An Exploration in Recursive Cellular Automata

Joshua Underwood

School of Physics and Astronomy,
Queen Mary, University of London,
London

April 11, 2014

Abstract

We present results from an experiment conducted using a recursive cellular automata, in which the rule used to update the universe is also updated at each new iteration. The results follow a brief review of prior studies on elementary cellular automata from which the key ideas have informed the methods utilised in the experiment. It was discovered that the behaviour exhibited by the recursive cellular automata differs vastly from the behaviour found in elementary cellular automata. Furthermore, a brief survey of the rulespace for larger recursive cellular automata indicates the possibility that more complicated and varied behaviour emerges in larger systems.

Contents

1 Introduction **3**
- 1.1 Rationale . . . 3
- 1.2 Architecture of Elementary CA . . . 3
- 1.3 Phenomenology and Behaviour of CA . . . 5
- 1.4 Statistical Analysis of CA . . . 6
  - 1.4.1 Local statistical properties . . . 7
  - 1.4.2 Global Properties . . . 8
- 1.5 Cellular Automata as Dynamical Systems . . . 10
  - 1.5.1 The Basin of Attraction Field . . . 11
  - 1.5.2 Hamiltonian for ECA . . . 12
- 1.6 Class 4: The Edge of Chaos . . . 12

2 Recursive Cellular Automata **14**
- 2.1 Characteristics and RCA Dynamics . . . 14
- 2.2 Method . . . 14
- 2.3 Results . . . 17
  - 2.3.1 Density of 1 States . . . 20
  - 2.3.2 Basin of Attraction Field . . . 24
  - 2.3.3 Basins of Attraction . . . 25
  - 2.3.4 Divergent Trajectories . . . 28
  - 2.3.5 Sampling the $5 = n$ RCA Rulespace . . . 29
- 2.4 Discussion . . . 29
- 2.5 Conclusion and Future Research . . . 30

A Data, Basins of Attraction and Source Code **32**
- A.1 Tables of Compared Trajectories . . . 32
- A.2 State transition graphs . . . 33
- A.3 Source Code . . . 38

B References **44**

1 Introduction

Cellular automata (CA) provide an example of a discrete, dynamical system of simple cells that evolve in parallel according to a simple rule. Physical models tend to be described by differential equations that model deterministic behaviour. CA with simplistic architecture provide models with the power and scope to produce highly complex behaviour from a very basic set of rules and definitions. Traditional mathematical methods tend to provide descriptions of particular nuances to the system but are not well suited to handling problems of higher order complexity, for instance gravitational three body problems have caused issues and required the use of approximation or direct numerical computation. [1]

CA offer an alternative and computational tool for understanding physics, owing to their intrinsic simplicity and potential to produce highly complex behaviour. Computational analysis techniques are often used to provide solutions to complex problems and CA provide a method that is intrinsically computational. [2]

CA differ from differential equations in that space and time are discrete coordinates, not continuous ones. They are used to conveniently model physical systems that are made up of many discrete elements, such as a gas systems. CA are not exclusively used to model physical systems and can be used as models of biological systems (as per their original intention by von Neumann and Ulam) [4]. The complex behaviours exhibited by some CA have in some cases proven to support universal computation.

1.1 Rationale

Presented in this paper is a brief overview of some of the key findings from previous studies of CA, some of which are then used to inform an investigation into a unique type of CA. Many different forms of CA exist, each with their own characteristic architecture and properties imposed by their construction. Later we shall address a type of recursive CA (RCA) that updates the ruleset with the current universe state and then updates the universe to the next iteration, applying the universe as the rule.

CA have been observed to exhibit highly complex behaviour from a relatively simple construction, unlike the early Von Neumann CA (2D, localised neighbourhood and 29 possible cell states) focus has been given to building simplified architectures that are still capable of producing complex behaviour. [6] The 3-neighbourhood RCA is built such that any initial state can only be applied to one specific transition function, as a starting state also provides a starting rule. The system can be considered highly limited as a result.

1.2 Architecture of Elementary CA

A cellular automaton, in the simple 1D case, is an array of cells of unbound size that each posses one of k states. Typically a binary state CA will use 0 or 1. We call this array the universe, U. The state of each cell is updated synchronously, meaning the entire set of cells is updated in one discrete time step. The evolution of each cell is determined by a simple deterministic rule represented in equation (1) as a boolean function, f, of the cell site and the 2 nearest neighbours a_i, a_{i-1}, a_{i+1}. [3] This is known as the transition

function.

$$a'_i = f(a_{i-1}, a_i, a_{i+1}) \tag{1}$$

The neighbourhood size, n, is one of the characteristic parameters of a CA. The cell being updated will have a rule selected from the ruleset according to the number that the neighbourhood represents. In other words, a binary number is generated by the neighbourhood, the decimal representation of this number serves as the index reference to the new state, that is contained in the rule set. For example, if we have a particular 3 neighbourhood given by {0,0,1}, the neighbourhood characteristic number is 001 in binary and 1 in decimal. The new state is found by referring to the 1st element of the ruleset.

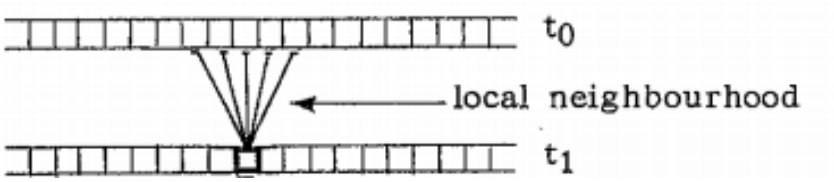

Figure 1: Diagram of a local neighbourhood of size $n = 5$. The array labelled t_0 is the universe state U_0, the target cell is updated by referring to the ith entry in the ruleset array, where i is the decimal representation of the neighbourhoods characteristic binary number. [8]

Certain limits are imposed on the number of elements contained in the ruleset. It must be possible for any permutation of the neighbourhood to refer to an element in the ruleset, so the minimum size of the ruleset is given by $R_{size} = k^n$. Though it is possible to have a ruleset larger than this, elements indexed by numbers larger than k^n will not be referenced and are thus unnecessary.

The number of possible distinct rules is again limited to the size of the neighbourhood and the number of possible states. The number of ruleset permutations is given by k^{k^n}, in the case of a $n = 3$ CA this is simply $2^8 = 256$.

The time evolution of a CA is characterised by successive applications of the transition rule on U_n. The genealogy of global states is considered the trajectory. For all CA, provided that the update is free of errors, then there is one unique trajectory for any given initial state and ruleset. [6]

The focus in this paper is on CA that have ordered and local architectures. The neighbourhood is constructed by taking the immediate neighbours either side of the cell that is being updated. This locality is imposed by the construction and it is possible to build CA neighbourhoods that take non-local cells according to a set regime (wiring) or by a regime that changes at each iteration. [6] An ordered CA retains the same neighbourhood structure and rule for each iteration. Wolfram ECA are ordered, and the RCA is disordered with regards to the ruleset, where the rule is also updated at each iteration. All CA discussed in this paper are deterministic.

A further feature of CA architecture is the boundary conditions on the universe, particularly how the neighbourhoods are constructed for the cells that sit on either edge. It

is possible to build a CA that will automatically update the cells on the edge to a given state, decided arbitrarily, without applying the regular transition function. Alternatively, the universe can be constructed with a circular topology, such that there are periodic boundaries. Periodic boundaries allow for the transition function to work without any arbitrarily defined conditions at the edges. For example, on the right edge, the update function f will take a neighbourhood defined as a_i, a_{i-1}, a_0. The circular topology was necessary in the construction of the RCA, but it is optional in ECA.

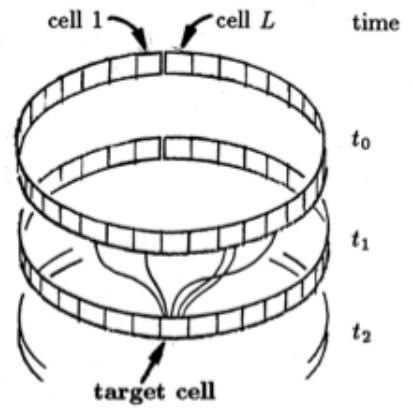

Figure 2: Diagram of the circular topology, for a CA with with non-local "wiring" and neighbour size $n = 5$. [6]

The dimension of a CA is given by the universe array dimension. The array dimension is the number of indices required to specify an element (or cell) within the array. The dimension is not the number of cells contained within the universe array.

For 1D CA, a space-time diagram is equivalent to the evolution genealogy of the system. Drawing the current universe iteration beneath the previous universe iteration yields a record of the trajectory. In ECA, such genealogies provide the phenomenology upon which empirical classifications have been made for different CA Rules. [6] For instance, the Wolfram ECA Rule 90 visually reproduces the Sierpinski Triangle, a fractal structure that can clearly be seen to exhibit the property of self similarity. [5]

Wolfram ECA impose the quiescent rule and a second restriction. The quiescent rule imposes that a null state remains null upon the next iteration, effectively preventing creation from nothing. The second restriction is that rules must be reflection symmetric, so neighbourhoods with characteristic numbers as 011 and 110 will yield the same time evolution. These restrictions leave a set of 32 rules, of the form $\alpha_1\alpha_2\alpha_3\alpha_4\alpha_2\alpha_5\alpha_4 0$. These restrictions are ultimately not necessary in defining a CA. [4] Such restrictions have not been considered appropriate for the RCA.

1.3 Phenomenology and Behaviour of CA

ECA built according to the architecture as discussed exhibit a number of behaviours [6], including:

- The universe at t_0 evolves into one and only one successor. It is not possible for there to be divergent evolutionary paths for any one universe state.
- There are a number of universe states that can be obtained from any number of previous universe states. It is possible for 2 predecessor universe states to evolve into an identical successor. This indicates convergent trajectories in the phase space of the CA system.
- A number of universe states can only ever be reached via manual input. They can never be attained as the successor to any universe state. These are the so called Garden of Eden (GoE) states.
- After a finite number of updates, the system will settle on an attractor cycle.

CA, despite their relatively simple construction, have the capacity to exhibit considerably complex behaviours. It is possible to directly observe in their space-time patterns behaviour such as self similar organisation in the form of fractal patterns, chaotic behaviour, random behaviour and even computation. [5] The behaviours can be placed into 4 broad categories, proposed by Wolfram:

- Homogeneous (Class 1): any initial randomness quickly decays into a stable, homogeneous state, a fixed point final state configuration.
- Stable Oscillators (Class 2): similar to Class 1 but oscillating, periodic structures emerge, it is comparable to a limit cycle final state configuration.
- Chaotic (Class 3): appearance of pseudo-random behaviours, stable structures are quickly destroyed. The behaviour is similar to strange attractors.
- Complex (Class 4): Stable structures emerge and are not quickly destroyed, the structures interact in interesting ways, Wolfram conjectured that Class 4 CA are capable of universal computation and this has been proven for rule 110. Class 4 are considered to be an intermediary of Classes 2 and 3 and are therefore rare.

This classification system is entirely empirical and open to interpretation. It is important to demonstrate these classifications, to further develop the idea of modelling dynamical systems with CA, as the observed behaviour is indicative of the nature of the attractor.

1.4 Statistical Analysis of CA

The complexity and range of behaviour exhibited by CA has been quantified by a localised or global approach focussing on statistical properties of CA time evolution, in research conducted by Wolfram in 1983. By taking the localised approach we aim to characterise the self organisation that is apparent on empirical observation of ECA. The global approach considers the phase space of the system, in which all possible configurations of a CA are contained.

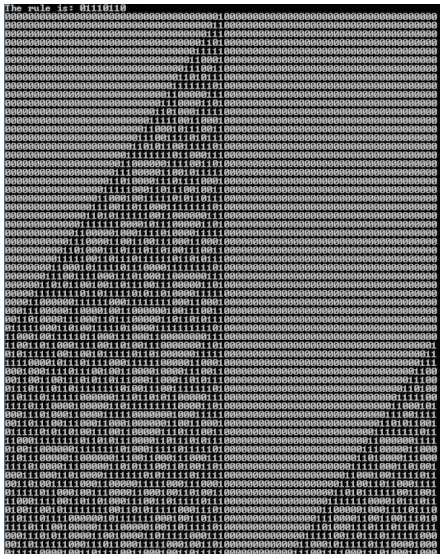

Figure 3: Wolfram ECA Rule 110, displaying Class 4 behaviour. Stable structures can be seen though their appearances are not predictable.

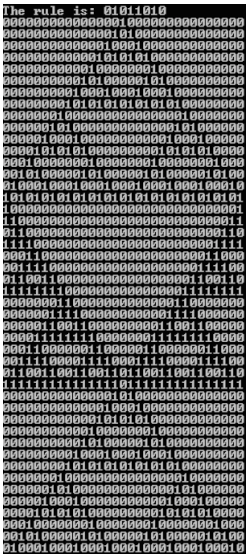

Figure 4: Wolfram ECA Rule 90, the space-time pattern reproduces the Sierpinski Triangle when the initial universe state consists of the central cell in the 1 state.

1.4.1 Local statistical properties

Local analysis starts by selecting a universe configuration that is disordered or random. A configuration is disordered if there is no statistical correlation between the sites (or cells) and each sites behaves as an independent random variable. Only one parameter is needed to describe a completely disorganised configuration, the probability for a site to be valued 1, denoted by p. [4] The most basic statistical quantity that can be found for any configuration is the state density ρ_τ, where τ is the number of time steps. Here we shall review the case for small τ.

For a disorganised state, where $\rho = p = 1/2$, we have a random configuration containing all eight possible three site neighbourhood configurations, each with equal probability. A single time evolution yields a new configuration in which the fraction of sites in state 1 is given by:

$$\rho_1 = \#_1(\boldsymbol{R})/(\#_0(\boldsymbol{R}) + \#_1(\boldsymbol{R})) = \#_1(\boldsymbol{R})/8 \tag{2}$$

$\#_1(R)$ is a function that simply counts the number of 1 states in the binary representation of the rule configuration. Though this result is specific to an initial state where $p = 1/2$ it may be generalised to initial states where this is not the case. This is achieved by using the $p(\sigma) = p^{\#_1(\sigma)}(1-p)^{\#_0(\sigma)}$ for each possible three neighbourhood configuration and adding the probabilities of the σ that yield value 1 on application of the rule.

The behaviour of the density in the limit of large τ is not described by the equation yielded for small τ. However, a similar approach can be used to derive statistical approximations of the behaviour. Some simple rules always evolve to null state rho = 0 regardless of starting configuration, though the length of transient states varies. For the

case of Rule $0 = \rho = 0 \,\forall\, \tau > 0$. this is true for rule 72 at $\tau > 1$.

Complex CA have a definite ρ_∞, independent of initial density ρ_0. These are the so called limiting densities. By ignoring the fact that CA evolution generates correlations between state values at different cells, all configurations may characterised by their average density ρ, or by p and $q = 1 - p$, with p and q assumed to be independent for each site. Following the Markov Approximation, a master equation may be derived:

$$\delta\rho/\delta\tau = \mathbf{\Gamma}(0 \to 1) - \mathbf{\Gamma}(1 \to 0),$$

$$\mathbf{\Gamma}(0 \to 1) = \boldsymbol{P} \cdot (00110011 \wedge \boldsymbol{R}),$$

$$\mathbf{\Gamma}(1 \to 0) = \boldsymbol{P} \cdot (11001100 \wedge \boldsymbol{R}),$$

$$\boldsymbol{P} = p^3, p^2q, p^2q, pq^2, p^2q, pq^2, pq^2, q^3$$

Where $\mathbf{\Gamma}(0 \to 1)$ is the average number of sites that change from the 0 state to the 1 state in each update. $\boldsymbol{R}$ is the characteristic binary number that specifies a CA rule. $\boldsymbol{P}$ is the probability vector for possible 3 neighbourhoods, with each site assumed to have independent probability $p = \rho$ for the value to be 1.

At equilibrium, the condition $\delta\rho/\delta\tau = 0$ is satisfied. This allows for polynomial equation solutions for p and ρ_∞. For the complex CA rules, approximate values, acquired via the master equation, of the equilibrium density are found to be within $10-20\%$ of the exact values. This is a result of the approximation that ignores the correlations between sites created by the application of CA rules, or the effect of "feedback".

The discrepancy indicates that application of CA rules creates correlations between cells. By using a correlation function the order exhibited by the CA evolution may be quantified. Long sequences are found to emerge from disordered initial states. This is indicated by the appearance of triangle structures in the time evolution of complex CA rules, such as Rule 146. [4]

1.4.2 Global Properties

By considering the set of all possible CA configurations for a given CA an analogy may be formed to classical statistical mechanics and dynamical systems analysis. The number of possible configurations of a CA is dependent on the array length of a given universe, N, and the number of possible states. For a binary state CA with a universe of size 8 this is simply $2^8 = 256$. When considering the binary representation of CA configurations within the phase space, the Hamming distance, $H(S_1, S_2)$, is an appropriate means of measuring distance. [4]

The Hamming distance is defined as the number of digits which differ between the binary forms of 2 states, S_1 and S_2. In the phase space, a possible configuration corresponds to a point. When the transition function is applied to an initial configuration, a trajectory is formed in time between configurations. For simple CA, it is found that the Hamming distance tends to small constant values. For complex CA however, the trajectories of most initial configurations with a small Hamming distance were found to have exponentially divergent trajectories.

Points within the phase space may be arbitrarily assigned a probability. In a set of disorganised states with $\rho = 1/2$ each configuration is equiprobable. These probabilities are modified on the application of a CA rule, giving rise to organisation. Some configurations will be modified to have zero probability and varying non-zero probabilities. The surviving configurations compose the equilibrium ensemble, which gives rise to average local features in local equilibrium configurations.

Local irreversibility of elementary CA is a feature indicated by the trajectories from different initial starting points leading to identical end points. Particular configurations may be reached from a number of different preceding configuration but does not necessarily evolve into more than one configuration on the next time step. The splitting of trajectories is not observed, yet the joining is. For instance, applying rule 0 to any possible initial configuration will always lead to the null configuration. The coalescence of trajectories gives rise to the self organisation evident in CA evolution genealogies. [4]

As a result of this behaviour, certain configurations may only be available as an initial configuration and are not attainable along any CA evolutionary path. These so called "Garden-of-Eden" (GoE) configurations may never be attained for $\tau > 1$ and have a zero probability in the ensemble.

The entropy in a CA system can be considered a measure of the information content in the CA evolution. In 1D systems, the definition of entropy closely resembles that of a continuous dynamical system. [7] The entropy is defined as:

$$S = \sum_i p_i \log_2 p_i \tag{3}$$

where p_i is the probability for state i.

Given an infinite universe size only a small fraction of states attained by CA evolution comprise the phase space. There are considerably more Garden of Eden states. The architecture of 1D ECA is irreversible, as the universe state U_n depends only on U_{n-1}. The diffusion equation, a first order time-dependent differential equation is similarly irreversible. [4]

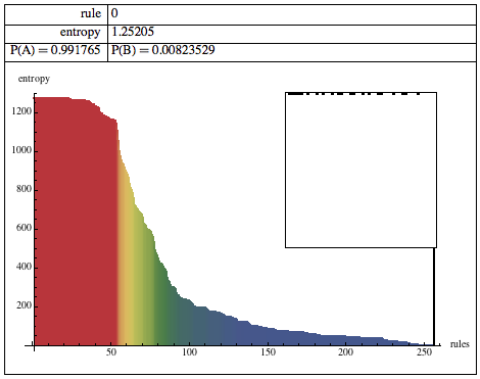

Figure 5: ECA Rule 0, depicted on a graph where the rules are ordered by decreasing entropy, $P(A)$ is the probability of a white cell[16]

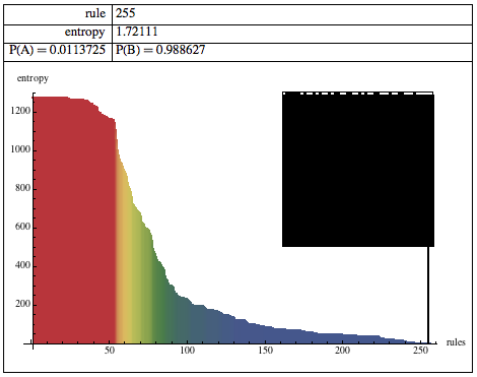

Figure 6: ECA Rule 255, depicted on a graph where the rules are ordered by decreasing entropy [16]

The entropy is lower in ordered systems and higher in disordered systems. [16]

1.5 Cellular Automata as Dynamical Systems

CA are simple, discrete (spatially and temporally) dynamical systems that evolve according to a deterministic rule. By considering the cellular universe of the CA an analogue to the spatial extents in our universe and by considering the transition function an analogue to the physical laws that govern the universe, CA can be considered a basic simulation (or model) of certain physical systems.

Dynamical Systems theory proposes that long term trajectories can be categorised into three regimes based on their geometry. The three regimes are thought to be the result of an attractor, a potential force to which the state transition trajectory evolves [15]. The different types of attractors are:

- fixed point attractors: Characterised by a trajectory that leads to a single point in phase space. In general, it can be considered a limit cycle with a period of 1, since application of the transition function yields an identical state to the input.
- limit cycle: Characterised by a trajectory that leads to a stable closed set of points, that oscillate with a set period. It can be considered a periodic attractor.
- strange attractor: Characterised by behaviour that never seems to stabilise or become predictable. However, the trajectories are restricted to a bounded manifold. The strange attractor possesses an intricate and complex geometry, or a fractal structure. Initial states of infinitesimal separation can produce significantly divergent trajectories, the property of so called "sensitivity to initial conditions."

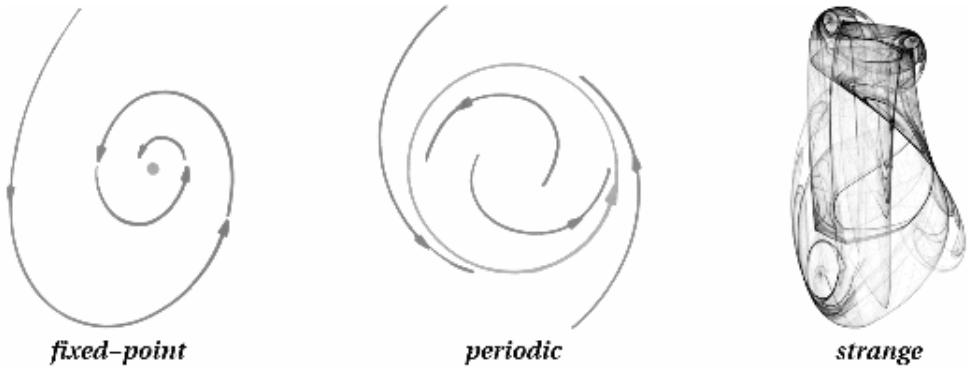

Figure 7: Geometric representation of attractor classes [15]

According to the Wolfram regime, we can maintain that Class 1 CA evolve to fixed point attractors whereas Class 2 CA evolve to limit cycles. Class 3 CA evolve to strange attractors and Class 4 CA exhibit behaviour that cannot be immediately compared to any of the known attractors, characterised by long transients.

1.5.1 The Basin of Attraction Field

A basin of attraction is the set of all trajectories that lead to a given attractor a state transition diagram that displays the complete set of trajectories that lead to an attractor can be considered a visual representation of the basin of attraction for that attractor. In the state transition diagram, a node may have any number of "pre-images" (predecessor states) but, owing to the deterministic nature of our CA, a node may never have multiple successors. [8] Trajectories outside of the attractor cycle, leading from GoE nodes, are the transients. The basin of attraction field links every element of the state space according to the dynamics of the system. The basin of attraction field is imposed on the state space by the system. [9]

As a trivial example, the basins of ECA Rule 0 or Rule 255 will simply depict a fixed point attractor, labelled 0 or 255 respectively, with all the pre-images as immediate neighbours.

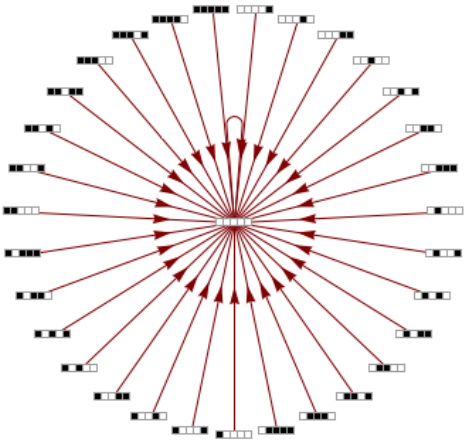

Figure 8: Basin of attraction for 3 neighbourhood, universe size 5 ECA Rule 0[13]

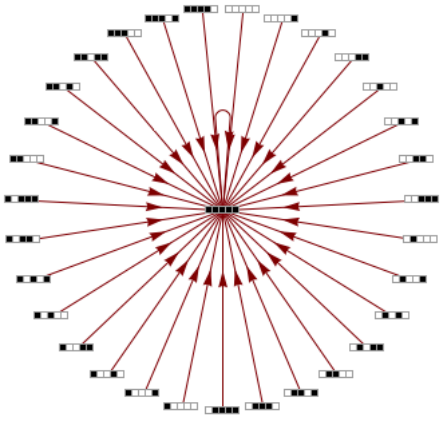

Figure 9: Basin of attraction for 3 neighbourhood, universe size 5 ECA Rule 255[13]

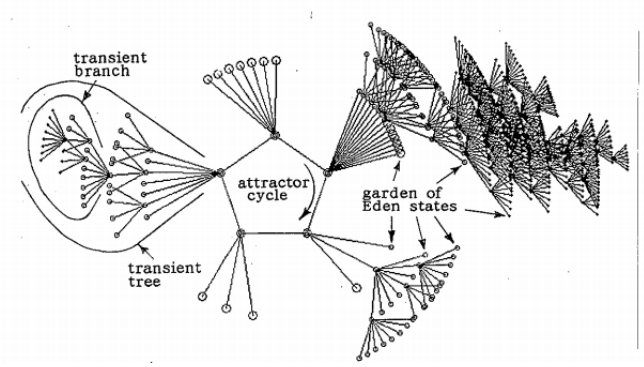

Figure 10: An example of a single basin of attraction with a limit cycle, with the key features labelled [8]

The basin of attraction field represents a second order of complexity for CA systems, where the first is considered the space-time diagrams of the system. Every possible interaction between the given universe and ruleset can be viewed synoptically providing potential insight into the nature of the system and the behaviour of the rule. [10] Later, it will become clear how this approach benefits the study of the RCA, where the global dynamics can be mapped quite easily owing to the intrinsic limitations of the system. Otherwise, reverse algorithms are constructed in order to discover the trajectories that, when reversed, lead to the GoE states.

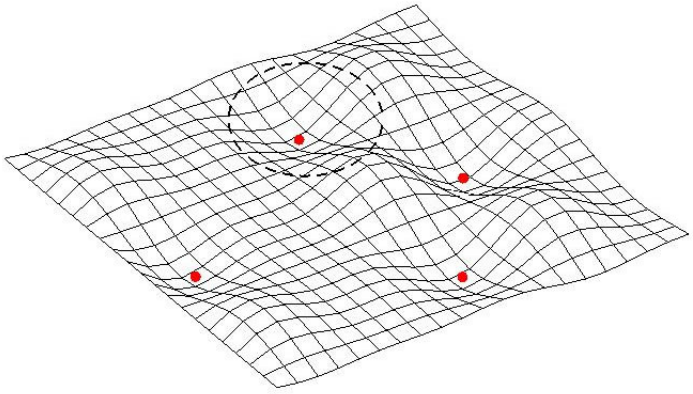

Figure 11: An example of a basin of attraction field for a Hopfield Network (a type of neural network), comprised only of point attractors [17]

1.5.2 Hamiltonian for ECA

A brief attempt to find the Hamiltonian function for a Wolfram ECA system was made. The initial step was to treat the state as a vector $\vec{q_n}$ and to identify a parameter akin to velocity in physical systems, the rate of change of configuration $\dot{\vec{q_n}}$.

Next, a matrix that represents the transition function, $\vec{R}$ was required, such that $\vec{q_n} \times \vec{R} = \vec{q_{n+1}}$. However, it became apparent that it is possible that $\vec{R}$ depends on $\vec{q}$. The question that remains to be answered is whether there can be a constant $\vec{R}$ that always yields the correct time evolution, independent of $\vec{q_n}$.

The Hamiltonian function would provide an equivalent to a potential found in physical systems and could perhaps provide the basis for more generalised, further comparisons or simulation.

1.6 Class 4: The Edge of Chaos

Langton (1991) conducted a study into the computational properties of Class 4 CA, seeking to identify the conditions under which computation is supported in a dynamical system. The features required to support computation are the storage, transmission and modification of information. [18]

Langton defined a parameter, λ that is defined as the fraction of rules in which the

updated cell is in the 1 state.

$$\lambda = \frac{k^n - N}{k^n} \tag{4}$$

where k is the number of different cell states, n is the neighbourhood size and N is the number of transitions to the quiescent state, s_q (an arbitrarily chosen state). [18]

For $\lambda = 1.0$ any neighbourhood that contains a 1 state will update the target cell to the 1 state. For $\lambda = 0.0$ then all cells will update to 0. Values of λ between 0 and 1 provide a means of sampling the rule space of a CA. Langton found that between $\lambda = 0.05$ and $\lambda = 0.15$ transients emerge, with increasing length as λ increases. At $\lambda = 0.20$ permanent periodic structures can emerge and at $\lambda = 0.25$ the structures may appear with a period of 1 (appearing fixed).

As λ increases, so does the length of transients, however at $\lambda = 0.45$ some rules have behaviour including propagating structures that undergo orbits around the circular universe with a given period.

As λ is raised above 0.50, the behaviour becomes less periodic with a tendency towards chaotic behaviour. Transient lengths begin to shorter for regimes with $\lambda > 0.55$. As we approach $\lambda = 0.75$ the long term behaviour is attained in fewer iterations and tends to represent chaotic behaviour (characterised by the density of 1 states being within 1% of the long term mean). [15]

Rules in the $0.45 > \lambda < 0.55$ provide the 'transition regime'. The final behaviour of rules in this regime represent a phase transition between periodic and chaotic behaviour, directly supporting Wolfram's hypothesis that Class 4 CA exist as the intermediary of Class 2 and 3. [18]

2 Recursive Cellular Automata

The novel type of CA introduced in this departs from the Wolfram ECA in that the collection of cell states that comprises the universe are used to update the rule at each new iteration step. At the start of a new iteration the ruleset is set to the same value as the universe.

$$R_n = U_n \tag{5}$$

Following the input of an initial state and universe, the CA is updated synchronously and once it has arrived at the new state the ruleset is updated to the same value as the universe state.

2.1 Characteristics and RCA Dynamics

Adopting the language of physical dynamical systems, the RCA is a system under which the "physics" changes alongside the state. This represents a departure from real world physical systems, where it is assumed that the physical law that governs state transitions is independent of the state itself.

The system is irreversible, given that the transition function references only U_n and not U_{n-1}. The universe array size is bound by the neighbourhood size. Here a system with neighbourhood size $n = 3$ has been studied, therefore the universe size is bound to $2^3 = 8$. Given this restriction and the coupling of the rule to the state, construction of the basin of attraction field requires nothing more than running the system for all 256 possible universe states, with no reverse algorithm required. In principle, this also applies for a 5-neighbourhood RCA, but the drawing of the state transition graphs would ideally be automated by software.

Furthermore, since initial conditions cannot be arbitrarily applied to any rule. It is possible to create a complete map of the phase space of the RCA, without any consideration of arbitrary initial conditions. As a result, sampling of random initial conditions is not possible. This allows for a systematically complete set of trajectories with relative ease.

It can be argued that "sensitivity to initial conditions" is a feature built directly into the RCA. Whereas in ECA, this feature is observed for independent rulesets. For this reason, it is important to observe the second order of complexity offered by the basin of attraction field. Since the rule changes with the system, the typical space-time diagrams studied extensively in the past may not provide adequate insight into the emergent behaviours of the RCA.

The only attractors known a priori are the ones with decimal encoding 0 and 255, $\{0, 0, 0, 0, 0, 0, 0, 0\}$ and $\{1, 1, 1, 1, 1, 1, 1, 1\}$ respectively. This is owing to the fact that these rulesets yield a universe state that is identical to the rule.

2.2 Method

As the RCA had not been hitherto investigated, a purely explorative approach was required. The study was conducted by initially producing output streams of the CA for

all possible ruleset configurations. These trajectories were later drawn in state transition diagrams and were used to compile data regarding the different attractors. The longest length of transient, shortest length of transient, number of Garden of Eden states that tended to the attractor, the average length of transient, number of states that tended to the attractor and the density of sites in the 1 state were the initial data taken from these diagrams.

A highly global approach was considered most appropriate to explore the RCA. This required analysis of the phase space, basins of attraction and basin of attraction field. Little attention has been given to the space-time diagrams. This is a result of the unique architecture of the RCA.

The length of transients is an indication of the number of updates required before the trajectory reaches a stable cycle or point. Transients that are significantly longer than the average, for a given attractor, indicate complexity for that particular trajectory.

The number of tier 1 branches (or near predecessors) is a measure of the in-degree for any given attractor. It is not limited to attractors but in this study, only the in-degree of each attractor was measured. In a previous study, for very ordered rules the in-degree is expected to be comparatively larger compared to disordered rules. [11] Here it can be argued that the in-degree represents how ordered a "rule trajectory" is, simply by observing the in-degree of the attractor.

Nearby states, defined as having a Hamming distance separation of 1 (since it is not possible to have infinitesimal separation), are tested for chaotic behaviour via calculation of the Lyapunov exponent. The approach taken deviates from other proposed methods in that the average is not computed in the limit that $t \to \infty$, but is instead taken up to the number of transitions required for both trajectories to be on the attractor cycle. Owing to the limitations of the system, any nearby trajectory is considered and not merely those that exist within the same basin of attraction, otherwise all the results would trivially be equal to 0. This provides a simple indication of the chaos present in the system.[14] The exponential divergence of initially nearby trajectories is considered a means of quantifying chaotic behaviour exhibited by the system.

A simple definition of the entropy of a given state is the number of pre-images that lead to that state, or the sum of the predecessors. Since the system is entirely deterministic and all trajectories are known, this is an appropriate way to measure the entropy. The entropy provides insight into the information content contained in each basin of attraction. Higher entropy attractors are considered to be more disordered and a naive assumption is that lower entropy attractors exhibit higher complexity.

The density of GoE states within a given basin of attraction is also measured. Owing to the existence of these GoE states, the system is not surjective. Furthermore, the system was found to not be injective since there are nodes with multiple predecessors.

The state transition diagrams were united for each attractor, giving a visual representation of the separate basins of attraction. The Hamming distance of each attractor was measured, providing an indication of the relative locations of the attractors. This has provided the first steps to describe the basin of attraction field of this particular system. [11] The Hamming distance between each attractor was also calculated. Previous work on random Boolean Networks by Apolloni et al. has suggested that ordered systems tend to have clustered attractors, whereas chaotic systems tend to have scattered attractors.

[12] The mean Hamming distance separation for each attractor was calculated to provide indication as to whether the attractors are clustered or scattered.

The density of cells in the 1 state of attractor was also considered an important property, taking cues from the ideas presented by Wolfram. This approach allows for the exploration into statistical properties of the attractors. One could naively predict that the mean density will be 0.5, as the obvious attractors are 0 and 255.

Finally, a brief sample of data taken from a $n = 5$ RCA is presented, particularly any attractor that displays behaviour not found in the $n = 3$ RCA.

2.3 Results

15 attractors were discovered to exist for this system. Table 1 presents system-wide mean values regarding attractor properties. Table 2 presents specific properties of the attractors.

Number of Attractors	Mean number of nodes to an attractor	Mean number of starting points to an attractor	Mean path length of all attractors	Mean longest path length	Mean shortest path length	Mean density of 1 state sites
15	16.067	8.1333	2.855	4.467	1.6	0.5

Table 1: System-wide mean values of attractor properties

The initial prediction has been confirmed, the mean density of 1 sites, $< \rho >$ is 0.5. However, it was only possible to predict that there would be at least 2 attractors.

ID of Attractor	Max Path Length	Min Path Length	Mean Path Length of attractor	Near predecessors	Total Nodes	Nodes (excluding GoE)	GoE nodes	G-density	ρ	Symmetrical
0	12	1	5.07	7	60	32	28	0.467	0	yes
4	4	3	3.67	1	6	3	3	0.5	0.125	no
36	5	5	5	1	5	4	1	0.2	0.25	yes
68	4	1	2.5	2	5	3	2	0.4	0.25	no
76	5	1	3.17	2	17	5	12	0.706	0.375	no
77	3	1	1.5	4	10	6	4	0.4	0.5	no
85	3	1	2.2	4	6	1	5	0.833	0.5	no
132	6	3	4.5	2	21	11	10	0.476	0.25	no
204	3	1	1.75	4	7	3	4	0.500	0.5	no
205	3	1	2.14	6	18	4	14	0.778	0.625	no
219	4	2	3.33	2	7	2	3	0.429	0.75	yes
221	3	1	1.67	3	5	4	3	0.6	0.75	no
222	1	1	1	1	1	0	1	1	0.75	no
223	1	1	1	2	2	0	2	1	0.875	no
255	10	1	4.33	8	71	41	30	0.423	1	yes

Table 2: Properties of the discovered attractors

By comparing specific properties of attractors to the system-wide mean values, we can make a first attempt at quantifying complex or disordered behaviour within the system.

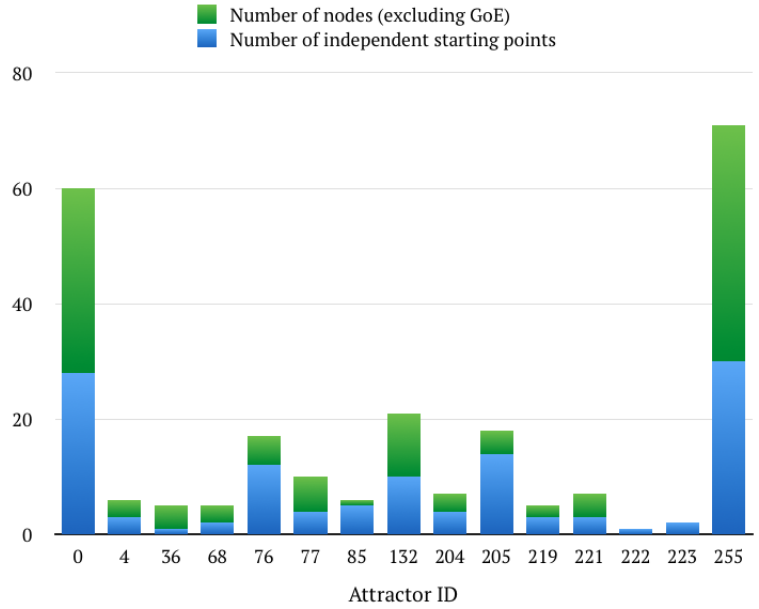

Figure 12: Bar chart depicting the total states that lead to any given attractor, a distinction has been made between the Garden of Eden states and the states with pre-images.

The entropy for each attractor can be observed in figure 12. It was known initially that both 0 and 255 would be attractors, but it was not clear how many of the possible universe states would lead to them. Interestingly, there is an uneven distribution, with 255 possessing 11 more pre-images than 0. However, there are only 2 more GoE pre-images that lead to 255 compared to 0. 255 and 0 are the highest entropy attractors with 222 being the least entropic.

The other 2 symmetric attractors, 36 and 219, are another pair of obvious interest. They are separated by $\rho = 0.5$ as the configuration space is simply an inversion of states. 36 is represented by $\{0, 0, 1, 0, 0, 1, 0, 0\}$ and 219 is represented by $\{1, 1, 0, 1, 1, 0, 1, 1\}$.

ID of Attractor	Max Path Length	Min Path Length	Mean Path Length	Near predecessors	Total nodes	Nodes (non-GoE)	GoE Nodes	G-density	ρ
36	5	5	5	1	5	4	1	0.2	0.25
219	4	2	3.33	2	7	2	3	0.429	0.75

Table 3: Comparison of the 2 reflection symmetric attractors, 36 and 219.

36 is closer to 0 in terms of ρ, with a difference of 0.25. 219 is closer to 255, also with a difference of 0.25. 36 has an entropy of 5 whereas 219 has an entropy of 7. Furthermore, 36 has a longer max path length, similar to how 0 has a longer max path length when

compared to 255. However, the G-density is lower in 36 compared to 219, whereas 0 has a higher G-density compared to 255.

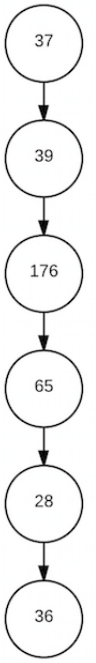

Figure 13: Basin of attraction for RCA Rule 36

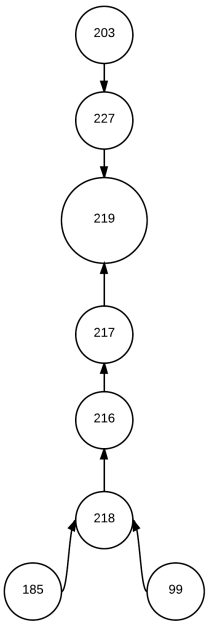

Figure 14: Basin of attraction for RCA rule 219

It appears that there are some similarities between the symmetrical attractors that are closer in regards to their ρ values.

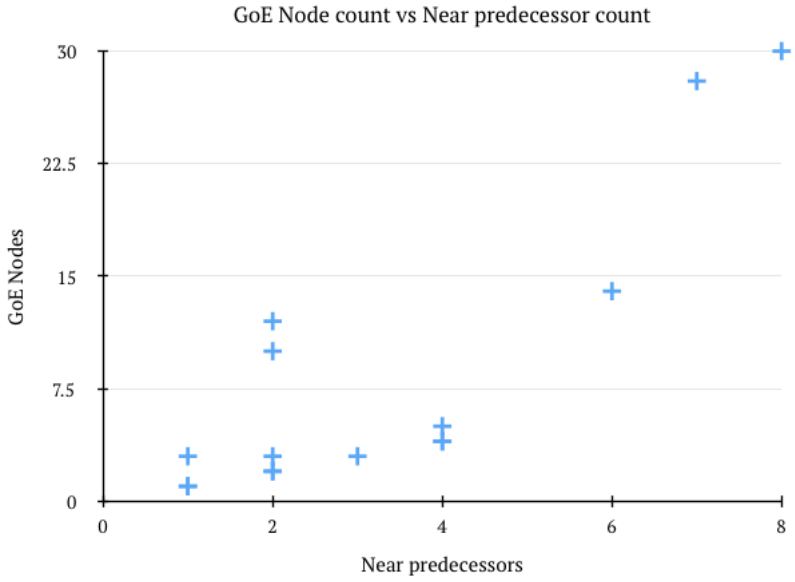

Figure 15

An attempt to find a trend between the GoE count and the near predecessor count for the attractors was made. Given the limited number of attractors, this may only be seen as a first step to find a trend. A linear trend would suggest that a similar structure is maintained by all attractor basins (as the GoE count should remain directly proportional to the near predecessor count), but outliers imply that this is not the case. A larger system with more attractors would be required to make a stronger attempt at finding a trend. Further analysis into the branching of specific transient trees would provide greater insight into the structural variation of the attractor basins.

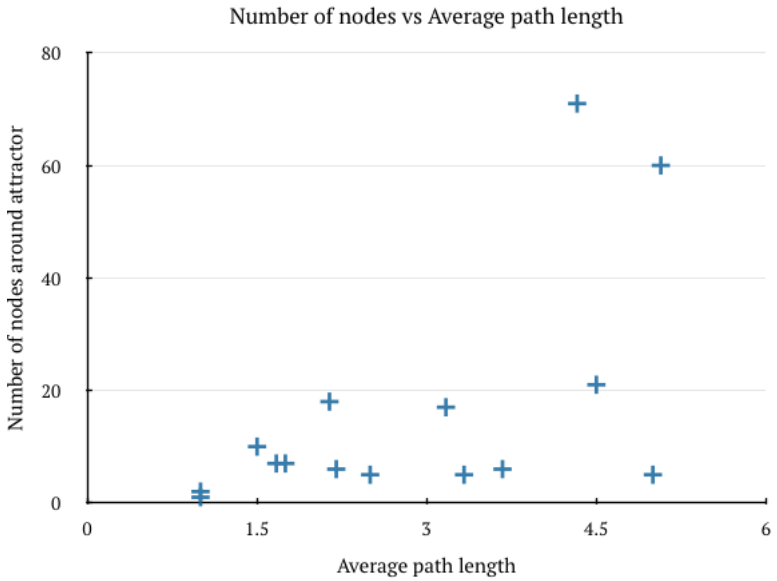

Figure 16

By comparing the number of nodes leading to an attractor (entropy) to the mean path of the attractor, no clear trend emerges. However, it is observed that the largest entropy is not correlated to the largest path length. This indicates structural variation between the basins of attraction, with long transients emerging regardless of the number of states that lead to the attractor cycle.

2.3.1 Density of 1 States

The attractors have been grouped in terms of the density of 1 states, ρ. This approach allows a comparison between attractors that differ only in regards to ρ. There are 6 attractors with $\rho > 0.5$ and there are 6 attractors with $\rho < 0.5$, this approach allows insight into this apparent symmetry in the system.

ρ	Total Nodes	Total nodes excluding GoE	Near predecessors	Attractors of given density	Max Path Length	Min Path Length	Symmetrical attractors	Total GoE nodes	GoE density
0	60	32	7	1	12	1	1	28	0.467
0.125	6	3	1	1	4	4	0	3	0.500
0.25	31	18	5	3	6	4	1	13	0.419
0.375	17	5	2	1	5	1	0	12	0.706
0.5	23	10	12	3	3	1	0	13	0.565
0.625	18	4	6	1	3	1	0	14	0.778
0.75	13	6	6	3	4	1	1	7	0.539
0.875	2	0	2	1	1	1	0	2	1
1	71	41	8	1	10	1	1	30	0.423

Table 4: Properties of the density groupings

Every possible value of ρ has at least 1 attractor. Figure 17 reveals that the system is multimodal with respect to ρ, with modes of 0.25, 0.5 and 0.75. It was suggested that members of the set of $\rho = 0.25$ attractors may be inversions of members of the set of $\rho = 0.75$ attractors, 36 and 219 have already been shown to be related by this kind of transformation. It is also the case that elements can be related via an inversion followed by a reflection. 132 was found to be related to 222 by inverting the states and then reflecting. There was no obvious relationship observed between 68 and 221.

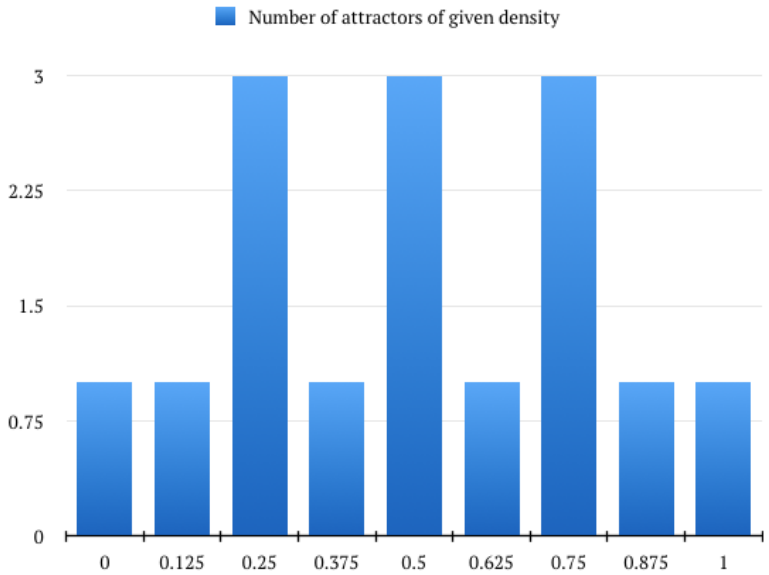

Figure 17: Bar chart depicting the number of attractors contained in each density grouping

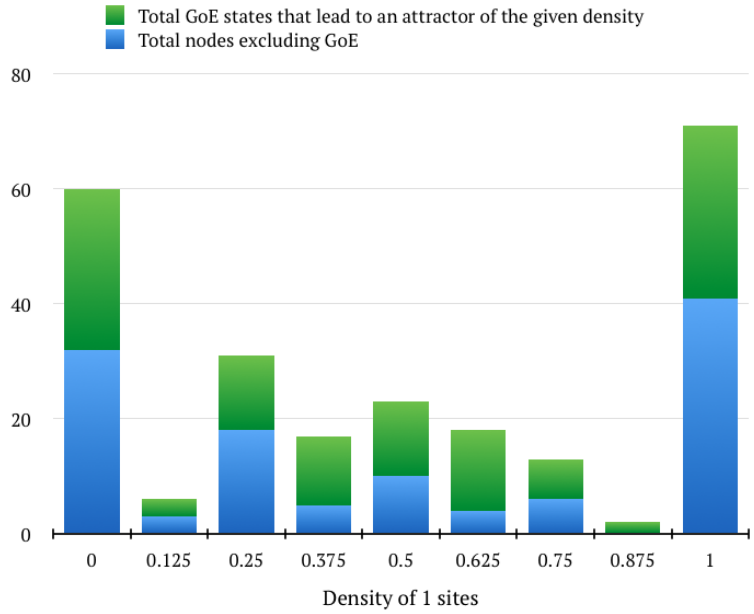

Figure 18: Bar chart depicting the number of nodes to each density group

The entropy of each attractor grouping (the sum of all nodes leading to attractors within the density group) has been presented in Figure 18. The group $\rho = 0.25$ has a larger entropy than $\rho = 0.75$, with 18 more nodes. However, the $\rho = 0$ group has 11 less nodes than $\rho = 1$. It is unclear why the groups either side of $\rho = 0.5$ have a different distribution of nodes amongst the groups. Furthermore, there are 114 nodes within the $\rho > 0.5$ groups, 10 more than the $\rho > 0.5$ groups.

For the $\rho < 0.5$ groups, the mean GoE density is 0.523 and the mean near predecessor density is 0.132, the ratio $\rho_{pred}/\rho_{Goe} = 0.252$. Whereas the mean GoE density is 0.685 for $\rho > 0.5$ and the mean near predecessor density is 0.212, the ratio $\rho_{pred}/\rho_{Goe} = 0.310$. These ratios give a crude indication of the branching present in each group. A higher ratio indicates that there is a smaller number of branches and therefore a higher degree of complexity. [11] However, since only the branching around the attractor has been taken into account, information regarding branching on sub-trees has not been considered, therefore this may only serve as a rough indicator.

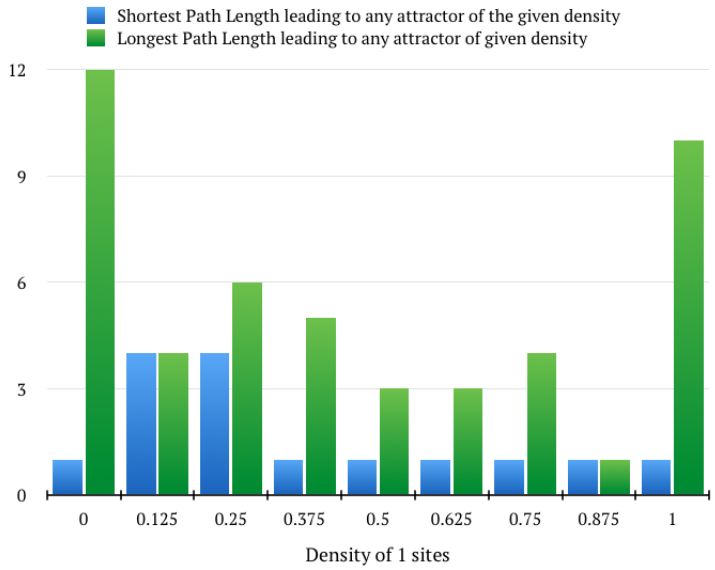

Figure 19: Chart depicting the maximum and minimum path lengths to each attractor

The longest transient in the system belongs in the basin of attraction for 0. Interesting transients occur for the densities 0.125 and 0.875 where the minimum is also the maximum. In the case of 0.125, this is owing to the fact that attractor 36 has only 1 transient path (with no branches), for 0.875, this is owing to the fact that there are only 2 near predecessors that are also GoE for attractor 223. The mean max path length for $\rho < 0.5$ is 6.75. For $\rho > 0.5$ it is 4.5. Higher ρ attractors seem to favour shorter transient lengths, indicating a higher degree of order. [11]

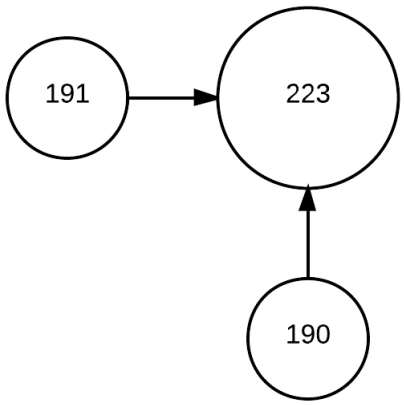

Figure 20: RCA attractor 223. Order is indicated by the short attractor length. Additionally, since there are few GoE that lead to this state (incidentally, separated by the minimum Hamming distance), it can be considered ordered.

2.3.2 Basin of Attraction Field

	0	4	36	68	76	77	85	132	204	205	219	221	222	223	255
0	0	1	2	2	3	4	4	2	4	5	6	6	6	7	8
4	1	0	1	1	2	3	3	1	3	4	6	5	5	6	7
36	2	1	0	2	3	4	4	2	4	5	8	6	6	7	6
68	2	1	2	0	1	2	2	2	1	2	6	4	4	5	6
76	3	2	3	1	0	1	3	3	1	2	5	3	3	4	5
77	4	3	4	2	1	0	2	4	2	1	4	2	4	3	4
85	4	3	4	2	3	2	0	4	4	3	4	2	4	3	4
132	2	1	2	2	3	4	4	0	2	3	6	4	4	4	5
204	4	3	4	1	1	2	4	2	0	1	4	2	2	3	4
205	5	4	5	2	2	1	3	3	1	0	3	1	3	2	3
219	6	6	8	6	5	4	4	6	4	3	0	2	2	1	2
221	6	5	6	4	3	2	2	4	2	1	2	0	2	1	2
222	6	5	6	4	3	4	4	4	2	3	2	2	0	1	2
223	7	6	7	5	4	3	3	4	3	2	1	1	1	0	1
255	8	7	6	6	5	4	4	5	4	3	2	2	2	1	0

Table 5: Hamming distance separation between all attractor pairs

Attractor	Mean separation from other attractors	Minimum separation	Maximum separation
0	4	1	8
4	3.2	1	7
36	4	1	8
68	2.67	1	6
76	2.6	1	5
77	2.67	1	4
85	3.067	2	4
132	3.067	1	6
204	2.467	1	4
205	2.53	1	5
219	3.93	1	8
221	2.8	1	6
222	3.2	1	6
223	3.2	1	7
255	3.93	1	8

Table 6: Statistical data regarding the attractor separation

The basin of attraction field discussed in section is considered a second order of complexity in the system. The Hamming distance between every pair of attractors was calculated and presented below.

The mean separation from all other attractors for any given attractor can provide an indication of how 'scattered' the basin of attraction field is. The maximum possible separation was known a priori, insofar that both 0 and 255 were obvious attractors. The largest mean separation is 4, for attractors 0 and 36. While most attractors have an immediate neighbouring attractor (separated by Hamming distance of 1), the minimum Hamming distance separation of attractor 85 is 2. Given this and that the largest mean separation is half the maximum possible Hamming distance, it can be said that the attractors are, to an extent, scattered.

2.3.3 Basins of Attraction

A sample of the attractor basins were chosen for comparison. The chosen attractors were chosen based on their apparent ordered behaviours and are separated by a Hamming distance of 1.

ID of attractor	Max Path Length	Min Path Length	Mean Path Length	Near predecessors	Total nodes around attractor	Total nodes (excluding GoE)	GoE nodes	G-density	ρ	Symmetry?
221	3	1	1.67	3	5	4	3	0.6	0.75	no
223	1	1	1	2	2	0	2	1	0.875	no

Table 7: Properties of nearby attractors 221 and 223

The majority of the properties of attractors 223 and 221 are below the global mean for each property. However, the G-density of 223 is the upper bound of 1. The 2 could perhaps be considered a cluster, owing to the short distance between the attractors in the basin of attraction field. The pair can both be considered ordered since they have short max path lengths and above average G-densities. Both have a small degree of entropy, a trait common with highly ordered ECA rules such as rule 0 and rule 255

ID of attractor	Max Path Length	Min Path Length	Mean Path Length	Near predecessors	Total nodes around attractor	Total nodes (excluding GoE)	GoE nodes	G-density	ρ	Symmetry?
222	1	1	1	1	1	0	1	1	0.75	no
223	1	1	1	2	2	0	2	1	0.875	no

Table 8: Properties of nearby attractors 222 and 223

The attractor 222 has only 1 near predecessor, which is also the GoE state within the basin of attraction. Again, a significant degree of order is displayed by these attractors wen compared to the system wide mean values. Interestingly, the trajectories leading to these 2 nodes are within short Hamming distance of each other.

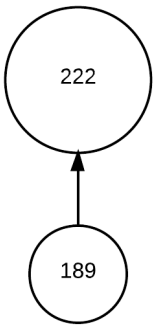

Figure 21: Attractor basin of state 222

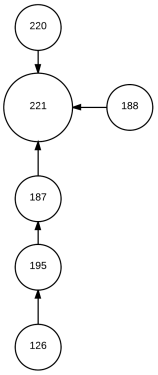

Figure 22: Attractor basin of state 221

A conjecture is that a certain common property may exists within these trajectories, that may be visible upon inspection of the time-space diagram of each evolution. However, it can be suggested that the attractors 221, 222 and 223 are part of an ordered cluster within the basin of attraction field.

Attractors 68 and 76 were also compared. Attractor 76 was found to be more entropic than 68, and more so than the system wide average of 16.067. 76 has a high G-density, an a relatively short path length, which indicates ordered trajectories. However, the multiple branching transients demonstrates more complex behaviour in the basin of attraction of 76 compared to 68. These attractors appear to exhibit higher complexity than 221, 222 and 223, as there are fewer GoE states that decay immediately to the attractor cycle.

ID of attractor	Max Path Length	Min Path Length	Mean Path Length	Near predecessors	Total nodes around attractor	Total nodes (excluding GoE)	GoE nodes	G-density	ρ	Symmetry?
68	4	1	2.5	2	5	3	2	0.4	0.25	no
76	5	1	3.17	2	17	5	12	0.706	0.375	no

Table 9: Properties of nearby attractors 68 and 76

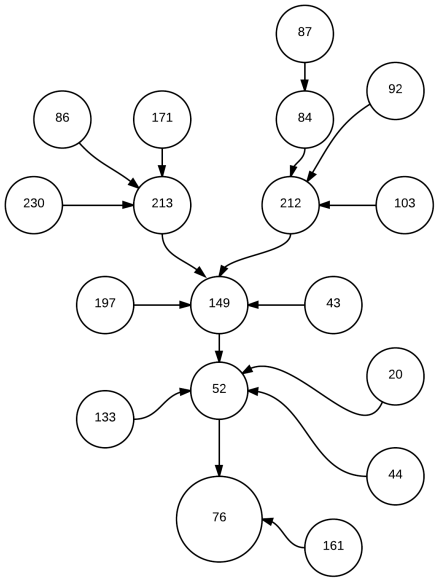

Figure 23: Attractor basin of state 76

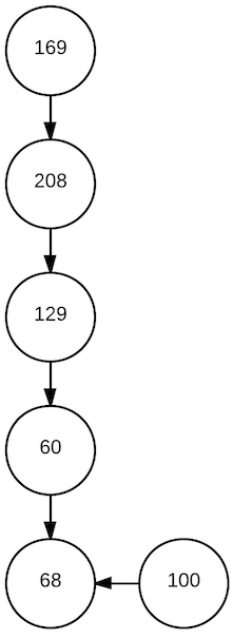

Figure 24: Attractor basin of state 68

ID of attractor	Max Path Length	Min Path Length	Mean Path Length	Near predecessors	Total nodes around attractor	Total nodes (excluding GoE)	GoE nodes	G-density	ρ	Symmetry?
76	5	1	3.17	2	17	5	12	0.706	0.375	no
77	3	1	1.5	4	10	6	4	0.4	0.5	no

Table 10: Properties of nearby attractors 76 and 77

Comparing the properties of 77 and 76, the entropy difference is less significant. The difference in G-density is similar to the comparison between 76 and 68. The difference in ρ is also similar. Despite the smaller difference in entropy, 76 still seems to exhibit more complex behaviour than 68 or 77, since 76 has a significantly larger number of sub tree branches.

These comparisons begins to identify the range of behaviours present in the attractors of the system. On inspection it can be seen that no pair of attractor basins share identical structures. Given that one of the few predictions that could be made of the system was that both 0 and 255 would be attractors, it is

2.3.4 Divergent Trajectories

Chaotic systems exhibit an exponential rate of divergence between 2 initially nearby trajectories. This property can be measured by the Maximum Lyapunov exponent (MLE). A system is considered chaotic if the MLE is positive. [14] Given the restricted universe size, the maximum Hamming distance is 8 and therefore the upper limit on the MLE is $\log_2 8 = 3$. Note that a base 2 logarithm has been chosen, owing the binary nature of the system.

Given the discrete space of this system, d_0 has been set to 1. Furthermore, the calculation takes the mean of the immediate Lyapunov exponents up to the point where the both trajectories are at the attractor.

$$\lambda =< \log_2 d > \tag{6}$$

A sample of 7 pairs of trajectories has been provided, though there is good reason to calculate each nearby pair in order to find the largest MLE in the system. However, this provides a first test to quantify the chaos present in the system.

t	Trajectory 1	Trajectory 2	Hamming distance (d)	$\log_2 d$
1	24	25	1	0
2	40	106	2	1
3	16	117	4	2
4	32	207	7	2.8
5	0	239	7	2.8
6	0	255	8	3

Table 11: A comparison of the trajectories starting at states 24 and 25

The table displays the instantaneous Lyapunov exponents along the given trajectories at different time steps. Below, the MLE for each of the compared trajectories is presented.

Trajectories	15 & 16	18 & 17	24 & 25	26 & 27	31 & 32	34 & 33	90 & 91
MLE	1.30	1.73	2.10	2.05	1.79	1.3	1.57

Table 12: MLE for the compared trajectories

The MLE are all positive, though this was expected since the compared trajectories were not bound to the same attractor. However, this still indicates chaotic behaviour within the system. A full comparison of all trajectories separated by $d_0 = 1$ would provide a complete overview. Furthermore, a RCA system with neighbourhood $n = 5$ could possibly contain strange attractors, so that nearby trajectories within the same basin of attraction could be compared for exponential divergence.

2.3.5 Sampling the $5 = n$ RCA Rulespace

The $5 = n$ RCA was briefly sampled. Aside from the obvious attractors, other attractors were found, however, it is not the case that attractors found in the $n = 3$ are preserved. Furthermore, attractors exhibiting type 2 behaviour were found.

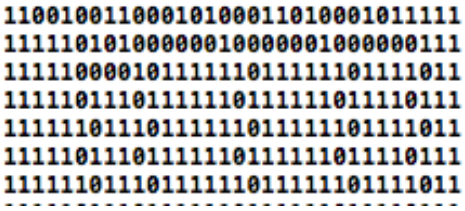

Figure 25: Space-time trajectory of initial state leading to a periodic attractor in the $n = 5$ RCA. Updates on the attractor cycle shift the ensemble by 1 space

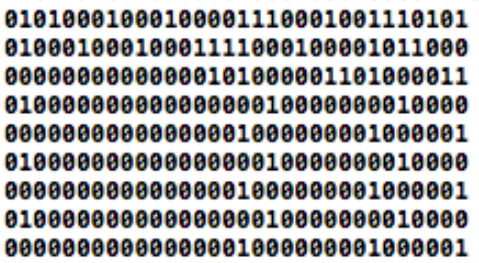

Figure 26: Another example, note updates on the attractor cycle shift the ensemble along by 2 cells

Not only are attractors of this class unattainable in the $n = 3$ RCA, but the transition functions between the two points on the attractor preserve the configuration, merely causing it to shift along. More examples of this behaviour were found, but in the brief sample that was taken, there was no evidence of strange attractors. The sample, however, indicates that the size of the system may limit the different types of behaviours exhibited by the attractors and that a systematic search of the $n = 5$ system could reveal more complicated attractors.

2.4 Discussion

The system exhibits a wide range of behaviours, which in itself may be indicative of disorder. The attractors in this system display a large range of behaviour and even though the attractors are all point attractors, there are varying lengths of transient throughout. It has been demonstrated that the smallest possible perturbations in initially nearby states can lead to largely diverging trajectories, indicated by a positive MLE. In some of the examples we have seen that one trajectory may stabilise on an attractor more quickly than the other. This is indicative of disorder.

There are different algorithms that can be employed to calculate the MLE of a system, one alternative is the non-directional MLE. This is the logarithm of the time averaged number of cells in a cells neighbourhood that change state at each iteration if the state

is perturbed. The neighbourhood size gives this value an upper bound of MLE $=\log n$. If the system evolves to a limit cycle then the MLE $= -\infty$. [14] However, the method employed in this paper still indicates the divergence of trajectories.

The two most intricate attractors within the system are 0 and 255. Both possess the highest number of states that lead into them and the longest transient trees, with the highest degree of branching. The basins of 0 and 255 are highly distinct, qualitatively, from their ECA counterparts. The ECA and RCA attractors 0 and 255 are common in that application of the rule yields a universe that is identical to the rule. A comparison may begin with the entropy of the attractors. ECA 0 and 255 are low entropy whereas for RCA they are comparatively high. This indicates a large difference in the behaviour of the ECA and RCA basin fields. This is evidenced further by comparing the state transition diagrams, in which the RCA attractors demonstrate significantly long transients with high degrees of branching and also highly ordered trajectories.

Comparing RCA 0 and 255 directly reveals that the properties are not identical, despite the fact that 0 and 255 are related by a state inversion operation. This indicates that there may be a favouring towards certain attractors in this system despite apparent symmetries between the attractors universe structure. Comparison of the other two symmetric attractors, 36 and 219, revealed similar trends, though the G-density was larger in the lower ρ attractor, 36. This indicates further that limitations on this system similar to those that Wolfram applied to ECA were not relevant. Additionally, this indicates that there are less obvious symmetries in this system than ECA.

However, it was found that between the groups $\rho = 0.25$ and $\rho = 0.75$ there existed simple relations between two of the pairs. Alongside the fact that these two groupings had the same number of members, there is the possibility of some order. Speculatively, there is a symmetry in the dynamics that is not immediately obvious that gives rise to this behaviour.

Comparing attractors, separated by short Hamming distances, revealed a highly ordered cluster in 221, 222 and 223. This cluster was particularly intriguing as the cluster 222 and 223 had the smallest entropy, and the near predecessor states were separated by small Hamming distances. The GoE states were similarly local to the attractors. Speculatively, this could indicate a highly ordered region within the basin field.

Longer path lengths were found in higher entropy attractors. It is possible that there exists a statistical relation, since more states would lead to a higher chance of long transient, but it is also further evidence of disorder in the system. It is speculated that if the system were completely ordered, all attractors would share a basic structure in which all leading states decay immediately to the attractor state (analogous to highly ordered space-time diagrams of ECA rules 0 and 255). However, this is evidently not the case.

2.5 Conclusion and Future Research

This first investigation into the behaviour of RCA has provided the basis upon which further, more directed research may be conducted. We have defined a new, recursive CA architecture and provided early comparisons between the behaviour of RCA and ECA.

On balance, it is clear that the $n = 3$ RCA demonstrates some disordered behaviour, but it is evident that there are fairly ordered transitions also. The macroscopic approach

provided by studying the state transition diagrams has indicated basins of attraction with significantly varying structure, indicating a significant degree of variation in the dynamics of RCA.

The RCA primarily studied here is the smallest possible RCA. A more extensive study into the $n = 5$ RCA would provide a larger data set and a further point of comparison. The brief sample of the $n = 5$ RCA indicated the existence of limit cycle attractors with a period of 2. This provides evidence that the degree of complexity in RCA systems increases with the size of the system.

Overall, the $n = 3$ RCA has demonstrated disordered behaviour. Increasing the size of the system to $n = 5$ demonstrates the emergence of slightly more complicated attractors, given a systematic study of all 2^{5^5} configurations, a full comparison between the two systems may be made. This will provide the basis of future research.

A Data, Basins of Attraction and Source Code

A.1 Tables of Compared Trajectories

t	attractor	attractor	hamming distance (d)	$\log_2 d$
1	18	17	1	0
2	45	102	4	2
3	55	85	3	1.59
4	200	85	5	2.32
5	192	85	3	1.59
6	128	85	5	2.32
7	0	85	4	2
8	0	85	4	2

Table 13: A comparison of the trajectories starting at states 18 and 17

t	attractor	attractor	hamming distance (d)	$\log_2 d$
1	26	27	1	0
2	41	233	2	1
3	16	241	4	2
4	32	231	5	2.32
5	0	215	6	2.59
6	0	151	5	2.32
7	0	115	5	2.32
8	0	206	5	2.32
9	0	238	6	2.59
10	0	255	8	3

Table 14: A comparison of the trajectories starting at states 18 and 17

t	attractor	attractor	hamming distance (d)	$\log_2 d$
1	15	16	1	0
2	225	32	3	1.58
3	205	0	5	2.32

Table 15: A comparison of the trajectories starting at states 15 and 16

t	attractor	attractor	hamming distance (d)	$\log_2 d$
1	34	33	1	0
2	17	12	4	2
3	102	4	3	1.59
4	85	4	3	1.59

Table 16: A comparison of the trajectories starting at states 34 and 33

t	attractor	attractor	hamming distance (d)	$\log_2 d$
1	90	91	1	0
2	153	27	2	1
3	170	233	3	1.59
4	85	241	3	1.59
5	85	231	4	2
6	85	215	2	1
7	85	151	3	1.59
8	85	115	3	1.59
9	85	206	5	2.32
10	85	238	6	2.58
11	85	255	4	2

Table 17: A comparison of the trajectories starting at states 90 and 91

t	attractor	attractor	hamming distance (d)	$\log_2 d$
1	31	32	1	0
2	205	35	3	1
3	205	152	4	2
4	205	41	4	2
5	205	16	6	2.59
6	205	32	6	2.59
7	205	0	5	2.32

Table 18: A comparison of the trajectories starting at states 31 and 32

A.2 State transition graphs

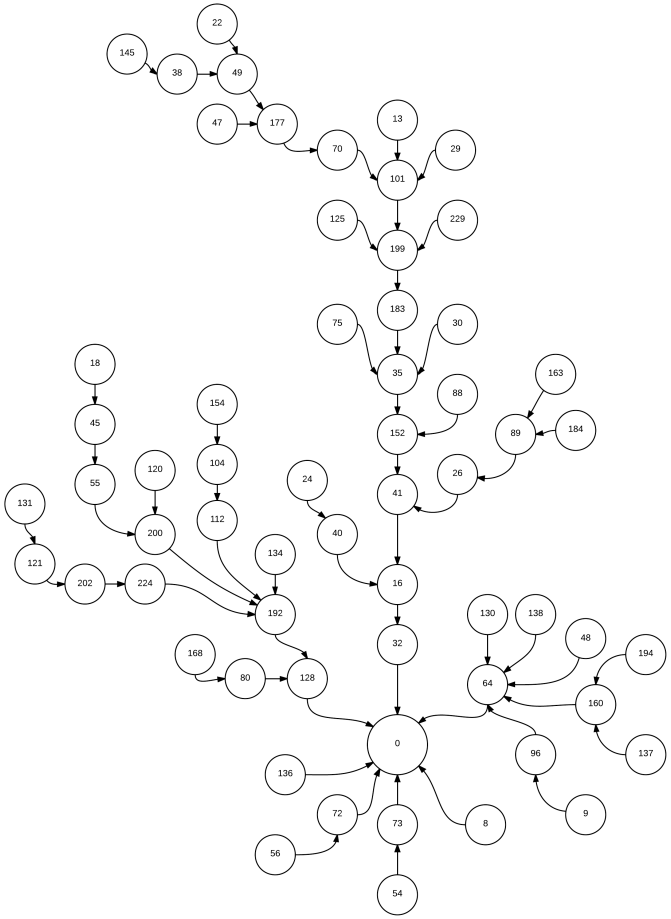

Figure 27: Basin of attraction of 0

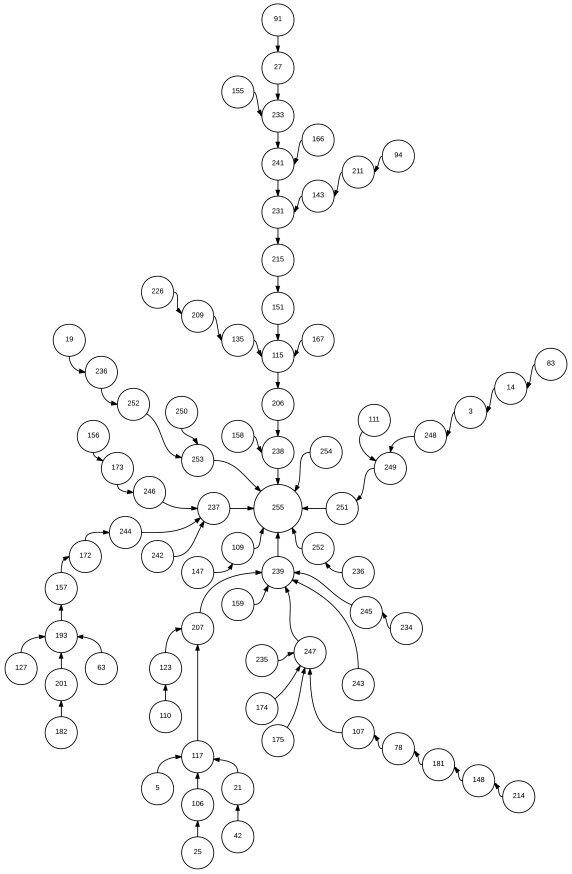

Figure 28: Basin of attraction of 255

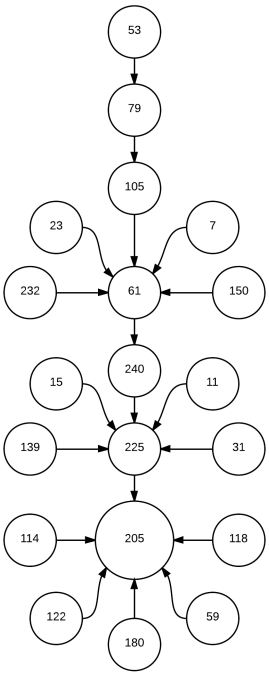

Figure 29: Basin of attraction of 205

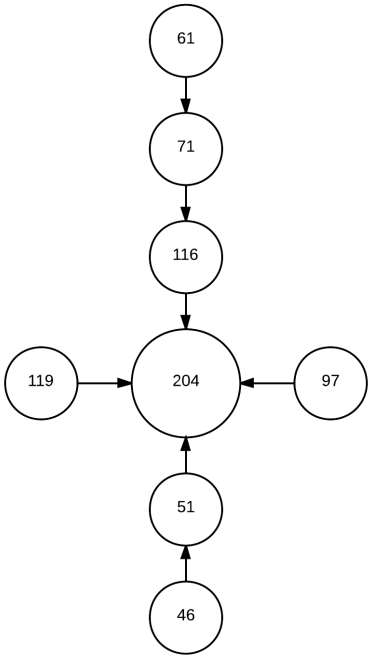

Figure 30: Basin of attraction of 204

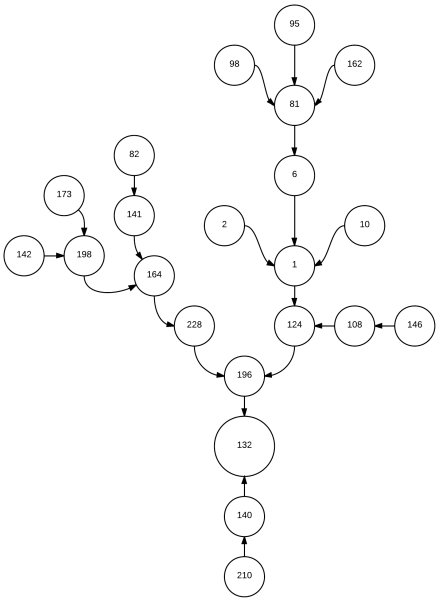

Figure 31: Basin of attraction of 132

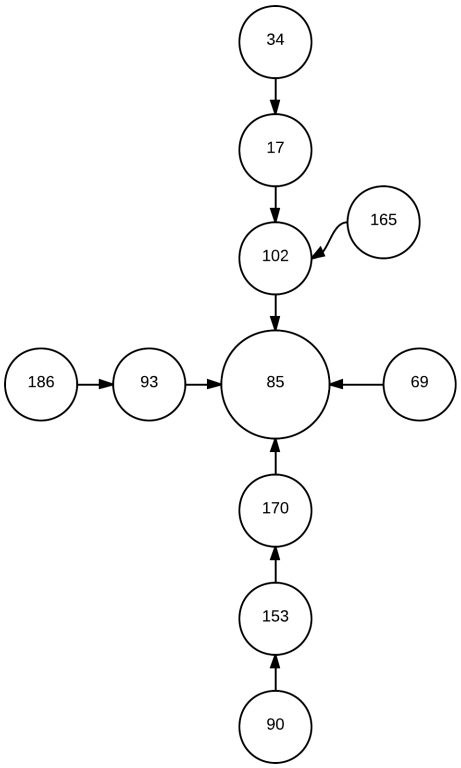

Figure 32: Basin of attraction of 85

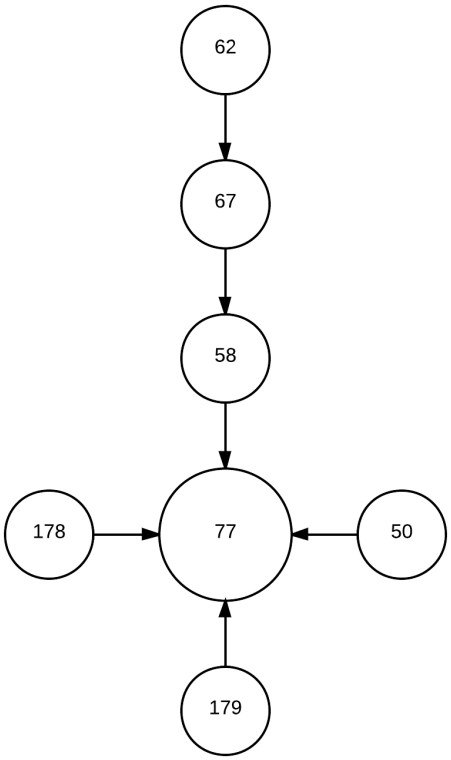

Figure 33: Basin of attraction of 77

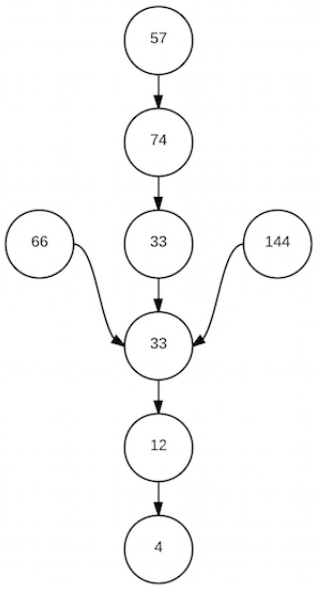

Figure 34: Basin of attraction of 132

A.3 Source Code

main.cpp 11/04/2014 01:51

```
//  main.cpp
//  CA 3 neighbourhood including recursive rule
//  Copyright (c) 2013 Joshua Underwood. All rights reserved.
//
#include <iostream>
#include <cstdlib>
#define _USE_MATH_DEFINES
#include <math.h>
#include <cfloat>
#include <climits>
#include <iomanip>
#include <fstream>
using namespace std;
//function that generates the next state for each cell, call it update!
int updater(int a, int b, int c, int n);
int statenumber(int a[]);
void generator();
int main()
{
    int width = 8;
    int neighbourhoodwidth = 3;
    cout << "3 NEIGHBOURHOOD SELF REFLEXIVE CELLULAR AUTOMATA"<<endl;
    //allocating 1-d array for cells and rule, dynamic to allow user
        determined universe size
    int * cellspace; //cells
    int * rulespace; //rules
    int * updatedcellspace; //updated cells
    cellspace = new int[width];
    rulespace = new int[width];
    updatedcellspace = new int[width];
    //INPUT OF RULE
    cout << "The rule, provide binary digit for each element:" << endl;
    for (int i = 0; i < width; i++){
        cout << "Rule element " << i + 1 << " : ";
        cin >> rulespace[i];
        //this while statement is in place to prevent the user putting in a
            non-binary digit
        while (rulespace[i] > 1){
            cin >> rulespace[i];
        }
        cout << endl;
    }
    //INPUT OF INITIAL STATE - Initial state becomes rulespace
    for (int i = 0; i < width; i++){
        cellspace[i] = rulespace[i];
        }

    //OUTPUT OF RULE AND INIT STATE
    cout << "The rule is: ";
    for (int i = 0; i < 8; i++){
        cout << rulespace[i];
    }
    cout << endl;
    //cout << "The initial state is: ";
    for (int i = 0; i < width; i++){
        cout << cellspace[i];
    }
    cout << " : " << statenumber(cellspace) << endl;
    ////////////////////////////////////////
    //set how many iterations to run through//
```

```
        ////////////////////////////////////////
        for (int count = 0; count < 256; count++){
            /////////////////
            //UPDATE THE CA//
            /////////////////
            for (int i = 0; i < width; i++) {
                ////////////////////////////////////////////////////////////////////
                    ///
                //to give CA circular topology will need to use specific
                    conditions//
                ////////////////////////////////////////////////////////////////////
                    ///
                if (i == 0){
                    int q = cellspace[width - 1];
                    int w = cellspace[i];
                    int e = cellspace[i + 1];
                    int upsta = updater(q, w, e, neighbourhoodwidth);
                    updatedcellspace[i] = rulespace[upsta];
                    i++;
                }
                if (i == width-1){
                    int q = cellspace[i - 1];
                    int w = cellspace[i];
                    int e = cellspace[0];
                    int upsta = updater(q, w, e, neighbourhoodwidth);
                    updatedcellspace[i] = rulespace[upsta];
                    i++;
                }
                else {
                    int q = cellspace[i-1];
                    int w = cellspace[i];
                    int e = cellspace[i+1];
                    int upsta = updater(q, w, e, neighbourhoodwidth);
                    updatedcellspace[i] = rulespace[upsta];
                }
            }
            //setting c and rulespace to the new values
            for (int i = 0; i < width; i++){
                cellspace[i] = updatedcellspace[i];
                rulespace[i] = updatedcellspace[i];
            }
            //now print the current state to see what's going on
            for (int i = 0; i < width; i++){
                cout << cellspace[i];
            }
            cout << " : " << statenumber(cellspace) << endl;
        }
        delete[] cellspace, rulespace, updatedcellspace;
        return 0;
    }
    int updater(int a, int b, int c, int n){
        //currently this only works for neighbourhood of width = 3
        int ruleelement;
        //need to make this loop and work for any size neighbourhood MAKE IT ADD
            THE STATE FROM THE CELL ARRAY TO CORRESPONDING NEIGHBOURHOOD ARRAY
            ELEMENT
        //  updaterarray = new int[n];
        //now to add elements
        a = a * 4;
        b = b * 2;
```

```
        ruleelement = a + b + c;
        //cout << ruleelement << endl;
        return ruleelement;
    }
     int statenumber (int a[]){
     int sn=0;
     //int[a] takes cellspace as argument and n is width
     //sum up the array components by iterative sum in order to have arbitrary
         neighbourhood
     for (int b = 0; b<8; b++){
     sn += a[b]*pow(2,b);
     }
     return sn;
     }
```

```
//  main.cpp
//  CA 5 neighbourhood including recursive rule
//  Copyright (c) 2013 Joshua Underwood. All rights reserved.
//
#include <iostream>
#include <cstdlib>
#define _USE_MATH_DEFINES
#include <math.h>
#include <cfloat>
#include <climits>
#include <iomanip>
#include <fstream>
using namespace std;
//function that generates the next state for each cell, call it update!
int updater(int a, int b, int c, int d, int e, int n);
int coin();
unsigned long long statenumber(int a[]);
int main()
{
    int width = 32;
    int neighbourhoodwidth = 5;
    cout << "5 NEIGHBOURHOOD SELF REFLEXIVE CELLULAR AUTOMATA"<<endl;
    //allocating 1-d array for cells and rule, dynamic to allow user
        determined universe size
    int * cellspace; //cells
    int * rulespace; //rules
    int * updatedcellspace; //updated cells
    int seed = 1;
    cellspace = new int[width];
    rulespace = new int[width];
    updatedcellspace = new int[width];
    srand(static_cast<int>(time(0)));
    //INPUT OF RULE
    cout << "The rule, provide binary digit for each element:" << endl;
    for (int i = 0; i < width; i++){
        cout << "Rule element " << i + 1 << " : ";
        rulespace[i] = coin();
        //this while statement is in place to prevent the user putting in a
            non-binary digit
        while (rulespace[i] > 1){
            cin >> rulespace[i];
        }
        cout << endl;
    }
    //INPUT OF INITIAL STATE - Initial state becomes rulespace
    for (int i = 0; i < width; i++){
        cellspace[i] = rulespace[i];
    }
    //OUTPUT OF RULE AND INIT STATE
    cout << "The rule is: ";
    for (int i = 0; i < width; i++){
        cout << rulespace[i];
    }
    cout << endl;
    //cout << "The initial state is: ";
    for (int i = 0; i < width; i++){
        cout << cellspace[i];
    }
    cout << " : " << statenumber(cellspace) << endl;
    ////////////////////////////////////////
```

```
        //set how many iterations to run through//
        ////////////////////////////////////////
        for (int count = 0; count < 256; count++){
            /////////////////
            //UPDATE THE CA//
            /////////////////
            for (int i = 0; i < width; i++) {
                ////////////////////////////////////////////////////////////////////
                    ///
                //to give CA circular topology will need to use specific
                    conditions//
                ////////////////////////////////////////////////////////////////////
                    ///
                if (i == 0){
                    int q = cellspace[width - 2];
                    int w = cellspace[width - 1];
                    int e = cellspace[i];
                    int r = cellspace[i+1];
                    int t = cellspace[i+2];
                    int upsta = updater(q, w, e, r, t, neighbourhoodwidth);
                    updatedcellspace[i] = rulespace[upsta];
                    i++;
                }
                if (i==1){
                    int q = cellspace[width - 1];
                    int w = cellspace[i];
                    int e = cellspace[i+1];
                    int r = cellspace[i+2];
                    int t = cellspace[i+3];
                    int upsta = updater(q, w, e, r, t, neighbourhoodwidth);
                    updatedcellspace[i] = rulespace[upsta];
                    i++;
                }
                if (i == width-2){
                    int q = cellspace[28];
                    int w = cellspace[29];
                    int e = cellspace[30];
                    int r = cellspace[31];
                    int t = cellspace[0];
                    int upsta = updater(q, w, e, r, t, neighbourhoodwidth);
                    updatedcellspace[i] = rulespace[upsta];
                    i++;
                }
                if (i == width-1){
                    int q = cellspace[29];
                    int w = cellspace[30];
                    int e = cellspace[31];
                    int r = cellspace[0];
                    int t = cellspace[1];
                    int upsta = updater(q, w, e, r, t, neighbourhoodwidth);
                    updatedcellspace[i] = rulespace[upsta];
                    i++;
                }
                else {
                    int q = cellspace[i-2];
                    int w = cellspace[i-1];
                    int e = cellspace[i];
                    int r = cellspace[i+1];
                    int t = cellspace[i+2];
                    int upsta = updater(q, w, e, r, t, neighbourhoodwidth);
```

```
                    updatedcellspace[i] = rulespace[upsta];
                }
            }
            //setting c and rulespace to the new values
            for (int i = 0; i < width; i++){
                cellspace[i] = updatedcellspace[i];
                rulespace[i] = updatedcellspace[i];
            }
            //now print the current state to see what's going on
            for (int i = 0; i < width; i++){
                cout << cellspace[i];
            }
            cout << " : " << statenumber(cellspace) << endl;
        }
        delete[] cellspace, rulespace, updatedcellspace;
        return 0;
    }
    int updater(int a, int b, int c, int d, int e, int n){
        //currently this only works for neighbourhood of width = 3
        int ruleelement;
        //need to make this loop and work for any size neighbourhood MAKE IT ADD
            THE STATE FROM THE CELL ARRAY TO CORRESPONDING NEIGHBOURHOOD ARRAY
            ELEMENT
        //  updaterarray = new int[n];
        //now to add elements
        a = a * 16;
        b = b * 8;
        c = c * 4;
        d = d * 2;
        ruleelement = a + b + c + d + e;
        //cout << ruleelement << endl;
        return ruleelement;
    }
    unsigned long long statenumber (int a[]){
        int sn=0;
        //int[a] takes cellspace as argument and n is width
        //sum up the array components by iterative sum in order to have arbitrary
            neighbourhood
        for (int b = 0; b<32; b++){
            sn += a[b]*pow(2,b);
        }
        return sn;
    }
    int coin()
    {
        int res;
        random();
        res= (rand() % 2);
        return res; // will give either 0 or 1
    }
```

B References

[1] Wolframscience.com. Historical Notes: Three-body problem. [Online] Available from:http://www.wolframscience.com/reference/notes/972d [Accessed 10 Apr 2014].

[2] Wolfram S. Theory and applications of cellular automata. Singapore: World Scientific; 1986. pp. v

[3] Wolfram S. Theory and applications of cellular automata. Singapore: World Scientific; 1986. pp. 1-4

[4] Wolfram S. Statistical mechanics of cellular automata. Reviews of modern physics. 1983; 55 (3): pp.601.

[5] Clewlow L. (1989). Cellular Automata and Dynamical Systems. Ph.D Thesis. University of Warwick: UK. pp. 12

[6] Wuensche A, Lesser M. The global dynamics of cellular automata. Reading, Mass.: Addison-Wesley; 1992. pp. 5 - 16

[7] Packard N, Wolfram S. Two-dimensional cellular automata. Journal of Statistical Physics. 1985; 38 (5-6): pp. 901–946.

[8] Wuensche A. The Ghost in the Machine. 1994; 17: 465–465.

[9] Uncomp.uwe.ac.uk. State-space and basins of attraction. [Online] Available from: http://uncomp.uwe.ac.uk/wuensche/boa_idea.html [Accessed 10 Apr 2014].

[10] Wuensche A, Lesser M. The global dynamics of cellular automata. Reading, Mass.: Addison-Wesley; 1992. pp. 49

[11] Wuensche A. Discrete dynamical networks and their attractor basins. LIFE Science Publications, Charles Stuart University. 1998.

[12] Apolloni N, Others. Analysis of attractor distances in random Boolean networks. 2011; 226: 201.

[13] Wolfram Demonstrations Project. Wolfram Demonstrations Project. [Online] Available from: wolfram.com/CellularAutomatonStateTransitionDiagrams/ [Accessed 10 Apr 2014].

[14] "Lyapunov Exponents of Elementary Cellular Automata" from the Wolfram Demonstrations Project. Wolfram Demonstrations Project. [Online] Available from: http://demonstrations.wolfram.com/LyapunovExponentsOfElementaryCellularAutomata/ [Accessed 10 Apr 2014].

[15] Theory.org. Theory of Cellular Automata. [Online] Available from:https://theory.org/complexity/cdpt/html/node4.html [Accessed 10 Apr 2014].

[16] "Cellular Automata Ordered by Entropy" from the Wolfram Demonstrations Project. Wolfram Demonstrations Project. [Online] Available from:http://demonstrations.wolfram.com/CellularAutomataOrderedByEntropy/ [Accessed 10 Apr 2014].

[17] Images.scholarpedia.org. Untitled. [Online] Available from: http://images.scholarpedia.org/w/images/5/56/Hopfieldattractor.jpg [Accessed 10 Apr 2014].

[18] Langton C. Computation at the edge of chaos. 1991.