Basins of Attraction of Cellular Automata and Discrete Dynamical Networks

Andrew Wuensche* *Discrete Dynamics Lab.*

July 21, 2017

Glossary

State-space

The set of unique states in a finite and discrete system. For a system of size n, and value-range v, the size of state-space $S{=}n^v$.

Cellular automata, CA

Although a CA is often treated as having infinite size, we are dealing here with finite CA, which usually consists of "cells" arranged in a regular lattice (1d, 2d, 3d) with periodic boundary conditions, making a ring in 1D and a torus in 2D ("null" and other boundary conditions may also apply). Each cell updates its value (usually in parallel, synchronously) as a function of the values of its close local neighbors. Updating across the lattice occurs in discrete time-steps. CA have one homogeneous function, the "rule", applied to a homogeneous neighborhood template. However, many of these constraints can be relaxed.

Random Boolean networks, RBN

Relaxing CA constraints, where each cell can have a different, random (possibly biased) non-local neighborhood, or put another way, random wiring of k inputs (but possibly with heterogenious-k) and heterogenious rules (a rule-mix) but possibly just one rule, or a bias of rule types.

Discrete dynamical networks

Relaxing RBN constraints by allowing a value-range that is greater than binary, $v \geq 2$, heterogenious-k, and a rule-mix.

*andy@ddlab.org, http://www.ddlab.org

Space-time pattern

A time-sequence of states from an initial state driven by the dynamics, making a trajectory. For 1D systems this is usually represented as a succession of horisontal value strings from the top down, or scrolling down the srceen.

Random maps, MAP

Directed graphs with out-degree one, where each state in state-space is *assigned* a successor, possibly at random, or with some bias, or according to a dynamical system. CA, RBN and DDN, which are usually sparsely connected ($k \ll n$), are all special cases of random maps. Random maps make a basin of attraction field, by definition.

Pre-images

A state's immediate predecessors

Garden-of-Eden state

A state having no pre-images, also called a leaf state.

Attractor, basin of attraction, subtree

The terms "attractor" and "basin of attraction" are borrowed from continuous dynamical systems. In this context the attractor signifies the repetitive cycle of states into which the system will settle. The basin of attraction in convergent (injective) dynamics includes the transient states that flow to an attractor as well as the attractor itself, where each state has one successor but possibly zero or more predesessors (pre-images). Convergent dynamics implies a topology of trees rooted on the attractor cycle, though the cycle can have a period of just one, a point attractor. Part of a tree is a sub-tree defined by its root and number of levels. These mathematical object may be refered to in general as "attracor basins".

Basin of attraction field

One or more basins of attraction comprising all of state-space.

State transition graph

A graph representing attracor basins consisting of directed arcs linking nodes, representing single time-steps linking states, with a topology of trees rooted on attractor cycles, where the direction of time is inward from garden-of-Eden states towards the attractor. Various graphical conventions determine the presentation. The terms "state transition graph" and various types of "attractor basins" may be used interchangeably.

Reverse algorithms

Computer algorithms for generating the pre-images of a network state. The information is applied to generate state transition graphs (attractor basins) according to a graphical convention. The software DDLab, applied here, utilises three different reverse algorithms. The first two generate pre-images directly so are more efficient than the exhaustive method, allowing greater system size.

- An algorithm for local 1d wiring [15] — 1d CA but rules can be heterogeneous.
- A general algorithm [16] for RBN, DDN, 2d or 3d CA, which also works for the above.
- An exhaustive algorithm that works for any of the above by creating a list of "exhaustive pairs" from forward dynamics. Alternatively, a random list of exhaustive pairs can be created to implement attractor basin of a "random map".

Definition of the subject

Basins of attraction of cellular automata and discrete dynamical networks link state-space according to deterministic transitions, giving a topology of trees rooted on attractor cycles. Applying reverse algorithms, basins of attraction can be computed and drawn automatically. They provide insights and applications beyond single trajectories, including notions of order, complexity, chaos, self-organisation, mutation, the genotype-phenotype, encryption, content addressable memory, learning, and gene regulation. Attractor basins are interesting as mathematical objects in their own right.

Introduction

"The Global Dynamics of Cellular Automata"[15] published in 1992 introduced a reverse algorithm for computing the pre-images (predecessors) of states for finite 1D binary cellular automata (CA) with periodic boundaries. This made it possible to reveal the precise graph of "basins of attraction" — state transition graphs — states linked into trees rooted on attractor cycles, which could be computed and drawn automatically as in figure 1. The book included an atlas for two entire categories of CA rule-space, the 3-neighbor "elementary" rules, and the 5-neighbor totalistic rules.

In 1993, a different reverse algorithm was invented[17] for the pre-images and basins of attraction of random Boolean networks (RBN) (figure 15) just in time to make the cover of Kauffman's seminal book[10] "The Origins of Order" (figure 3). The RBN algorithm was later generalised for "discrete dynamical networks" (DDN) described in "Exploring Discrete Dynamics"[26]. The algorithms, implemented in the software DDLab[27], compute pre-images directly,

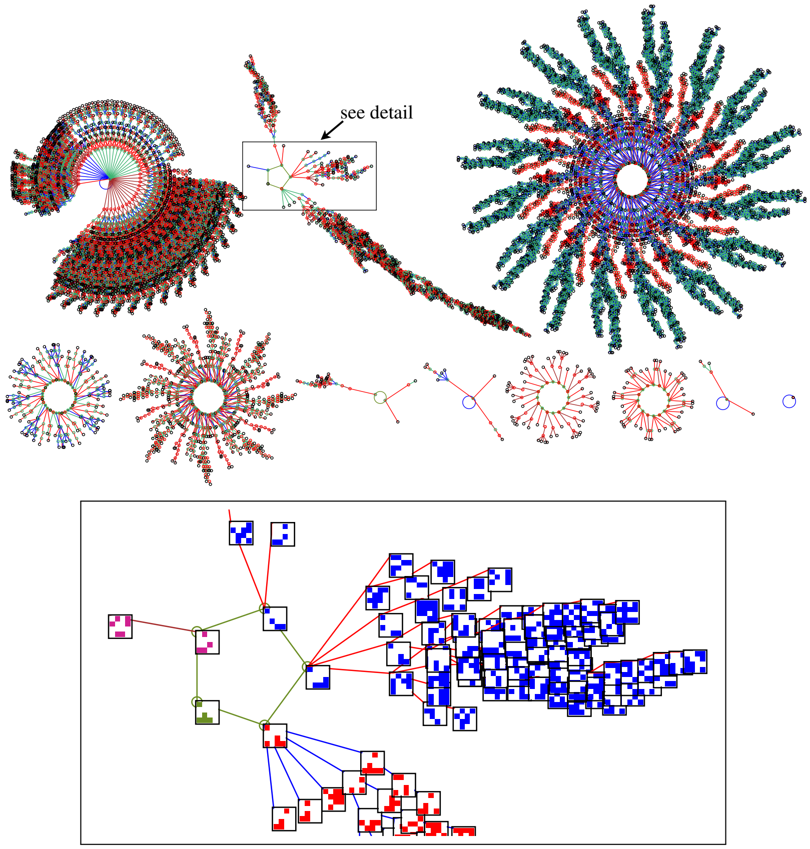


Figure 1: *Top:* The basin of attraction field of a 1D binary CA, k=7, n=16[21]. The 2^{16} states in state-space are connected into 89 basins of attraction, only the 11 nonequivalent basins are shown, with symmetries characteristic of CA[15]. Time flows inwards, then clockwise at the attractor. *Below*: A detail of the second basin, where states are shown as 4×4 bit patterns.

Figure 2: Space-time pattern for the same CA as in figure 1 but for a much larger system (n=700). About 200 time-steps from a random initial state. Space is across and time is down. Cells are colored according to neighborhood look-up instead of the value.

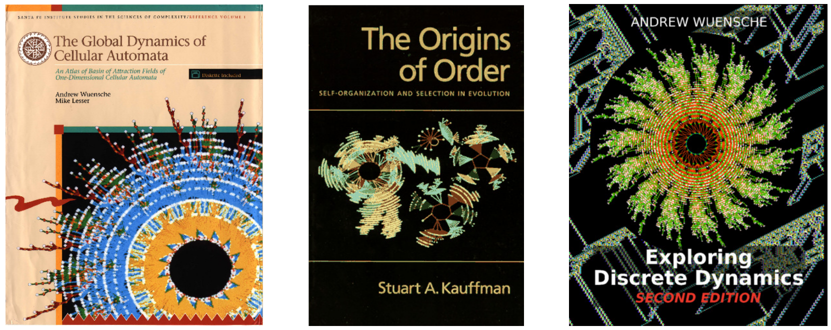


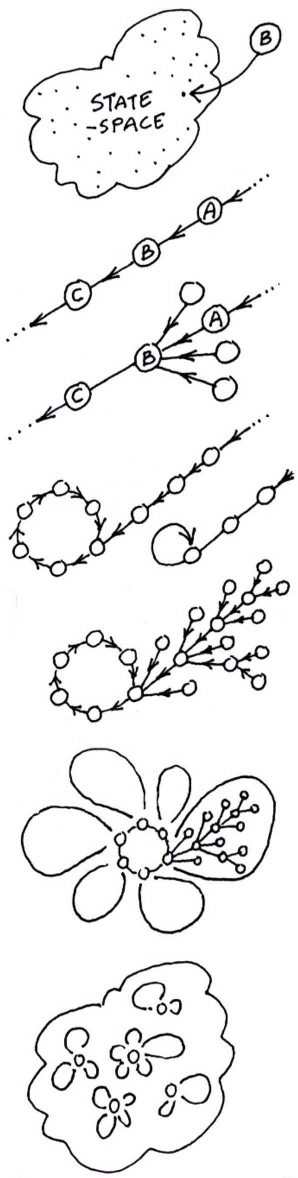
Figure 3: The front covers of Wuensche and Lesser's (1992) "The Global Dynamics of Cellular Automata" [15], Kauffman's (1993) "The Origins of Order" [10], and Wuensche's(2016) "Exploring Discrete Dynamics" 2nd Ed [26].

and basins of attraction are drawn automatically following flexible graphic conventions. There is also an exhaustive "random map" algorithm limited to small systems, and a statistical method for dealing with large systems. A more general algorithm can apply to a less general system (MAP $\rightarrow$ DDN $\rightarrow$ RBN $\rightarrow$ CA) for a reality check. The idea of subtrees, basins of attraction, and the entire "basin of attraction field" imposed on state-space is set out in figure 4.

The dynamical systems considered in this chapter, whether CA, RBN, or DDN, comprise a finite set of n elements with discrete values v, connected by directed links — the wiring scheme. Each element updates its value synchronously, in discrete time-steps, according to a logical rule applied to its k inputs, or a lookup table giving the output of v^k possible input patterns. CA form a special subset with a universal rule, and a regular lattice with periodic boundaries, created by wiring from a homogeneous local neighbourhood, an architecture that can support emergent complex structure, interacting gliders, glider-guns, and universal computation[2, 16, 21, 23, 6]. Langton[5] has aptly described CA as "a discretised artificial universe with its own local physics".

Classical RBN[9] have binary values and homogeneous k, but "random" rules and wiring, applied in modeling gene regulatory networks. DDN provides a further generalisation allowing values greater than binary and heterogeneous k, giving insights into content addressable memory and learning[19]. There are countless variations, intermediate architectures, and hybrid systems, between CA and DDN. These systems can also be seen as instances of "random maps with out-degree one" (MAP)[19, 26], a list of "exhausive pairs" where each state in state-space is assigned a random sucessor, possibly with some bias. All these systems reorganise state-space into basins of attraction.

Running a CA, RBN or DDN backwards in time to trace all possible branching ancestors opens up new perspectives on dynamics. A forward "trajectory" from some initial state can be placed in the context of the "basin of attraction field" which sums up the flow in state-space leading to attractors. The earliest reference I have found to the concept is Ross Ashby's "kinematic map"[1].



For a binary network size n, an example of one of its states B might be $1010\ldots0110$. *State-space* is made up of all $S=2^n$ states ($S=v^n$ for multi-value) — the space of all possible bitstrings or patterns.

Part of a *trajectory* in state-space, where C is a successor of B, and A is a *pre-image* of B, according to the dynamics of the network.

The state B may have other pre-images besides A, the total number is the *in-degree*. The pre-image states may have their own pre-images or none. States without pre-images are known as *garden-of-Eden* or *leaf* states.

Any trajectory must sooner or later encounter a state that occurred previously - it has entered an attractor cycle. The trajectory leading to the attractor is a *transient*. The period of the attractor is the number of states in its cycle, which may be just one - a point attractor.

Take a state on the attractor, find its pre-images (excluding the pre-image on the attractor). Now find the pre-images of each pre-image, and so on, until all leaf states are reached. The graph of linked states is a *transient tree* rooted on the attractor state. Part of the transient tree is a subtree defined by its root.

Construct each transient tree (if any) from each attractor state. The complete graph is the *basin of attraction*. Some basins of attraction have no transient trees, just the bare "attractor".

Now find every attractor cycle in state-space and construct its basin of attraction. This is the *basin of attraction field* containing all 2^n states in state-space, but now linked according to the dynamics of the network. Each discrete dynamical network imposes a particular basin of attraction field on state-space.

Figure 4: The idea of subtrees, basins of attraction, and the entire "basin of attraction field" imposed on state-space by a discrete dynamical network.

The term "basins of attraction" is borrowed from continuous dynamical systems, where attractors partition phase-space. Continuous and discrete dynamics share analogous concepts — fixed points, limit cycles, sensitivity to initial conditions. The separatrix between basins has some affinity to unreachable (garden-or-Eden) leaf states. The spread of a local patch of transients mea-

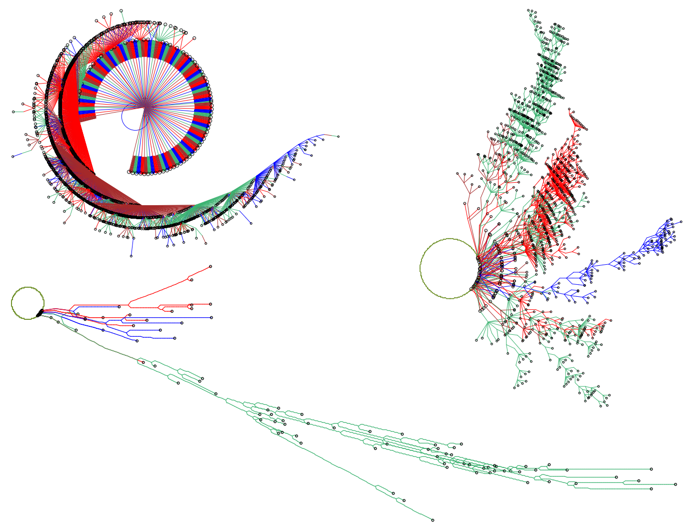


Figure 5: Three basins of attraction with contrasting topology, n=15, k=3, CA rules 250, 110 and 30. One complete set of equivalent trees is shown in each case, and just the nodes of unreachable leaf states. The topology varies from very bushy to sparsely branching, with measures such as leaf density, transient length, and in-degree distribution predicted by the rule's Z-parameter.

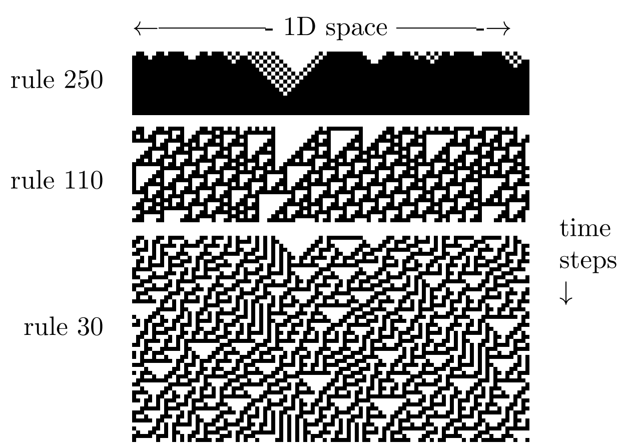


Figure 6: 1D space-time patterns of the k=3 rules in fig.5, characteristic of order, complexity and chaos. System size n=100 with periodic boundaries. The same random initial state was used in each case. A space-time pattern is just one path through a basin of attraction.

sured by the Liapunov exponent has its analog in the degree of convergence or bushiness of subtrees. However, there are also notable differences. For example, in discrete systems trajectories are able to merge outside the attractor, so a sub-partition or sub-category is made by the root of each subtree, as well as by attractors.

The various parameters and measures of basins of attraction in discrete dynamics are summarised in the remainder of this chapter[1], together with some insights and applications, firstly for CA, then for RBN/DDN.

Basins of attraction in CA

Notions of order, complexity and chaos, evident in the space-time patterns of single trajectories, either subjectively (figure 6) or by the variabiliy of input-entropy (figure 7, 10), relate to the topology of basins of attraction (figure 5). For order, subtrees and attractors are short and bushy. For chaos long and sparsely branching (figure 12). It follows that leaf density for order is high because each forward time-step abandons many states in the past, and unreachble by further forward dynamics — for chaos the opposite is true, with very few states abandoned.

This general law of convergence in the dynamical flow applies for DDN as well as CA, but for CA it can be predicted from the rule itself by its Z-parameter (figure 8), the probability that the next unknown cell in a pre-image can be derived unambiguously by the CA reverse algorithm[15, 16, 21]. As Z is tuned from 0 to 1, dynamics shift from order to chaos (figure 8), with transient/attractor length (figure 5), leaf density (figure 9), and the in-degree frequency histogram[21, 26] prividing measures of convergence.

CA rotational symmetry

CA with periodic boundary conditions, a circular array in 1d (or a torus in 2d), impose restrictions and symmetries on dynamical behaviour and thus on basins of attraction. The "rotational symmetry" is the maximum number of repeating segments s into which the ring can be divided. The size of a repeating segment g is the minimum number of cells through which the circular array can be rotated and still appear identical. The array size n=$s \times g$. For uniform states (i.e. 000000...) s=n and g=1. If n is prime, for any non-uniform state s=1 and g=n.

It was shown in [15] that s cannot decrease, may only increase in a transient, and must remain constant on the attractor. So uniform states must occur later in time than any other state — close to or on the attractor, followed by states consisting of repeating pairs (i.e. 010101.. where g=2), repeating triplets, and so on. It follows that each state is part of a set of g equivalent states, which make equivalent subtrees and basins of attraction[15, 26, 27]. This allows the automatic regeneration of subtrees once a prototype subtree has been computed,

[1]This review is based on the author's prior publications [15] to [27] and especially [25]

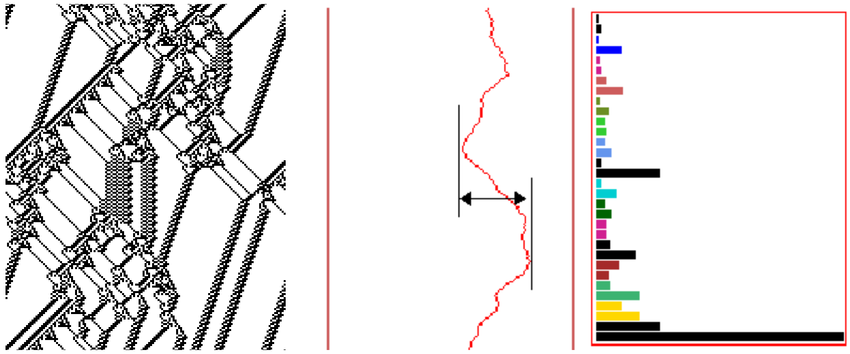


Figure 7: *Left:* The space-time patterns of a 1D complex CA, n=150 about 200 time-steps. *Right:* A snapshot of the input frequency histogram measured over a moving window of 10 time-steps. *Centre:* The changing entropy of the histogram, its variability providing a non-subjective measure to discriminate between ordered, complex and chaotic rules automatically. High variability implies complex dynamics. This mesure is used to automaticaly categorise rule-space[21, 26] (figure 10).

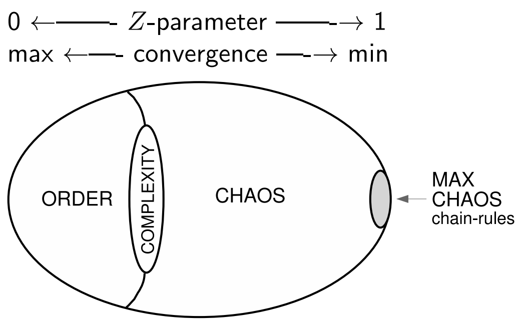


Figure 8: A view of CA rule-space, after Langton[5]. Tuning the Z-parameter from 0 to 1 shifts the dynamics from maximum to minimum convergence, from order to chaos, traversing a phase transition where complexity lurks. The chain-rules on the right are maximally chaotic and have the very least convergence, decreasing with system size, making them suitable for dynamical encryption.

and the "compression" of basins — showing just the non-equivalent prototypes, (figure 1).

CA equivalence classes

Binary CA rules fall into equivalence classes[14, 15] consisting of a maximum of four rules, whereby every rule R can be transformed into its "negative" R_n, its "reflection" R_r and it "negative/reflection" R_{nr}. Rules in an equivalence class have equivalent dynamics thus basins of attraction. For example, the 256 $k3$ "elementary rules" fall into 88 equivalence classes whose description suffices to

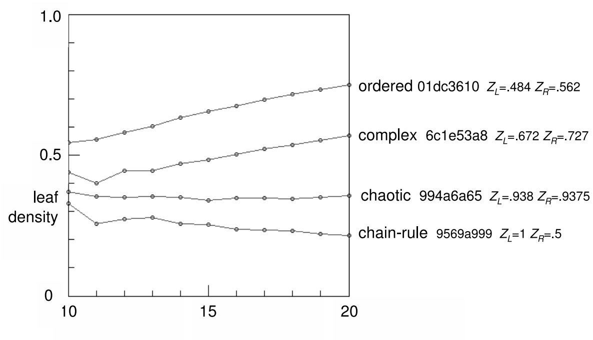


Figure 9: Leaf (garden-of-Eden) density plotted against system size n, for four typical CA rules, reflecting convergence which is predicted by the Z-parameter. Only the maximally chaotic chain-rules show a decrease. The measures are for the basin of attraction field, so for the entire state-space. k=5, n= 10 to 20.

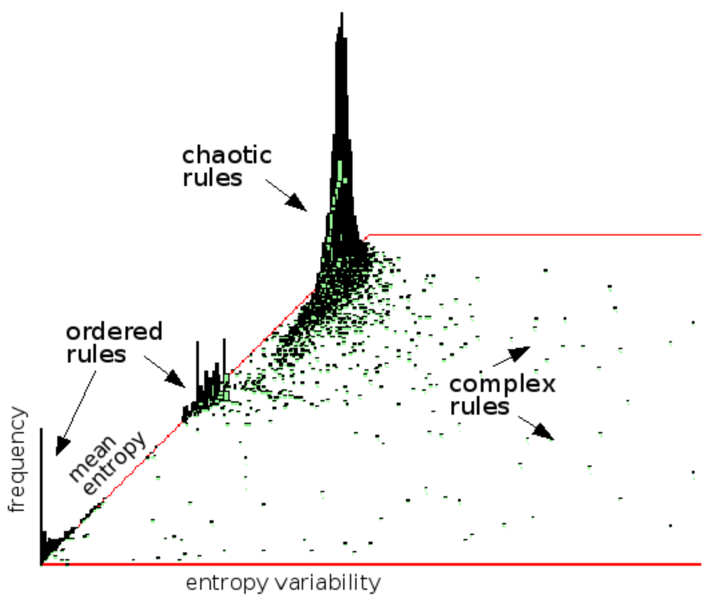
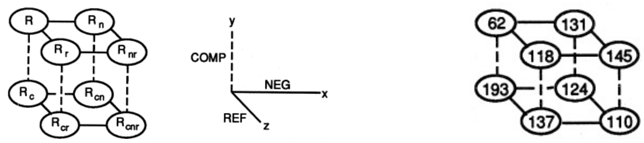
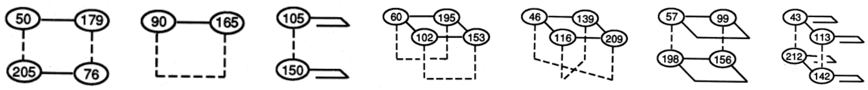


Figure 10: Scatter plot of a sample of 15800 2d hexagonal CA rules (v=3, k=6), plotting mean entropy against entropy variability[21, 26], which classifies rules between ordered, complex and chaotic. The vertical axis shows the frequency of rules at positions on the plot — most are chaotic. The plot automatically classifies rule-space as follows,

	order	complexity	chaos
mean entropy	low	medium	high
entropy variability	low	high	low



Left: 8-rule cluster schematic. *Right*: A non-collapsable rule cluster, total of type=20.



Examples of collapsed clusters, totals of each type in order: 12, 4, 4, 2, 2, 2, 2.

Figure 11: Graphical representation of rule clusters of the $v2k3$ "elementary" rules, and examples, taken from [15], where its shown that the 256 rules in rule-space breaks down into 88 equivalence classes and 48 clusters. The rule cluster is depicted as 2 complimentary sets of 4 equivalent rules at the corners of a box — with negative, reflection, and complimentary transformation links on the x, y, z edges, but these edges may also collapse due to identities between a rule and its transformation.

characterise rule-space, and there is a further collapse to 48 "rule clusters" by a complimentary transformation(figure 11). Equivalence classes can be combined with their compliments to make "rule clusters" which share many measures and properties[15], including the Z-parameter, leaf-density, and Derrida plot. Likewise, the 64 $k5$ totalistic rules fall into 36 equivalence classes.

CA glider interaction and basins of attraction

Of exceptional interest in the study of CA is the phenomenon of complex dynamics. Self-organization and emergence of stable and mobile interacting particles, gliders and glider-guns, enables universal computation at the "edge of chaos"[5]. Notable examples studied for their particle collision logic are the 2D "game-of-Life"[2], the elementary rule 110[12], and the hexagonal 3-value spiral-rule[23]. More recently discovered is the 2D binary X-rule and its offshoots[6, 7].

Here we will simply comment on complex dynamics seen from a basin of attraction perspective[16, 3], were basin topology, and the various measures such as leaf density, in-degree distribution and the Z-parameter are intermediate between order and chaos. Disordered states, before the emergence of particles and their backgrounds make up leaf states or short dead-end side branches along the length of long transients where particle interactions are progressing. States dominated by particles and their backgrounds are special, a small subcategory of state-space. They constitute the glider interaction phase, making up the main lines of flow within long transients. Gliders in their interaction phase can be regarded as competing sub-attractors. Finally, states made up solely periodic glider interactions, non-interacting gliders, or domains free of gliders, must cycle and therefore constitute the relatively short attractors.

Information hiding within chaos

State-space by definition includes every possible piece of information encoded within the size of the CA lattice — including Shakespeare's sonnets, copies of the Mona Lisa, one's own thumb print, but mostly disorder. A CA rule organises state-space into basins of attraction where each state has its specific location, and where states on the same transient are linked by forward time-steps, so the statement "state $B = A + x$ time-steps" is legitimate. But the reverse "state $A = B - x$" is usually not legitimate because backward trajectories will branch by the in-degree at each backward step, and the correct branch must be selected. More importantly, most states are leaf states without pre-images, or close to the leaves, so for these states "$-x$" time-steps would not exist.

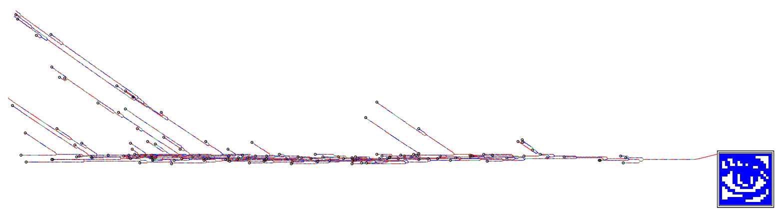


Figure 12: A subtree of a chain-rule 1D CA n=400. The root state (the eye) is shown in 2d (20×20). Backwards iteration was stopped after 500 reverse time-steps. The subtree has 4270 states. The density of both leaf states and states that branch is very low (about 0.03) - where maximum branching equals 2.

In-degree, convergence in the dynamical flow, can be predicted from the CA rule itself by its Z-parameter, the probability that the next unknown cell in a pre-image can be derived unambiguously by the CA reverse algorithm[15, 16, 21]. This is computed in two direction, Z_{left} and Z_{right}, with the higher value taken as Z. As Z is tuned from 0 to 1, dynamics shift from order to chaos (figure 8), with leaf density, a good measure of convergence, decreasing (figures 5, 9). As the system size increases, convergence increases for ordered rules, at a slower rate for complex rules, and remains steady for chaotic rules which make up most of rule-space (figure 10). However, there is a class of maximally chaotic "chain" rules where Z_{left} XOR Z_{right} equals 1, where convergence and leaf density decrease with system size n (figure 9). As n increases, in-degrees$\geq$ 2, and leaf density, become increasingly rare (figure 12), and vanishingly small in the limit. For large n, for practical purposes, transients are made up of long chains of states without branches, so it becomes possible to link two states separated in time, both forwards and backwards. Figure 13 describes how information can be encryped and decrypted, in this example for an 8-value (8 color) CA. About the square root of binary rule-space is made up of chain rules, which can be constructed at random to provide a huge number of encryption keys.

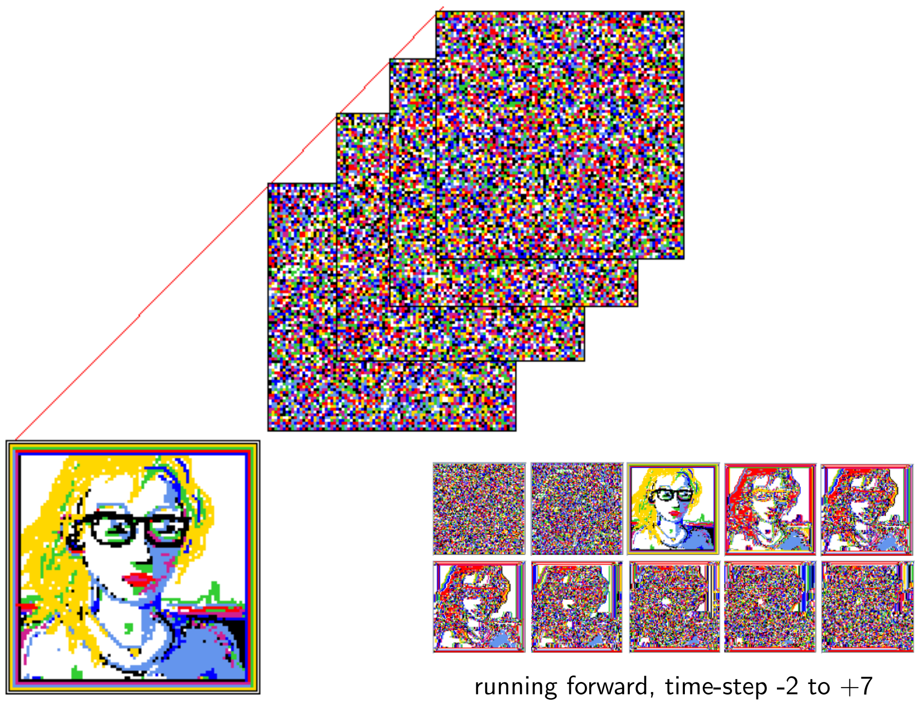


Figure 13: *Left*: A 1D pattern is displayed in 2D (n=7744, 88×88). The "portrait" was drawn with the drawing function in DDLab. With a v=8, k=4 chain-rule constructed at random, and the portrait as the root state, a subtree was generated with the CA reverse algorithm, set to stop after 4 backward time-steps. The state reached is the encryption. To decrypt, run forwards by the same number of time-steps. *Right*: Starting from the encrypted state the CA was run forward to recover the original image. This figure shows time-steps -2 to +7 to illustrate how the image was scrambled both before and after time step 0.

Memory and learning

The RBN basin of attraction field (figure 15) reveals that content addressable memory is present in discrete dynamical networks, and shows its exact composition, where the root of each subtree (as well as each attractor) categorises all the states that flow into it, so if the root state is a trigger in some other system, all the states in the subtree could in principle be recognised as belonging to a particular conceptual entity. This notion of memory far from equilibrium [17, 18] extends Hopfield's[4] and other classical concepts of memory in artificial neural networks, which rely just on attractors.

As the dynamics descend towards the attractor, a hierarchy of sub-categories unfolds. Learning in this context is a process of adapting the rules and connections in the network, to modify sub-categories for the required behaviour —

modifying the fine structure of subtrees and basins of attraction. Classical CA are not ideal systems to implement these subtle changes, restricted as they are to a universal rule and local neighbourhood, a requirement for emergent structure, but which severely limits the flexibility to categorise. Moreover, CA dynamics have symmetries and hierarchies resulting from their periodic boundaries[15]. Nevertheless, CA can be shown to have a degree of stability in behaviour when mutating bits in the rule-table – with some bits more sensitive than others. The rule can be regarded as the genotype and basins of attractionn as the phenotype[15]. Figure 14 shows CA mutant basins of attraction.

With RBN and DDN there is greater freedom to modify rules and connections than with CA. Algorithms for learning and forgetting[17, 18, 19] have been devised, implemented in DDLab. The methods assign pre-images to a target state by correcting mismatches between the target and the actual state, by flipping specific bits in rules or by moving connections. Among the side effects, generalisation is evident, and transient trees are sometimes transplanted along with the reassigned pre-image.

Modelling neural networks

Allowing some conjecture and speculation, what are the implications of the basin of attraction idea on memory and learning in animal brains[17, 18]? The fist conjecture, perhaps no longer controversial, is that the brain is a dynamical system (not a computer or Turing machine) composed of interacting sub-networks. Secondly, neural coding is based on distributed patterns of activation in neural sub-networks (not the frequency of firing of single neurons) where firing is

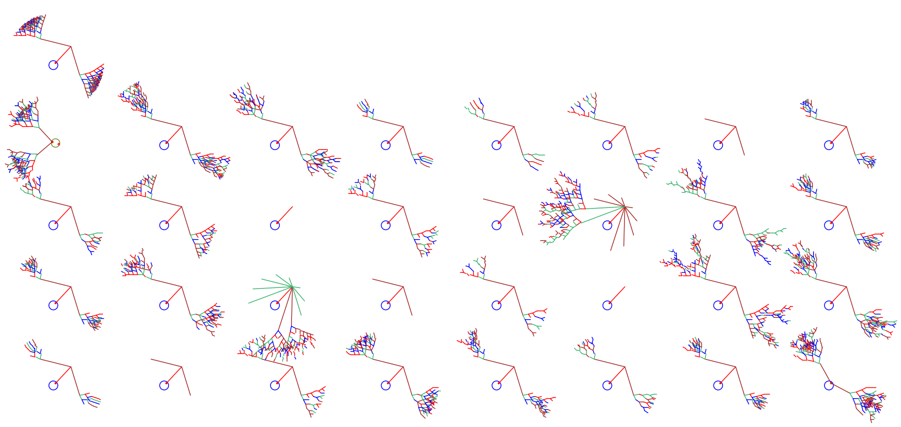


Figure 14: Mutant basins of attraction of the v=2, k=3, rule 60 (n=8, seed all 0s). *Top left*: The original rule, where all states fall into just one very regular basin. The rule was first transformed to its equivalent k=5 rule (f00ff00f in hex), with 32 bits in its rule table. All 32 one-bit mutant basins are shown. If the rule is the genotype, the basin of attraction can be seen as the phenotype.

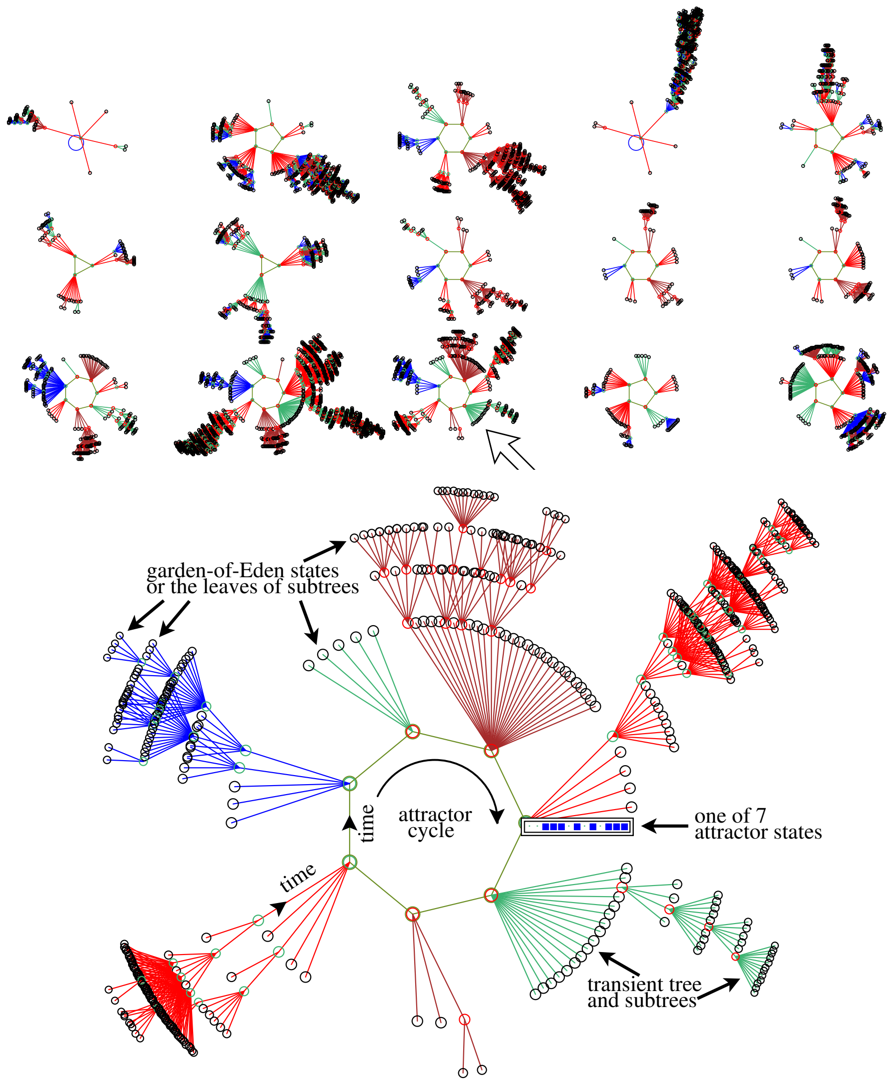


Figure 15: <u>*Top:*</u> The basin of attraction field of a random Boolean network, k=3, n=13. The $2^{13} = 8192$ states in state-space are organised into 15 basins, with attractor periods ranging between 1 and 7, and basin volume between 68 and 2724. <u>*Bottom:*</u> A basin of attraction (arrowed above) which links 604 states, of which 523 are leaf states. The attractor period = 7, and one of the attractor states is shown in detail as a bit pattern. The direction of time is inwards, then clock-wise at the attractor.

synchronised by many possible mechanisms: phase locking, inter-neurons, gap junctions, membrane nanotubes, ephaptic interactions.

Learnt behaviour and memory work by patterns of activation in sub-networks flowing automatically within the subtrees of basins of attraction. Recognition is easy because an initial state is provided. Recall is difficult because an association must be conjured up to initiate the flow within the correct subtree.

At a very basic level, how does a DDN model a semi-autonomous patch of neurons in the brain whose activity is synchronised? A network's connections model the subset of neurons connected to a given neuron. The logical rule at a network element, which could be replaced by the equivalent tree-like combinatorial circuit, models the logic performed by the synaptic micro-circuitry of a neuron's dendritic tree, determining whether or not it will fire at the next time-step. This is far more complex than the threshold function in artificial neural networks. Learning involves changes in the dendritic tree, or more radically, axons reaching out to connect (or disconnect) neurons outside the present subset.

Modelling genetic regulatory networks

The various cell types of multicellular organisms, muscle, brain, skin, liver and so on (about 210 in humans) have the same DNA so the same set of genes. The different types result from different patterns of gene expression. But how do the patterns maintain their identity? How does the cell remember what it is supposed to be?

It is well known in biology that there is a genetic regulatory network, where genes regulate each other's activity with regulatory proteins[13]. A cell type depends on its particular subset of active genes, where the gene expression pattern needs to be stable but also adaptable. More controversial to cell biologists less exposed to complex systems, is Kauffman's classic idea[9, 10, 20] that the genetic regulatory network is a dynamical system where cell types are attractors which can be modelled with the RBN or DDN basin of attraction field. However, this approach has tremendous explanatory power and it is difficult to see a plausible alternative.

Kauffman's model demonstrates that evolution has arrived at a delicate balance between order and chaos, between stability and adaptability, but leaning towards convergent flow and order[10, 8]. The stability of attractors to perturbation can be analysed by the jump-graph (figure 16) which shows the probability of jumping between basins of attraction due to single bit-flips (or value-flips) to attractor states[22, 26]. These methods are implemented in DDLab and generalised for DDN where the value range, v, can be greater that 2 (binary), so a gene can be fractionally on as well as simply on/off.

A present challenge in the model, the inverse problem, is to infer the network architecture from information on space-time patterns, and apply this to infer the real genetic regulatory network from the dynamics of observed gene expression[8].

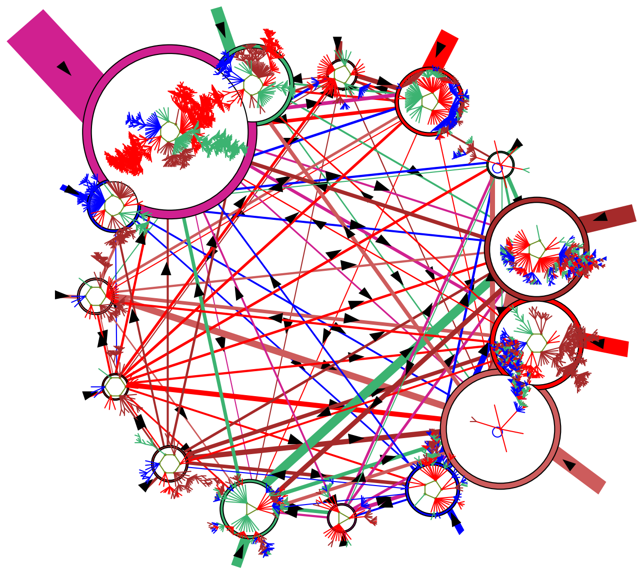


Figure 16: The jump-graph (of the same RBN as in figure 15) shows the probability of jumping between basins due to single bit-flips to attractor states. Nodes representing basins are scaled according the number of states in the basin (basin volume). Links are scaled according to both basin volume and the jump probability. Arrows indicate the direction of jumps. Short stubs are self-jumps; more jumps return to their parent basin than expected by chance, indicating a degree of stability. The relevant basin of attraction is drawn inside inside each node.

Future directions

This chapter has reviewed a variety of discrete dynamical networks where knowledge of the structure of their basins of attraction provides insights and applications: in complex cellular automata particle dynamics and self-organisation, in maximally chaotic cellular automata where information can be hidden and recovered from a stream of chaos, and in random Boolean and multi-value networks that are applied to model neural and genetic networks in biology. Many avenues of enquiry remain — whatever the discrete dynamical system, its worthwhile to think about it from the basin of attraction perspective.

References

[Note] Most references by A.Wuensche are available online at `http://www.uncomp.ac.uk/wuensche/publications.html`

[1] Ashby W.R., "An Introducion to Cybernetics", Chapman & Hall, 1956.

[2] Conway,J.H., (1982) "What is Life?" in "Winning ways for your mathematical plays", Berlekamp,E, J.H.Conway and R.Guy, Vol.2, chap.25, Academic Press, New York.

[3] Domain C., and H Gutowitz, "The Topological Skeleton of Cellular Automata Dynamics", Pysica D, Vol. 103, Nos. 1-4, 155-168, 1997.

[4] Hopield,J.J., Neural networks and physical systems with emergent collective abilities, Proceeding of the National Academy of Sciences 79 2554-2558, 1982.

[5] Langton, C.G., Computation at the edge of chaos: Phase transitions and emergent computation, Physica D, 42, 12-37, 1990.

[6] Gomez-Soto,JM., and A.Wuensche, "The X-rule: universal computation in a non-isotropic Life-like Cellular Automaton", JCA, Vol 10, No.3-4, 261-294, 2015. preprint: http://arxiv.org/abs/1504.01434/

[7] Gomez-Soto,JM., and A.Wuensche, "X-Rule's Precursor is also Logically Universal", to appear in JCA. preprint: https://arxiv.org/abs/1611.08829/

[8] Harris SE, Sawhill BK, Wuensche A, and Kauffman SA, A Model of Transcriptional Regulatory Networks Based on Biases in the Observed Regulation Rules, Complexity, Vol.7/no.4, 23–40, 2002.

[9] Kauffman SA, Metabolic Stability and Epigenesis in Randomly Constructed Genetic Nets, Theoretical Biology, 22(3), 439–467, 1969

[10] Kauffman SA, The Origins of Order, Oxford University Press,1993.

[11] Kauffman SA, Investigations. Oxford University Press, 2000.

[12] , Cook M, "Universality in Elementary Cellular Automata", Complex Systems 15: 1-40, 2004.

[13] Somogyi R and Sniegoski CA, Modeling the complexity of genetic networks: understanding multigene and pleiotropic regulation, Complexity 1, 45-63, 1996.

[14] Walker,C.C., and W.R.Ashby, (1966) "On the Temporal Characteristics of Behavior in Certain Complex Systems", Kybernetick 3(2), 100-108, 1966.

[15] Wuensche A, and M.J. Lesser. "The Global Dynamics of Cellular Automata; An Atlas of Basin of Attraction Fields of One-Dimensional Cellular Automata", Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley, Reading, MA, 1992.

[16] Wuensche A, Complexity in 1D cellular automata; Gliders, basins of attraction and the Z parameter, Santa Fe Institute Working Paper 94-04-025, 1994.

[17] Wuensche A, The ghost in the machine: Basin of attraction fields of random Boolean networks. In: Artificial Life III, ed. Langton CG, Addison-Wesley, Reading, MA, 496–501, 1994.

[18] Wuensche A, The Emergence of Memory: Categorisation Far From Equilibrium, in Towards a Science of Consciousness: The First Tuscon Discussions and Debates, eds. Hameroff SR, Kaszniak AW and Scott AC, MIT Press, Cambridge, MA, 383–392, 1996.

[19] Wuensche A, Attractor basins of discrete networks: Implications on self-organisation and memory. Cognitive Science Research Paper 461, DPhil Thesis, University of Sussex, 1997.

[20] Wuensche A, Genomic Regulation Modeled as a Network with Basins of Attraction, Proceedings of the 1998 Pacific Symposium on Biocomputing", World Scientific, Singapore, 1998.

[21] Wuensche A, Classifying cellular automata automatically; finding gliders, filtering, and relating space-time patterns, attractor basins, and the Z parameter, Complexity, Vol.4/no.3, 47–66, 1999.

[22] Wuensche A, "Basins of Attraction in Network Dynamics: A Conceptual Framework for Biomolecular Networks", in "Modularity in Development and Evolution", eds G.Schlosser and G.P.Wagner. Chicago University Press, chapter 13, 288-311, 2004.

[23] Wuensche A and A. Adamatzky, On spiral glider-guns in hexagonal cellular automata: activator-inhibitor paradigm, International Journal of Modern Physics C,Vol. 17, No. 7,1009–1026, 2006.

[24] Wuensche A, Cellular Automata Encryption: The Reverse Algorithm, Z-Parameter and Chain Rules, Parallel Processing Letters, Vol 19, No 2, 283–297, 2009.

[25] Wuensche,A., (2010), "Complex and Chaotic Dynamics, Basins of Attraction, and Memory in Discrete Networks", ACTA PYSICA POLONICA B, Vol 3, No 2, 463-478.

[26] Wuensche,A.,"Exploring Discrete Dynamics — Second Edition", Luniver Press, 2016.

[27] Wuensche,A., Discrete Dynamics Lab (DDLab) `http://www.ddlab.org/`, 1993-2017.