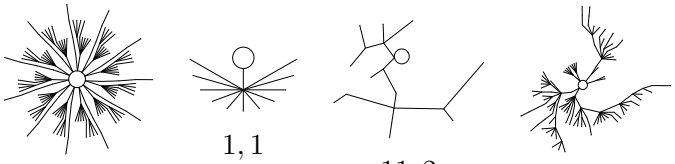


Rule 110 (class 4, max transient 20)



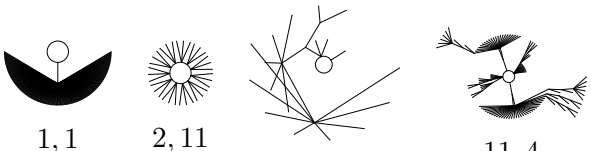
$11, 7$ $1, 1$ $1, 110$

Rule 122 (class 3, max transient 9)



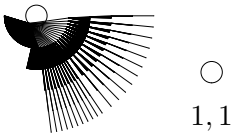
$2, 11$ $1, 1$ $11, 2$ $11, 4$

Rule 126 (class 3, max transient 6)



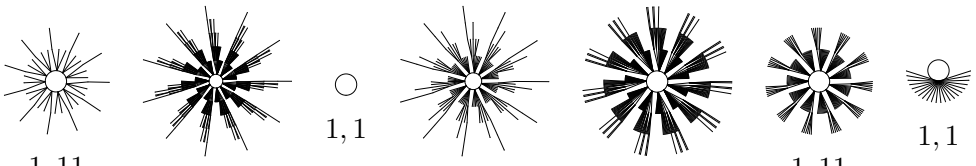
$1, 1$ $2, 11$ $11, 2$ $11, 4$

Rule 128 (class 1, max transient 5)

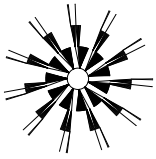


$1, 1$ $1, 1$

Rule 130 (class 2, max transient 5)

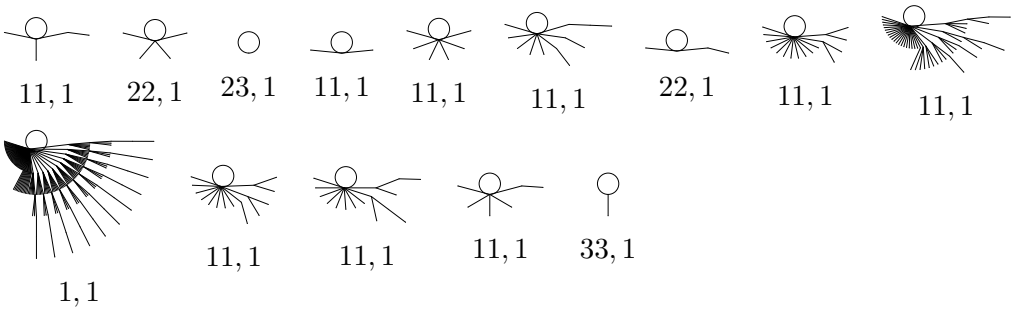


$1, 11$ $1, 11$ $1, 1$ $1, 11$ $1, 11$ $1, 11$ $1, 1$

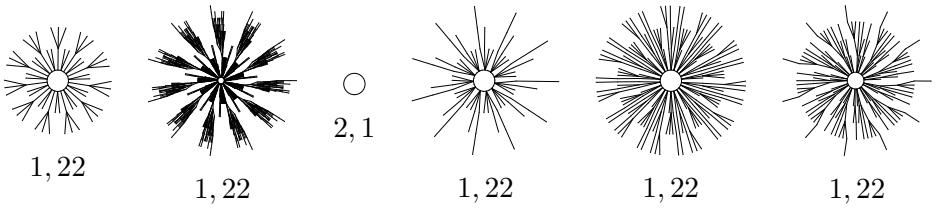


$1, 11$

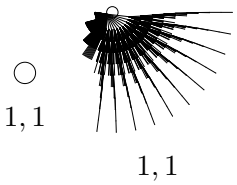
Rule 132 (class 2, max transient 5)



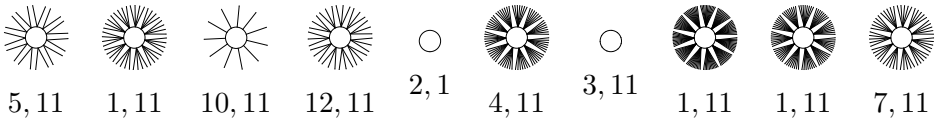
Rule 134 (class 2, max transient 11)



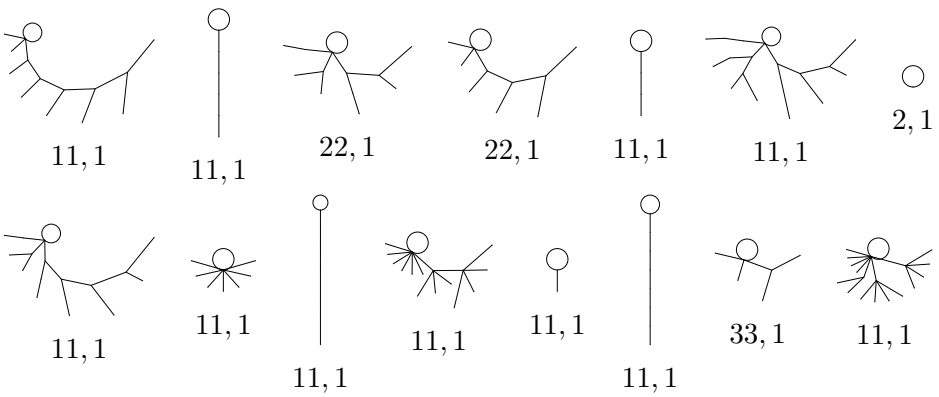
Rule 136 (class 1, max transient 10)



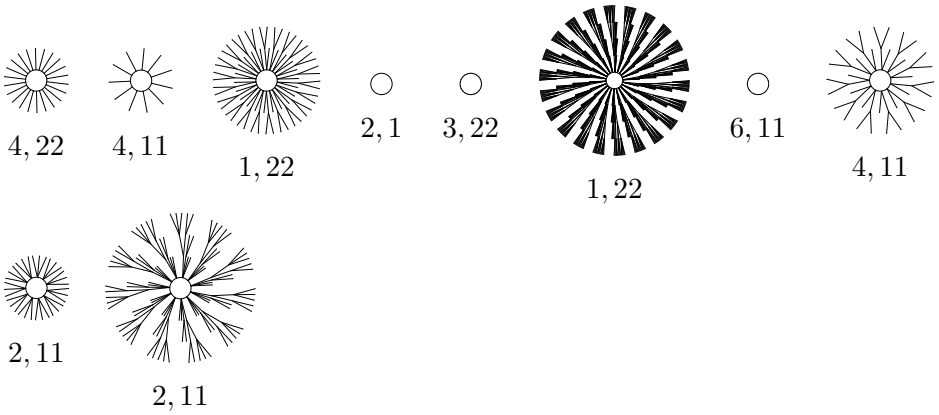
Rule 138 (class 2, max transient 1)



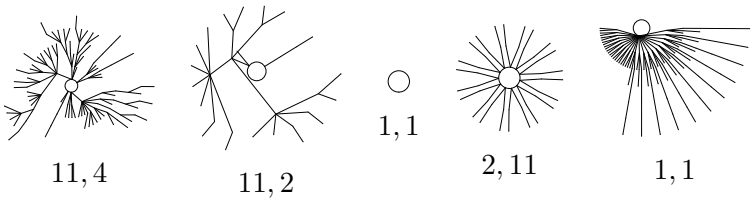
Rule 140 (class 2, max transient 9)



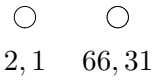
Rule 142 (class 2, max transient 4)



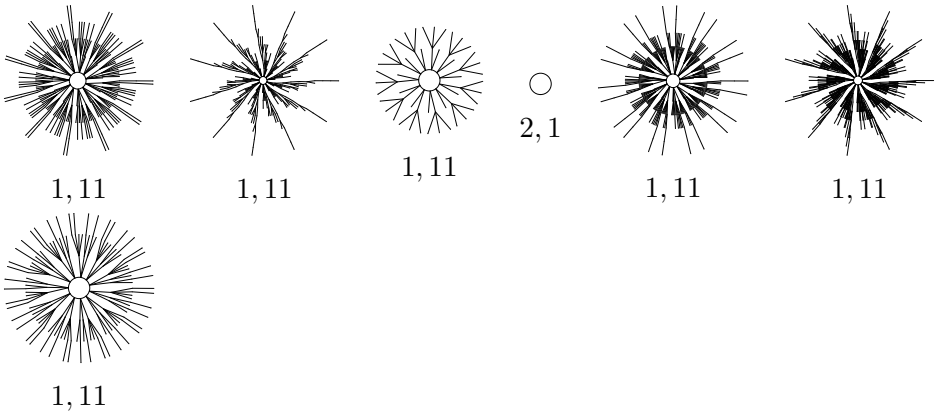
Rule 146 (class 3, max transient 6)



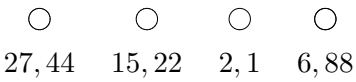
Rule 150 (class 3, max transient 0)



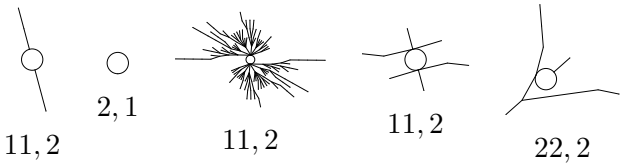
Rule 152 (class 2, max transient 9)



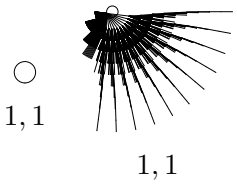
Rule 154 (class 2, max transient 0)



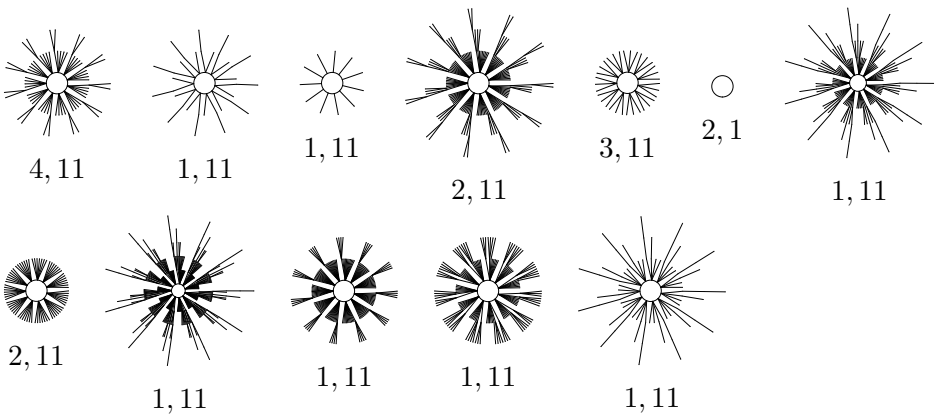
Rule 156 (class 2, max transient 8)



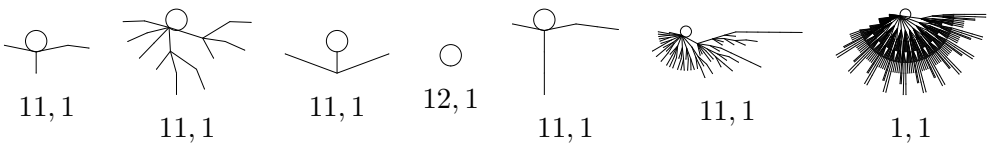
Rule 160 (class 1, max transient 10)



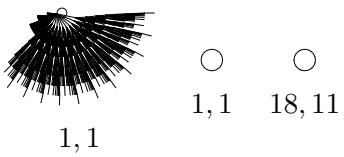
Rule 162 (class 2, max transient 5)



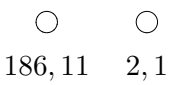
Rule 164 (class 2, max transient 10)



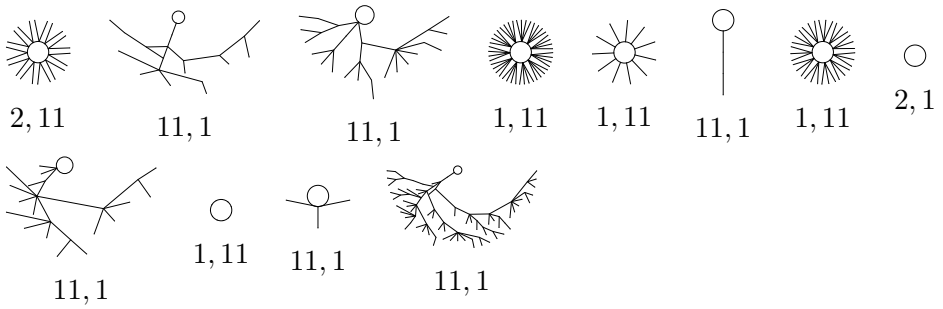
Rule 168 (class 1, max transient 9)



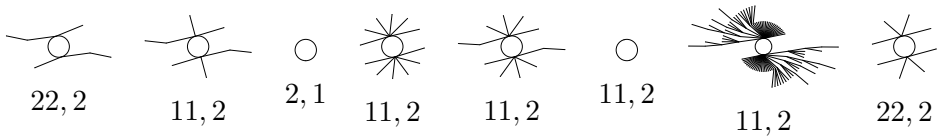
Rule 170 (class 2, max transient 0)



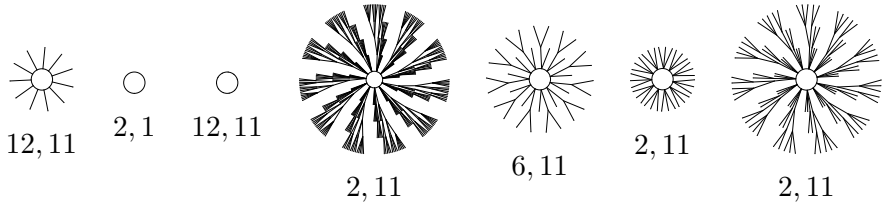
Rule 172 (class 2, max transient 9)



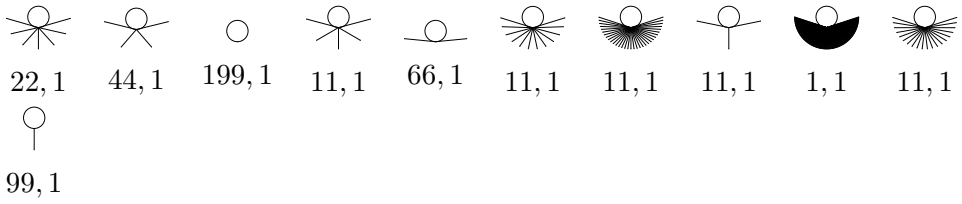
Rule 178 (class 2, max transient 4)



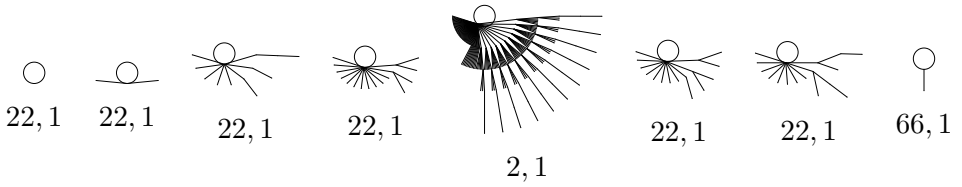
Rule 184 (class 2, max transient 4)



Rule 200 (class 2, max transient 1)



Rule 232 (class 2, max transient 5)



APPENDIX E

Computing cycle lengths and multiplicities for ECA rule 90

The following Mathematica code implements Martin et al's [MOW84] algorithm for finding cycle lengths and multiplicities for ECA rule 90, as used in Section 8.2. In the absence of comments, function and variable names match the notation of [MOW84].

```
D2[M_] := D2[M] = 2^IntegerExponent[M,2]

(* Fast (log-time) simulation of rule 90, see Section 3.3 *)
Rule90Power2[initial_, power_] := BitXor[
    RotateLeft [initial, Mod[2^(power-1), Length[initial]]],
    RotateRight[initial, Mod[2^(power-1), Length[initial]]] ]

Rule90Fast[initial_, step_] := Block[{bits, steps, cfg}, (
    bits = IntegerDigits[step,2];
    steps = bits * Range[Length[bits], 1, -1];
    cfg = initial;
    Do[ If[s > 0, cfg = Rule90Power2[cfg,s]] , {s,steps} ];
    cfg
)]

BigPi[M_] := BigPi[M] =
    If[M == D2[M], 1,
        If[EvenQ[M], 2*BigPi[M/2],
        Catch[
            cfg = Append[ConstantArray[0,M-1], 1];
            cfg = BitXor[ RotateLeft[cfg], RotateRight[cfg] ];
            bigPiStar = 2^MultiplicativeOrder[2,M,{1,-1}] - 1;
            Do[
                If[Rule90Fast[cfg, d] == cfg, Throw[d]]
            , {d, Divisors[bigPiStar]} ]
        ]]
    ]

(*  FactorList returns a list of {factor, power} lists
    We take the first element of each pair (the /@), and get rid
        of any constant factors (the Select)
*)
Cd[d_] := Select[
    #[[1]] & /@ FactorList[Cyclotomic[d, x], Modulus -> 2],
    Exponent[#,x] > 0 &
]
```

```
36
37 Cdi[d_,i_] := Cd[d] [[i]]
38
39 Ttothepi[pi_, M_] := Ttothepi[pi, M] =
40     Rule90Fast[Append[ConstantArray[0,M-1],1], pi] . x^Range[M
       -1,0,-1]
41
42 TtothepimodCditotheD2minuss[pi_, d_, i_, s_, M_] :=
     TtothepimodCditotheD2minuss[pi,d,i,s,M] = (
43     PolynomialRemainder[ Ttothepi[pi,M] , Cdi[d,i]^(D2[M]-s) , x
         , Modulus->2 ]
44 )
45
46 (* Find the smallest pi such that (x+x^-1)^pi is congruent to 1
     *)
47 Pidis[d_,i_,s_,M_] := Pidis[d,i,s,M] = Catch[
48     Do[
49         If[ PolynomialRemainder[ TtothepimodCditotheD2minuss[pi,
               d,i,s,M] - 1 , Cdi[d,i]^(D2[M]-s) , x , Modulus->2 ]
               == 0,
50             Throw[pi]
51         ]
52     , {pi, Divisors[BigPi[M]]} ]
53 ]
54
55 CycleLengths[M_] := If[M == D2[M], {1}, (
56     pidiss = Union @@ Reap[
57         Do[
58             Do[
59                 Do[
60                     Sow[Pidis[d,i,s,M]]
61                 , {s, 0, D2[M]} ]
62             , {i, EulerPhi[d] / MultiplicativeOrder[2,d]} ]
63         , {d, Select[ Divisors[M/D2[M]], # != 1 & ]} ]
64     ] [[2]];
65     Union[ (LCM @@ #)& /@ Subsets[pidiss,{1,Length[pidiss]}] ]
66 )]
67
68 Vrd[r_,d_,M_] := If[r == D2[M], 1,
69     (
70         ord = MultiplicativeOrder[2,d];
71         2^(ord*(D2[M]-r)) - 2^(ord*(D2[M]-r-1))
72     )
73 ]
74
75 Possiblerdis[M_,pi_] := Block[{ dis = {}, rs = {}, rdis = {},
     finalrdis = {} },
76     Do[
77         For[i=1, i <= EulerPhi[d] / MultiplicativeOrder[2,d], i
             ++,
78             AppendTo[dis, {d,i}];
79             AppendTo[rs, OldPossiblerdi[d,i,M,pi]]
```

```
80          ]
81      , {d, Select[ Divisors[M/D2[M]], # != 1 & ]} ];
82      rdis = Tuples[rs];
83      Do[
84          pis = {};
85          For[j=1, j<=Length[dis], j++,
86              {d,i} = dis[[j]];
87              r = rs[[j]];
88              AppendTo[pis, Pidis[d,i,r,M]];
89          ];
90          If[LCM @@ pis == pi, AppendTo[finalrdis, rs]]
91      , {rs, rdis} ];
92      {dis,finalrdis}
93  ]
94
95  (* A set of rdis is only possible if the lcm of the
        corresponding pidis is equal to the desired pi *)
96
97  OldPossiblerdi[d_,i_,M_,pi_] := Block[{ rs = {} },
98      For[r=0, r<=D2[M], r++,
99          If[ PolynomialRemainder[ TtothepimodCditotheD2minuss[pi,
                d,i,r,M] - 1 , Cdi[d,i]^(D2[M]-r) , x , Modulus->2 ]
                == 0,
100                 AppendTo[rs, r]
101         ];
102     ];
103     rs
104 ]
105
106 (* The main function, returns a list of {length, multiplicity}
        pairs. *)
107 CycleMultiplicities[M_] := If[M == D2[M], {{1,1}}, (
108     mults = {};
109     lens = CycleLengths[M];
110     Do[
111         {dis,rdis} = Possiblerdis[M,pi];
112         mult = 0;
113         Do[
114             prodV = 1;
115             For[j=1, j<=Length[dis], j++,
116                 {d,i} = dis[[j]];
117                 r = rs[[j]];
118                 prodV = prodV * Vrd[r,d,M];
119             ];
120             mult = mult + prodV;
121         , {rs, rdis} ];
122         AppendTo[mults, {pi,mult/pi}];
123     , {pi, lens} ];
124     mults
125 )]
```

APPENDIX F

Tables of results for Chapter 9

Table F.1: String lengths for ECAs (see Section 9.1.1)

Rule	$q=0$							$q=1$						
	Permitted l_0			Permitted l_1			Behaviour	Permitted l_0			Permitted l_1			Behaviour
	1	2	≥ 3	1	2	≥ 3		1	2	≥ 3	1	2	≥ 3	
0	•	•	•	•	•	•	Exponential							Constant
1	•	•		•	•	•	Exponential							Constant
2	•			•	•	•	Exponential							Constant
3	•			•	•	•	Exponential							Constant
4	•	•	•		•	•	Exponential				•			Constant
5	•	•			•	•	Exponential				•			Constant
6	•				•	•	Exponential				•			Constant
7	•				•	•	Exponential				•			Constant
8	•	•	•	•			Exponential							Constant
9	•	•		•			Exponential							Constant

Table F.1: String lengths for ECAs (continued)

Rule	$q=0$ Permitted l_0			$q=0$ Permitted l_1			$q=0$ Behaviour	$q=1$ Permitted l_0			$q=1$ Permitted l_1			$q=1$ Behaviour
	1	2	≥ 3	1	2	≥ 3		1	2	≥ 3	1	2	≥ 3	
10	•			•			Period 2							Constant
11	•			•			Period 2							Constant
12	•	•	•				Constant				•			Constant
13	•	•					Constant				•			Constant
14	•						Constant				•			Constant
15	•						Constant				•			Constant
18	•			•	•	•	Exponential		•					Constant
19	•			•	•	•	Exponential		•	•				Constant
22	•				•	•	Exponential		•		•			Period 3
23	•				•	•	Exponential		•	•	•			Exponential
24	•			•			Period 2							Constant
25	•			•			Period 2							Constant
26	•			•			Period 2		•					Constant
27	•			•			Period 2		•	•				Constant
28	•						Constant				•			Constant
29	•						Constant				•			Constant
30	•						Constant		•		•			Period 3
32		•	•	•	•	•	Exponential	•						Constant

Table F.1: String lengths for ECAs (continued)

Rule	$q=0$ Permitted l_0			$q=0$ Permitted l_1			$q=0$ Behaviour	$q=1$ Permitted l_0			$q=1$ Permitted l_1			$q=1$ Behaviour
	1	2	≥ 3	1	2	≥ 3		1	2	≥ 3	1	2	≥ 3	
33		•		•	•	•	Exponential	•						Constant
34				•	•	•	Constant	•						Constant
35				•	•	•	Constant	•						Constant
36		•	•		•	•	Exponential	•			•			Period 2
37		•			•	•	Exponential	•			•			Period 2
38					•	•	Constant	•			•			Period 2
40		•	•	•			Exponential	•						Constant
41		•		•			Period 3	•						Constant
42				•			Constant	•						Constant
43				•			Constant	•						Constant
44		•	•				Constant	•			•			Period 2
45		•					Constant	•			•			Period 2
46							Constant	•			•			Period 2
50				•	•	•	Constant	•	•					Constant
51				•	•	•	Constant	•	•	•				Constant
54					•	•	Constant	•	•		•			Exponential
56				•			Constant	•						Constant
57				•			Constant	•						Constant

Table F.1: String lengths for ECAs (continued)

Rule	$q=0$ Permitted l_0			$q=0$ Permitted l_1			$q=0$ Behaviour	$q=1$ Permitted l_0			$q=1$ Permitted l_1			$q=1$ Behaviour
	1	2	≥ 3	1	2	≥ 3		1	2	≥ 3	1	2	≥ 3	
58				•			Constant	•	•					Constant
60							Constant	•			•			Period 2
62							Constant	•	•		•			Exponential
72	•	•	•	•			Exponential					•		Constant
73	•	•		•			Exponential					•		Constant
74	•			•			Period 2					•		Constant
76	•	•	•				Constant				•	•		Constant
77	•	•					Constant				•	•		Constant
78	•						Constant				•	•		Constant
90	•			•			Period 2		•			•		Period 4
94	•						Constant		•		•	•		Exponential
104		•	•	•			Exponential	•				•		Period 3
105		•		•			Period 3	•				•		Period 3
106				•			Constant	•				•		Period 3
108		•	•				Constant	•			•	•		Exponential
110							Constant	•			•	•		Exponential
122				•			Constant	•	•			•		Exponential
126							Constant	•	•		•	•		Exponential

Table F.1: String lengths for ECAs (continued)

Rule	$q=0$ Permitted l_0			$q=0$ Permitted l_1			$q=0$ Behaviour	$q=1$ Permitted l_0			$q=1$ Permitted l_1			$q=1$ Behaviour
	1	2	≥ 3	1	2	≥ 3		1	2	≥ 3	1	2	≥ 3	
128	•	•	•	•	•		Exponential							Constant
130	•			•	•		Exponential							Constant
132	•	•	•		•		Exponential				•			Constant
134	•				•		Period 3				•			Constant
136	•	•	•	•			Exponential							Constant
138	•			•			Period 2							Constant
140	•	•	•				Constant				•			Constant
142	•						Constant				•			Constant
146	•			•	•		Exponential		•					Constant
150	•				•		Period 3		•		•			Period 3
152	•			•			Period 2							Constant
154	•			•			Period 2		•					Constant
156	•						Constant				•			Constant
160		•	•	•	•		Exponential	•						Constant
162				•	•		Constant	•						Constant
164		•	•		•		Exponential	•			•			Period 2
168		•	•	•			Exponential	•						Constant
170				•			Constant	•						Constant

Table F.1: String lengths for ECAs (continued)

Rule	$q=0$ Permitted l_0			$q=0$ Permitted l_1			$q=0$ Behaviour	$q=1$ Permitted l_0			$q=1$ Permitted l_1			$q=1$ Behaviour
	1	2	≥ 3	1	2	≥ 3		1	2	≥ 3	1	2	≥ 3	
172		•	•				Constant	•			•			Period 2
178				•	•		Constant	•	•					Constant
184				•			Constant	•						Constant
200	•	•	•	•			Exponential					•	•	Constant
204	•	•	•				Constant				•	•	•	Constant
232		•	•	•			Exponential	•				•	•	Exponential

Table F.2: Characteristic polynomials for ECAs. The polynomials are ordered by their roots of largest magnitude (see Table F.3), and so that the polynomials are grouped by the classes of behaviour given in Table F.3, but other than that the ordering (and thus the numbering) is arbitrary.

c_i	Polynomial
c_1	λ^4
c_2	$\lambda^4 - \lambda^3$
c_3	$\lambda^4 - 2\lambda^3 + \lambda^2$
c_4	$\lambda^4 - \lambda^2$
c_5	$\lambda^4 - \lambda^3 - \lambda^2 + \lambda$
c_6	$\lambda^4 - 2\lambda^3 + 2\lambda - 1$
c_7	$\lambda^4 - \lambda$
c_8	$\lambda^4 - \lambda^3 - \lambda + 1$
c_9	$\lambda^4 - 1$
c_{10}	$\lambda^4 - \lambda - 1$
c_{11}	$\lambda^4 - \lambda^2 - \lambda$
c_{12}	$\lambda^4 - \lambda^3 - \lambda^2 + 1$
c_{13}	$\lambda^4 - \lambda^3 - 1$
c_{14}	$\lambda^4 - \lambda^3 - \lambda$
c_{15}	$\lambda^4 - 2\lambda^3 + \lambda^2 - \lambda + 1$
c_{16}	$\lambda^4 - \lambda^3 - \lambda^2$
c_{17}	$\lambda^4 - 2\lambda^3 + \lambda$
c_{18}	$\lambda^4 - \lambda^3 - \lambda - 1$
c_{19}	$\lambda^4 - 2\lambda^3 + \lambda^2 - 1$
c_{20}	$\lambda^4 - \lambda^2 - 2\lambda - 1$
c_{21}	$\lambda^4 - 2\lambda^3 + \lambda^2 - \lambda$
c_{22}	$\lambda^4 - \lambda^3 - \lambda^2 - \lambda$
c_{23}	$\lambda^4 - 2\lambda^3$

Table F.3: Roots of characteristic polynomials for ECAs

c_i	Cartesian form	Modulus	Argument/π	Behaviour
c_1	0	0	0	Constant
	0	0	0	
	0	0	0	
	0	0	0	
c_2	1	1	0	Constant
	0	0	0	
	0	0	0	
	0	0	0	
c_3	1	1	0	Constant
	1	1	0	
	0	0	0	
	0	0	0	
c_4	1	1	0	Period 2
	-1	1	1	
	0	0	0	
	0	0	0	
c_5	1	1	0	Period 2
	1	1	0	
	-1	1	1	
	0	0	0	
c_6	1	1	0	Period 2
	1	1	0	
	1	1	0	
	-1	1	1	
c_7	$-\frac{1}{2}+\frac{\sqrt{3}}{2}i$	1	$\frac{2}{3}$	Period 3
	$-\frac{1}{2}-\frac{\sqrt{3}}{2}i$	1	$-\frac{2}{3}$	
	1	1	0	
	0	0	0	
c_8	$-\frac{1}{2}+\frac{\sqrt{3}}{2}i$	1	$\frac{2}{3}$	Period 3
	$-\frac{1}{2}-\frac{\sqrt{3}}{2}i$	1	$-\frac{2}{3}$	
	1	1	0	
	1	1	0	
c_9	1	1	0	Period 4
	i	1	$\frac{1}{2}$	
	$-i$	1	$-\frac{1}{2}$	
	-1	1	1	

Table F.3: Roots of characteristic polynomials for ECAs (continued)

c_i	Cartesian form	Modulus	Argument/π	Behaviour
c_{10}	1.22074	1.22074	0	Exponential
	$-0.24813 + 1.03398i$	1.06334	0.57497	
	$-0.24813 - 1.03398i$	1.06334	-0.57497	
	-0.72449	0.72449	1	
c_{11}	1.32472	1.32472	0	Exponential
	$-0.66236 - 0.56228i$	0.86884	-0.77596	
	$-0.66236 + 0.56228i$	0.86884	0.77596	
	0	0	0	
c_{12}	1.32472	1.32472	0	Exponential
	1	1	0	
	$-0.66236 - 0.56228i$	0.86884	-0.77596	
	$-0.66236 + 0.56228i$	0.86884	0.77596	
c_{13}	1.38028	1.38028	0	Exponential
	$0.21945 + 0.91447i$	0.94044	0.42503	
	$0.21945 - 0.91447i$	0.94044	-0.42503	
	-0.81917	0.81917	1	
c_{14}	1.46557	1.46557	0	Exponential
	$-0.23279 - 0.79255i$	0.82603	-0.59094	
	$-0.23279 + 0.79255i$	0.82603	0.59094	
	0	0	0	
c_{15}	1.46557	1.46557	0	Exponential
	1	1	0	
	$-0.23279 - 0.79255i$	0.82603	-0.59094	
	$-0.23279 + 0.79255i$	0.82603	0.59094	
c_{16}	$\frac{1+\sqrt{5}}{2}$	$\frac{1+\sqrt{5}}{2}$	0	Exponential
	$\frac{1-\sqrt{5}}{2}$	$\frac{\sqrt{5}-1}{2}$	1	
	0	0	0	
	0	0	0	
c_{17}	$\frac{1+\sqrt{5}}{2}$	$\frac{1+\sqrt{5}}{2}$	0	Exponential
	1	1	0	
	$\frac{1-\sqrt{5}}{2}$	$\frac{\sqrt{5}-1}{2}$	1	
	0	0	0	

Table F.3: Roots of characteristic polynomials for ECAs (continued)

c_i	Cartesian form	Modulus	Argument/π	Behaviour
c_{18}	$\frac{1+\sqrt{5}}{2}$	$\frac{1+\sqrt{5}}{2}$	0	Exponential
	i	1	$\frac{1}{2}$	
	$-i$	1	$-\frac{1}{2}$	
	$\frac{1-\sqrt{5}}{2}$	$\frac{\sqrt{5}-1}{2}$	1	
c_{19}	$\frac{1+\sqrt{5}}{2}$	$\frac{1+\sqrt{5}}{2}$	0	Exponential
	$\frac{1}{2}-\frac{\sqrt{3}}{2}i$	1	$-\frac{1}{3}$	
	$\frac{1}{2}+\frac{\sqrt{3}}{2}i$	1	$\frac{1}{3}$	
	$\frac{1-\sqrt{5}}{2}$	$\frac{\sqrt{5}-1}{2}$	1	
c_{20}	$\frac{1+\sqrt{5}}{2}$	$\frac{1+\sqrt{5}}{2}$	0	Exponential
	$-\frac{1}{2}+\frac{\sqrt{3}}{2}i$	1	$\frac{2}{3}$	
	$-\frac{1}{2}-\frac{\sqrt{3}}{2}i$	1	$-\frac{2}{3}$	
	$\frac{1-\sqrt{5}}{2}$	$\frac{\sqrt{5}-1}{2}$	1	
c_{21}	1.75488	1.75488	0	Exponential
	$0.12256-0.74486i$	0.75488	-0.44809	
	$0.12256+0.74486i$	0.75488	0.44809	
	0	0	0	
c_{22}	1.83929	1.83929	0	Exponential
	$-0.41964-0.60629i$	0.73735	-0.69272	
	$-0.41964+0.60629i$	0.73735	0.69272	
	0	0	0	
c_{23}	2	2	0	Exponential
	0	0	0	
	0	0	0	
	0	0	0	

Table F.4: Characteristic polynomials of de Bruijn matrices for ECAs

Rule	char D_0	char D_1	Rule	char D_0	char D_1
0	c_{23}	c_1	56	c_3	c_1
1	c_{22}	c_2	57	c_2	c_2
2	c_{17}	c_1	58	c_3	c_1
3	c_{16}	c_2	60	c_3	c_4
4	c_{21}	c_1	62	c_3	c_{11}
5	c_{18}	c_2	72	c_{17}	c_1
6	c_{15}	c_1	73	c_{12}	c_2
7	c_{14}	c_2	74	c_6	c_1
8	c_{17}	c_1	76	c_3	c_1
9	c_{12}	c_2	77	c_2	c_2
10	c_6	c_1	78	c_3	c_1
11	c_5	c_2	90	c_6	c_9
12	c_3	c_1	94	c_3	c_{10}
13	c_2	c_2	104	c_{15}	c_7
14	c_3	c_1	105	c_8	c_8
15	c_2	c_2	106	c_3	c_7
18	c_{17}	c_1	108	c_3	c_{11}
19	c_{16}	c_2	110	c_3	c_{11}
22	c_{15}	c_7	122	c_3	c_{10}
23	c_{14}	c_{14}	126	c_3	c_{20}
24	c_6	c_1	128	c_{22}	c_2
25	c_5	c_2	130	c_{12}	c_2
26	c_6	c_1	132	c_{14}	c_2
27	c_5	c_2	134	c_8	c_2
28	c_3	c_1	136	c_{16}	c_2
29	c_2	c_2	138	c_5	c_2
30	c_3	c_7	140	c_2	c_2
32	c_{21}	c_1	142	c_2	c_2
33	c_{14}	c_2	146	c_{12}	c_2
34	c_3	c_1	150	c_8	c_8
35	c_2	c_2	152	c_5	c_2
36	c_{19}	c_4	154	c_5	c_2
37	c_{13}	c_5	156	c_2	c_2
38	c_3	c_4	160	c_{18}	c_2
40	c_{15}	c_1	162	c_2	c_2
41	c_8	c_2	164	c_{13}	c_5

Table F.4: Characteristic polynomials of de Bruijn matrices for ECAs (continued)

Rule	char D_0	char D_1	Rule	char D_0	char D_1
42	c_3	c_1	168	c_{14}	c_2
43	c_2	c_2	170	c_2	c_2
44	c_3	c_4	172	c_2	c_5
45	c_2	c_5	178	c_2	c_2
46	c_3	c_4	184	c_2	c_2
50	c_3	c_1	200	c_{16}	c_2
51	c_2	c_2	204	c_2	c_2
54	c_3	c_{11}	232	c_{14}	c_{14}

Bibliography

[Ada02] Andrew Adamatzky, editor. *Collision-Based Computing*. Springer, 2002.

[Bae02] John Baez. Noether's Theorem in a nutshell. `http://math.ucr.edu/home/baez/noether.html`, 2002.

[BCG82] Elwyn R. Berlekamp, John H. Conway, and Richard K. Guy. *Winning ways for your mathematical plays*, volume 2. Academic Press, 1982.

[Boy03] Tim Boykett. Towards a Noether-like conservation law theorem for one dimensional reversible cellular automata. `http://arxiv.org/abs/nlin/0312003`, 2003.

[Bur70] Arthur W. Burks, editor. *Essays on Cellular Automata*. University of Illinois Press, 1970.

[CDS95] Dragoš M. Cvetković, Michael Doob, and Horst Sachs. *Spectra of Graphs*. Johann Ambrosius Barth, third edition, 1995.

[Cha06] Sam Chapman. String similarity metrics for information integration. `http://www.dcs.shef.ac.uk/~sam/stringmetrics.html`, 2006.

[Coo04] Matthew Cook. Universality in elementary cellular automata. *Complex Systems*, 15(1):1–40, 2004.

[CR91] Douglas M. Campbell and David Radford. Tree isomorphism algorithms: Speed vs. clarity. *Mathematics Magazine*, 64(4):252–261, 1991.

[CY88] Karel Culik II and Sheng Yu. Undecidability of CA classification schemes. *Complex Systems*, 2:177–190, 1988.

[dB46] Nicolaas Govert de Bruijn. A combinatorial problem. *Nederl. Akad. Wetensch. Proceedings*, 49:758–764, 1946.

[Epp] David Eppstein. Wolfram's classification of cellular automata. `http://www.ics.uci.edu/~eppstein/ca/wolfram.html`.

[FHP86] U. Frisch, B. Hasslacher, and Y. Pomeau. Lattice-gas automata for the Navier-Stokes equation. *Physical Review Letters*, 56(14):1505–1508, Apr 1986.

[Flo67] Robert W. Floyd. Nondeterministic algorithms. *Journal of the ACM*, 14(4):636–644, 1967.

[Fra03] John B. Fraleigh. *A First Course in Abstract Algebra*. Addison-Wesley, seventh edition, 2003.

[Gam06] Emily Gamber. Equicontinuity properties of D-dimensional cellular automata. *Topology Proceedings*, 30(1):197–222, 2006.

[Gan60] Felix R. Gantmacher. *Matrix Theory*, volume I. Chelsea, 1960.

[Gar70] Martin Gardner. Mathematical games: The fantastic combinations of John Conway's new solitaire game "life". *Scientific American*, 223:120–123, 1970.

[GDH07] Stéphane Gobron, François Devillard, and Bernard Heit. Retina simulation using cellular automata and GPU programming. *Machine Vision and Applications*, 18(6):331–342, 2007.

[Gle88] James Gleick. *Chaos*. Vintage, 1988.

[GN00] Emden R. Gansner and Stephen C. North. An open graph visualization system and its applications to software engineering. *Software — Practice and Experience*, 30(11):1203–1233, 2000.

[GT88] Harold N. Gabow and Robert E. Tarjan. A linear-time algorithm for finding a minimum spanning pseudoforest. *Information Processing Letters*, 27(5):259–263, 1988.

[Har69] Frank Harary. *Graph Theory*. Addison-Wesley, 1969.

[Hol98] John H. Holland. *Emergence: From chaos to order*. Oxford University Press, 1998.

[HT91] Tetsuya Hattori and Shinji Takesue. Additive conserved quantities in discrete-time lattice dynamical systems. *Physica D*, 49:295–322, 1991.

[JD06] Iztok Jeras and Andrej Dobnikar. Algorithms for computing preimages of cellular automata configurations. `http://www.rattus.info/al/files/preimages.pdf`, 2006.

[Joh01] Steven Johnson. *Emergence: The connected lives of ants, brains, cities and software*. Penguin, 2001.

[KK89] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15, 1989.

[Knu69] Donald E. Knuth. *The Art of Computer Programming*, volume 2: Seminumerical Algorithms. Addison Wesley, 1969.

[Kůr97] Petr Kůrka. Languages, equicontinuity and attractors in cellular automata. *Ergodic Theory and Dynamical Systems*, 17:417–433, 1997.

[Lam98] Lui Lam, editor. *Nonlinear Physics for Beginners: Fractals, chaos, solitons, pattern formation, cellular automata and complex systems*. World Scientific, 1998.

[Lan90] Chris G. Langton. Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D*, 42:12–37, 1990.

[LP90] Wentian Li and Norman Packard. The structure of the elementary cellular automata rule space. *Complex Systems*, 4:281–297, 1990.

[LPL90] Wentian Li, Norman H. Packard, and Chris Langton. Transition phenomena in cellular automata rule space. *Physica D*, 45:77–94, 1990.

[McI90] Harold V. McIntosh. Linear cellular automata via de Bruijn diagrams. `http://delta.cs.cinvestav.mx/~mcintosh/comun/cf/debruijn.pdf`, 1990.

[MH05] Carsten Marr and Marc-Thorsten Hütt. Topology regulates pattern formation capacity of binary cellular automata on graphs. *Physica A*, 354:641–662, 2005.

[MOW84] Olivier Martin, Andrew M. Odlyzko, and Stephen Wolfram. Algebraic properties of cellular automata. *Communications in Mathematical Physics*, 93:219–258, 1984.

[OdOO01] Gina M. B. Oliveira, Pedro P. B. de Oliveira, and Nizam Omar. Definition and application of a five-parameter characterization of one-dimensional cellular automata rule space. *Artificial Life*, 7:277–301, 2001.

[Ost08] Stephen Ostermiller. Finding a loop in a singly linked list. `http://ostermiller.org/find_loop_singly_linked_list.html`, 2008.

[Ped92] John Pedersen. Decision problems for cellular automata and their semigroups. In *MFCS '92: Proceedings of the 17th International Symposium on Mathematical Foundations of Computer Science*, number 629 in Lecture Notes in Computer Science, pages 421–429. Springer-Verlag, 1992.

[Pos47] Emil L. Post. Recursive unsolvability of a problem of Thue. *Journal of Symbolic Logic*, 12(1):1–11, 1947.

[Pow07] Edward Powley. Mathematical properties of cellular automata. Qualifying dissertation, Department of Computer Science, University of York, 2007.

[Pow08] Edward Powley. Quad prize submission: Simulating elementary CAs with Trid CAs, 2008. To appear in Journal of Cellular Automata.

[PP02] Athanasios Papoulis and S. Unnikrishna Pillai. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, fourth edition, 2002.

[PS08] Edward Powley and Susan Stepney. Automorphisms of transition graphs for a linear cellular automaton. In Andrew Adamatzky, Ramon Alonso-Sanz, Anna Lawniczak, Genaro Juarez Martinez, Kenichi Morita, and Thomas Worsch, editors, *proceedings of Automata 2008: Theory and Applications of Cellular Automata*, pages 55–68. Luniver Press, 2008.

[PS09a] Edward J. Powley and Susan Stepney. Automorphisms of transition graphs for elementary cellular automata. *Journal of Cellular Automata*, 4(2):125–136, 2009.

[PS09b] Edward J. Powley and Susan Stepney. Automorphisms of transition graphs for linear cellular automata. *Journal of Cellular Automata*, 4(4):293–310, 2009.

[PS09c] Edward J. Powley and Susan Stepney. Counting preimages of homogeneous configurations in 1-dimensional cellular automata. *Journal of Cellular Automata*, 2009. In press.

[PS09d] Edward J. Powley and Susan Stepney. Distribution of Hamming distances for cellular automaton transitions. *Complex Systems*, 2009. Submitted.

[PW85] Norman H. Packard and Stephen Wolfram. Two-dimensional cellular automata. *Journal of Statistical Physics*, 38(5):901–946, March 1985.

[Ral82] Anthony Ralston. De Bruijn sequences — a model example of the interaction of discrete mathematics and computer science. *Mathematics Magazine*, 55(3):131–143, 1982.

[Ren02] Paul Rendell. Turing univerality of the game of life. In Adamatzky [Ada02], chapter 18.

[Rot00] Joseph J. Rotman. *A First Course in Abstract Algebra*. Prentice-Hall, second edition, 2000.

[Sil06] Stephen Silver. Life lexicon home page. `http://www.argentum.freeserve.co.uk/lex_home.htm`, 2006.

[Smi71] Alvy Ray Smith, III. Simple computation-universal cellular spaces. *Journal of the ACM*, 18(3):339–353, 1971.

[STCS02] Barry Shackleford, Motoo Tanaka, Richard J. Carter, and Greg Snider. FPGA implementation of neighborhood-of-four cellular automata random number generators. In *FPGA '02: Proceedings of the 2002 ACM/SIGDA tenth international symposium on Field-programmable gate arrays*, pages 106–112. ACM, 2002.

[Sut09] Klaus Sutner. Classification of cellular automata. In *Encyclopedia of Complexity and Systems Science*. Springer, 2009.

[TM87] Tommaso Toffoli and Norman Margolus. *Cellular Automata Machines*. MIT Press, 1987.

[Tof08] Tommaso Toffoli. Background for the Quad prize. `http://uncomp.uwe.ac.uk/automata2008/files/quad.pdf`, 2008. To appear in Journal of Cellular Automata.

[TS90] Daisuke Takahashi and Junkichi Satsuma. A soliton cellular automaton. *Journal of the Physical Society of Japan*, 59(10):3514–3519, 1990.

[vNB66] John von Neumann and A. W. Burks. *Theory of self-reproducing automata*. University of Illinois Press, 1966.

[WL92] Andrew Wuensche and Mike Lesser. *The Global Dynamics of Cellular Automata: An Atlas of Basin of Attraction Fields of One-Dimensional Cellular Automata*. Addison Wesley, 1992.

[Wol83] Stephen Wolfram. Statistical mechanics of cellular automata. *Reviews of Modern Physics*, 55(3):601–644, 1983.

[Wol84] Stephen Wolfram. Universality and complexity in cellular automata. *Physica D*, 10:1–35, 1984.

[Wol86a] Stephen Wolfram. Random sequence generation by cellular automata. *Advances in Applied Mathematics*, 7:123–169, 1986.

[Wol86b] Stephen Wolfram, editor. *Theory and Applications of Cellular Automata*. World Scientific, 1986.

[Wol94] Stephen Wolfram. *Cellular Automata and Complexity: Collected Papers*. Westview Press, 1994.

[Wue97] Andrew Wuensche. Attractor basins of discrete networks. D.Phil thesis, University of Sussex, 1997.

[Wue02] Andrew Wuensche. Finding gliders in cellular automata. In Adamatzky [Ada02], chapter 13, pages 381–410.