COMPLEX AND CHAOTIC DYNAMICS, BASINS OF ATTRACTION, AND MEMORY IN DISCRETE NETWORKS*

Andrew Wuensche†

Discrete Dynamics Lab
www.ddlab.org

(Received February 18, 2010)

The emergence of interacting structures in cellular automata is intimately connected with notions of order, complexity and chaos, which depend on the degree of converge to attractors. Information can be encrypted by hiding in chaotic trajectories. In the general case of "random" networks, content addressable memory is apparent in the precise arrangement of state-space into basins of attraction and subtrees, a concept of memory and learning at the most basic level. This paper is an overview of the ideas, results and applications illustrated with images created in DDLab.

PACS numbers: 89.75.–k

1. Introduction

"The Global Dynamics of Cellular Automata" [1] published in 1992 introduced a reverse algorithm for computing the pre-images (predecessors) of states for any finite 1D binary cellular automata (CA) with periodic boundaries. This made it possible to reveal the precise topology of "basins of attraction" — state transition graphs — states linked into trees rooted on attractor cycles, which could be computed and drawn automatically (Fig. 1).

Not long after, a different reverse algorithm [2] was invented for computing the pre-images of random Boolean networks (RBN) allowing their basins of attraction to be drawn (Fig. 2), just in time to make the cover of Kauffman's 1993 book [3] "The Origins of Order" (Fig. 3). The RBN algorithm is now generalised for the most versatile "discrete dynamical networks" (DDN). These algorithms compute pre-images directly without relying on exhaustive testing of state-space, and are implemented in the software DDLab [4].

* Presented at the Summer Solstice 2009 International Conference on Discrete Models of Complex Systems, Gdańsk, Poland, June 22–24, 2009.

† Send any remarks to andy@ddlab.org

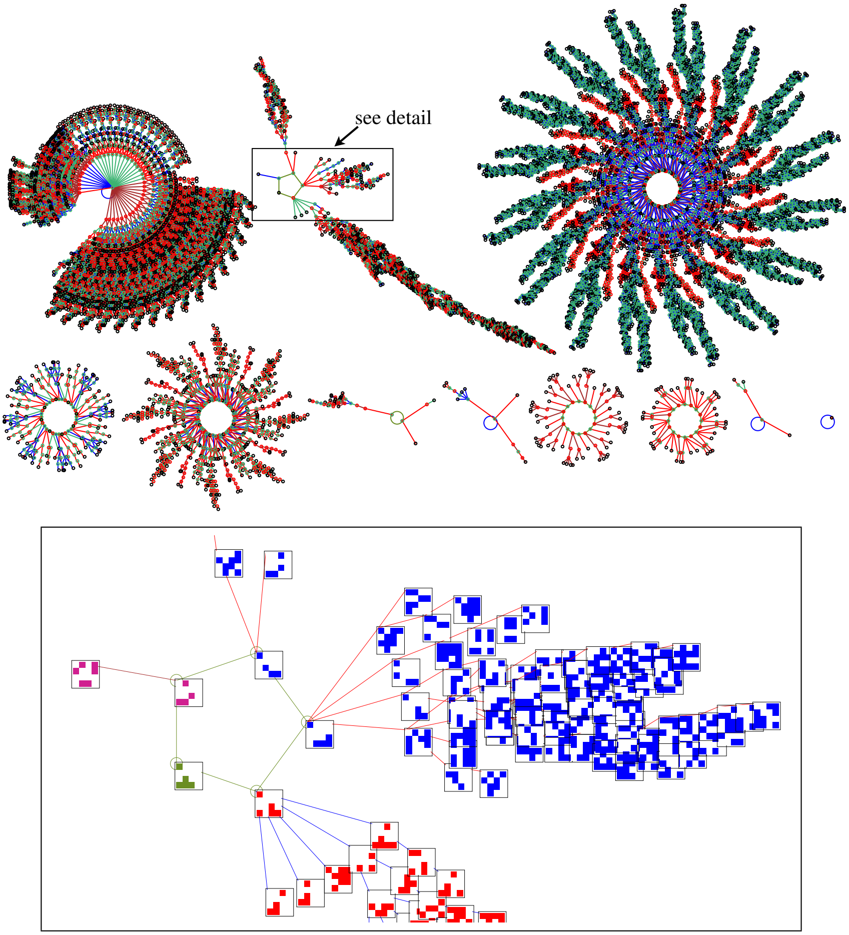

Fig. 1. Top: The basin of attraction field of a 1D binary CA, $n = 16$. The 2^{16} states in state-space are connected into 89 basins of attraction, only the 11 nonequivalent basins are shown, with symmetries characteristic of CA [1]. Time flows inwards, then clockwise at the attractor. Bottom: A detail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

In this paper, a DDN is a finite set of n elements with discrete states or values. Elements are connected with directed links — the wiring scheme. Each element updates its value synchronously according to a logical rule applied to its k inputs — the system updates in discrete time-steps. CA are much more restricted with a universal rule, and a regular lattice with periodic boundaries, created by wiring from a homogeneous local neighbourhood connecting "cells". Langton [5] has aptly described CA as "a discretised artificial universe with is own local physics". Obviously there are countless

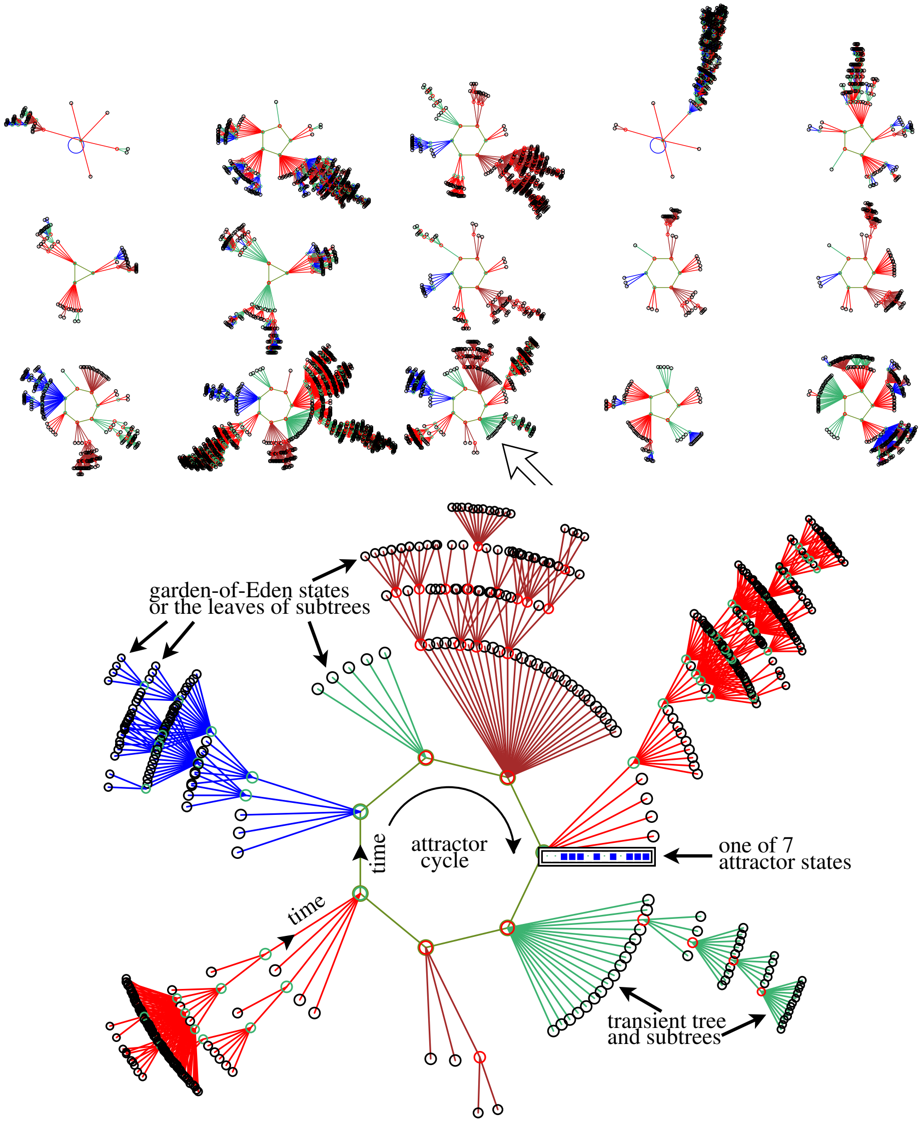

Fig. 2. Top: The basin of attraction field of a random Boolean network, $k = 3$, $n = 13$. The $2^{13} = 8192$ states in state-space are organised into 15 basins, with attractor periods ranging between 1 and 7, and basin volume between 68 and 2724. Bottom: A basin of attraction (arrowed above) which 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.

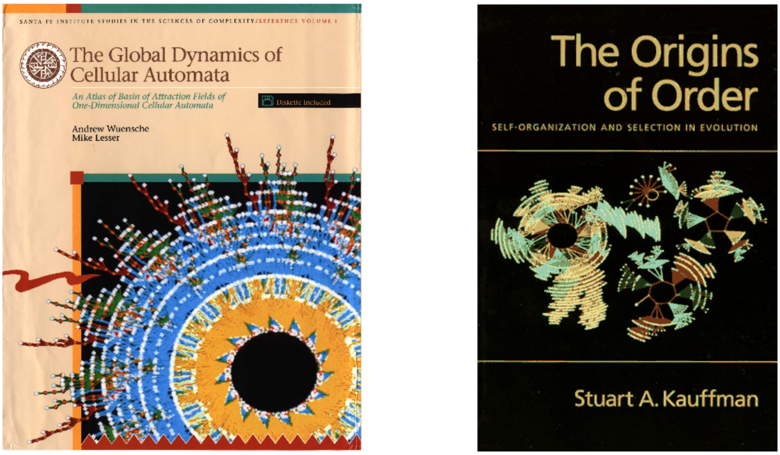
SANTA FE INSTITUTE STUDIES IN THE SCIENCES OF COMPLEXITY/REFERENCE VOLUME I
The Global Dynamics of Cellular Automata
An Atlas of Basin of Attraction Fields of One-Dimensional Cellular Automata
Diskette Included
Andrew Wuensche
Mike Lesser
The Origins of Order
SELF-ORGANIZATION AND SELECTION IN EVOLUTION
Stuart A. Kauffman

Fig. 3. The front covers of Wuensche and Lesser's (1992) "The Global Dynamics of Cellular Automata" [1] and Kauffman's (1993) "The Origins of Order" [3]. A basin of attraction of a CA, and the basin of attraction field of a RBN, where computed and drawn with the precursor of DDLab.

variations and intermediate architectures between DDN and CA — classical RBN [6] have binary values and homogeneous k, but all these systems reorganise state-space into basins of attraction.

Running a CA, RBN or DDN backwards in time, tracing all possible branching ancestors, opens up new perspectives on dynamics. A trajectory from some initial state can be placed in the context of the flow in state-space leading to attractors, analogous to Poincaré's "phase portrait" in continuous dynamics, but applied to systems (as in nature) where time and space are discrete. The implications were discussed by Langton in his foreword to [1]. Continuous and discrete dynamics share analogous concepts: fixed points, limit cycles, chaos, sensitivity to initial conditions, and chaotic attractors. The seperatrix or boundary between basins has some affinity to unreachable (garden-or-Eden) leaf states. The spreading of a local patch of transients measured by the Liapunov exponent has its analog in the degree of convergence or bushiness of subtrees — the in-degree of a typical state, predicted by the Z-parameter [1, 7].

The list of analogies and disparities could go on (and deserves an in-depth dissertation) but three significant behavioural phenomena, reviewed in this paper, are arguably more palpable in discrete as opposed to continuous dynamics: complexity by interacting structure — for insights into self-organisation, information hiding within chaos — with applications for encryption, and the concept of memory and learning at the most basic level — for modelling neural and genetic networks.

2. Complexity by interacting structure

In some rare CA, static and mobile interacting structures or particles — gliders and glider-guns, emerge and then dominate the dynamics. Particular examples, the Game-of-Life, the 1D rule 110, and the recently discovered spiral-rule [8] (Fig. 4), are studied for their particle collision logic and (universal) computational properties.

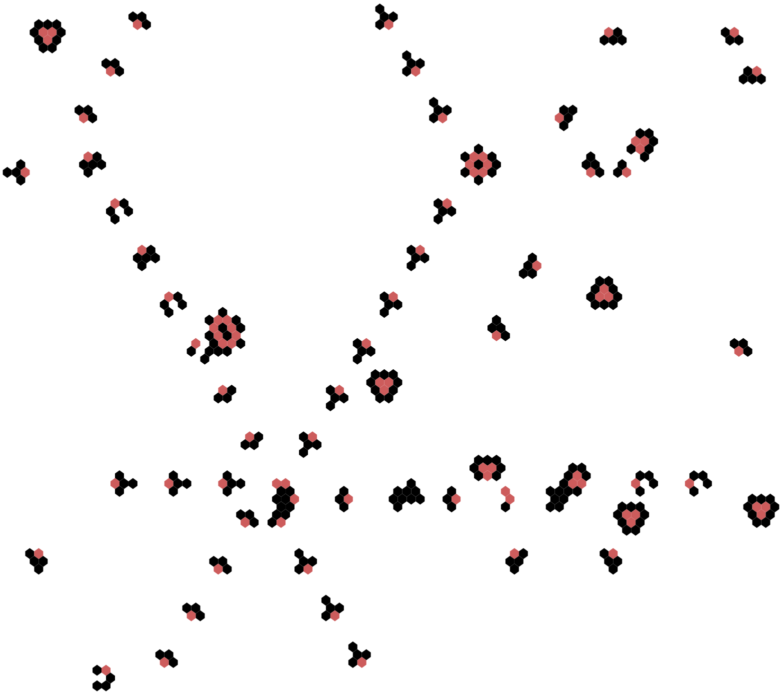

Fig. 4. A snapshot of the 3-value 2D CA spiral-rule [8] on a hexagonal lattice. $n = 88 \times 88$, $k = 7$, where a diversity of interacting mobile and static structures emerge: spiral glider-guns, mobile glider-guns, self-reproduction by glider collisions. A glider moves in the direction of its red (grey) nose.

From another perspective, these are extremely simple and completely defined systems, yet still able to self-organise ever more complex compound structures. Emergence seems unpredictable, open-ended, limited only by the size of the lattice. Like nature, this complex CA behaviour can be described at ascending levels — from the underlying "physics", to observed "laws" of particle collisions, to descriptions of increasingly complex interactions, reaching out — it might be argued — into Kauffman's "adjacent possible" [9]. From this perspective, the complexity of a system is the number of its existential levels of description [10].

The question arises, what is self-organisation? It is well accepted that complex rules are rare, and occur at a transition in rule-space between order and chaos [5] (Fig. 5) — but finding them is not straight forward. A wide variety of complex rules are probably needed to uncover general principles of self-organisation. Unlimited samples of complex rules can be found by using input-entropy to automatically classify rule-space between order, complexity and chaos [7,8].

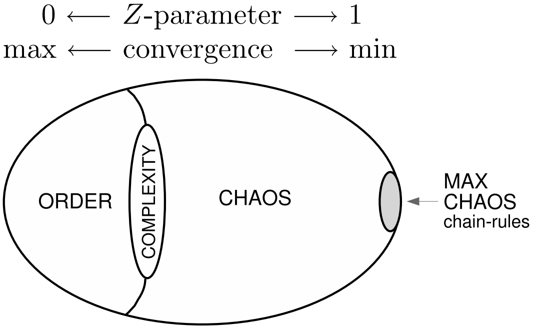

Fig. 5. A view of rule-space (after Langton [5]). Tuning the Z-parameter from 0 to 1 shifts the dynamics from maximum to minimum convergence, from order to chaos, traversing a phase transition where complexity lurks. The chain-rules on the right are maximally chaotic and have the very least convergence,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

Figs. 6 and 7 illustrate the method (implemented in DDLab) which tracks the Shannon entropy of the frequency of rule-table inputs generating space-time patterns, and the entropy variability[1], giving the following results,

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

Only complex rules have high entropy variability so can be separated — mean entropy separates order and chaos (Fig. 7). High variability implies large scale structural interactions, often produced by particles colliding, because collisions create local chaos raising the entropy, from which particles re-emerge lowering entropy.

[1] The variability is taken as the standard deviation, or alternatively as the maximum interval between a minimum followed by a maximum entropy.

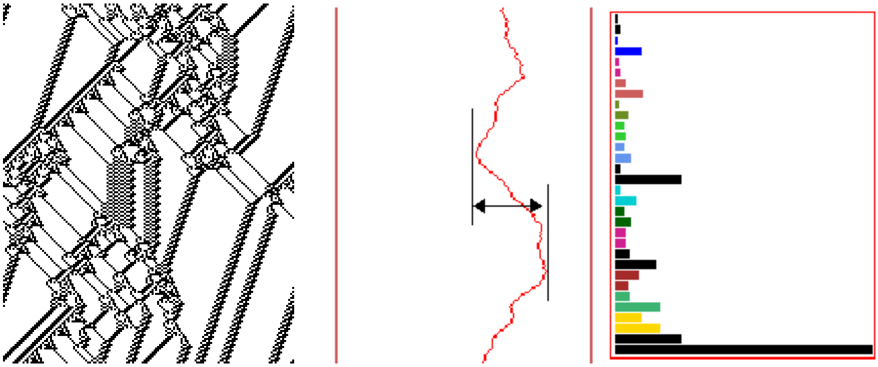

Fig. 6. Left: The space-time patterns of a 1D complex CA, n-150 about 200 time-steps. Right: A snapshot of the input frequency histogram measured over a moving window of 10 time-steps. Centre: The changing entropy of the histogram, its variability providing a non-subjective measure to discriminate between ordered, complex and chaotic rules automatically. High variability implies complex dynamics.

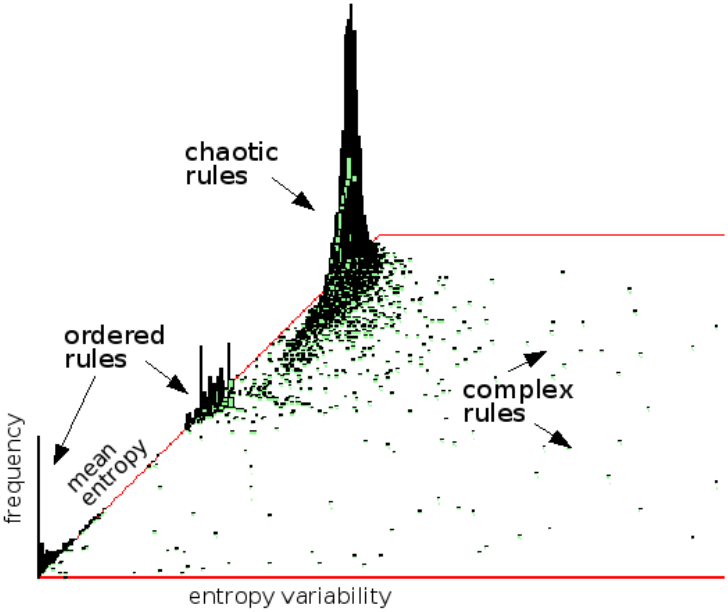

Fig. 7. Scatter plot of a sample of 15800 2d hexagonal CA rules ($v = 3$, $k = 6$), plotting mean entropy against entropy variability, which classifies rules between ordered, complex and chaotic. The vertical axis shows the frequency of rules at positions on the plot — most are chaotic. The plot automatically classifies rule-space.

Particle dynamics can be seen from a basin of attraction perspective. Disordered states, before the emergence of particle dynamics, make up leaf states or short dead-end side branches along the length of long transients where particle interactions are progressing. The final particle survivors persist in attractors.

3. 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 "$A = B - x$" 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.

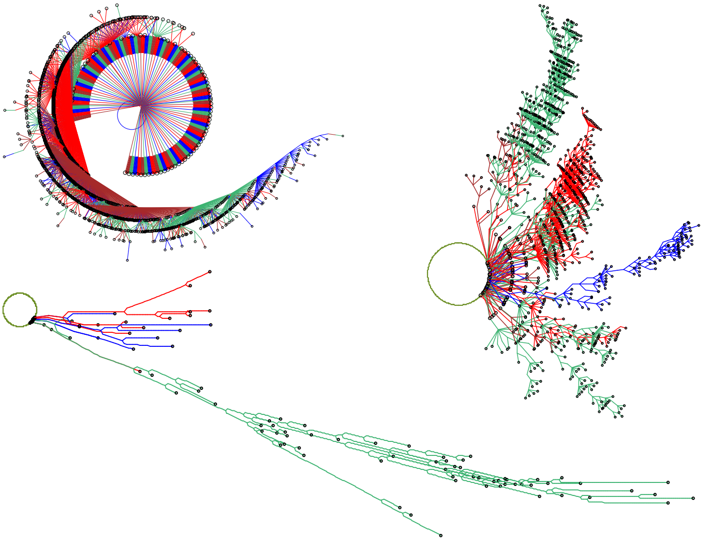

Fig. 8. Three basins of attraction with contrasting topology, $n = 15$, $k = 3$, rules 250, 110 and 30. The time flows inward towards the attractor, then clockwise. One complete set of equivalent trees is shown in each case, and just the nodes of unreachable leaf states. The topology varies from very bushy to sparsely branching, with measures such as leaf density, transient length, and in-degree distribution predicted by the rule's Z-parameter.

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 [1,7,10]. 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 (Fig. 5), with leaf density, a good measure of convergence, decreasing (Figs. 8 and 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 (Fig. 10).

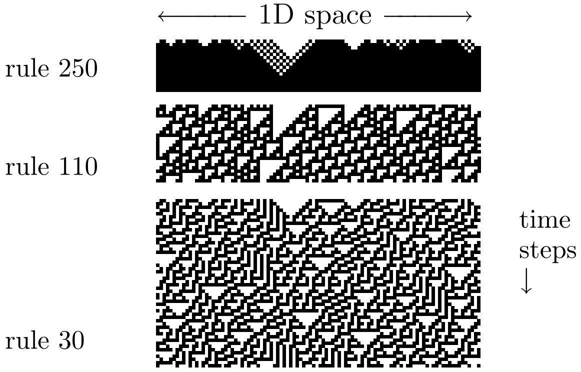

Fig. 9. 1D space-time patterns of the $k = 3$ rules in Fig. 8, characteristic of order, complexity and chaos. System size $n = 100$ with periodic boundaries. The same random initial state was used in each case. A space-time pattern is just one path through a basin of attraction.

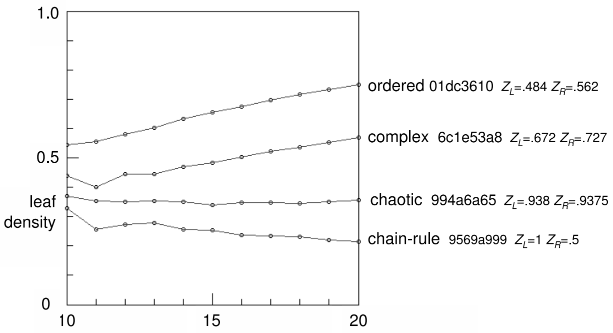

Fig. 10. Leaf (Garden-of-Eden) density plotted against system size n, for four typical CA rules, reflecting convergence which is predicted by the Z-parameter. Only the maximally chaotic chain-rules show a decrease. The measures are for the basin of attraction field, so for the entire state-space. $k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

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 (Fig. 10). As n increases, in-degrees ≥ 2, and leaf density, become increasingly rare (Fig. 11), and vanishingly small in the limit. For large n, for practical purposes, transients are made up of long chains of states without branches (Fig. 12), so it becomes possible to link two states both forwards and backwards.

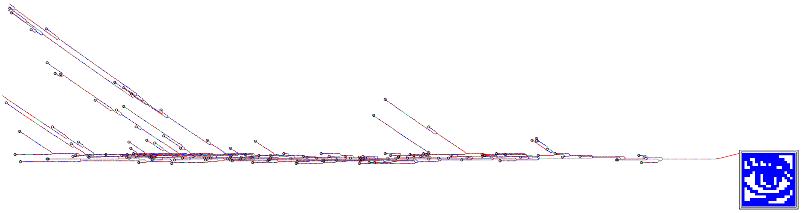

Fig. 11. A subtree of a chain-rule 1D CA $n = 400$. The root state (the eye) is shown in 2d (20×20). Backwards iteration was stopped after 500 reverse time-steps. The subtree has 4270 states. The density of both leaf states and states that branch is very low (about 0.03) — where maximum branching equals 2.

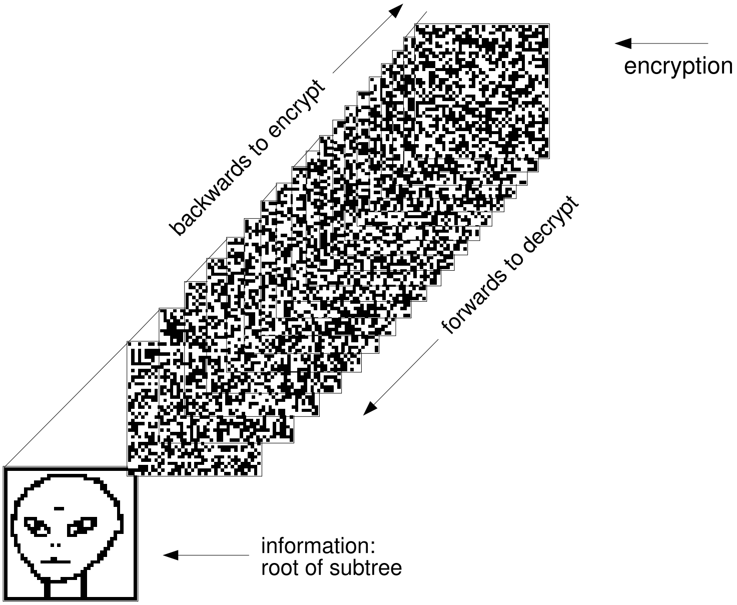

Fig. 12. A subtree of a $k = 7$ 1D CA chain-rule encrypting the "alien" root, a 1d bit-pattern displayed in 2d ($n = 1600$, 40×40), but which could be ASCII, or any other form of information. Backwards iteration was stopped after 19 time-steps.

Suppose B is a state with information. It can be encrypted (Fig. 12) by iterating backwards to $A = B - x$ with the CA reverse algorithm, which is especially efficient for chain rules. A can be decrypted (Fig. 13) by running forward by x time-steps using the correct rule — the encryption key. 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. Fig. 13 shows the information bearing state embedded in chaotic states along the transient. It pops suddenly out of chaos, then merges back into chaos, limited by the "speed of light" of the 1D CA. The methods [11] are implemented in DDLab.

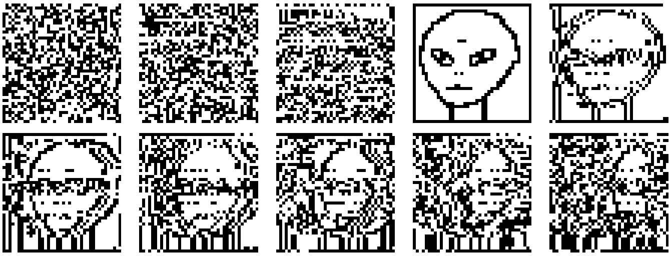

Fig. 13. To decrypt, starting from the encrypted state in Fig. 12, run forward by 19 time-steps with the same chain-rule. This figure shows time-steps -3 to $+6$ to illustrate how the "alien" image was scrambled both before and after time step 0.

4. Memory and learning

The basin of attraction field (Fig. 2) 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 [2, 12] extends Hopfield's [13] 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 subcategories 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 [1]. Nevertheless, CA can be shown to have a degree of stability in behaviour when mutating bits in the rule-table — with some bits are more sensitive than others. The rule can be regarded as the genotype and behaviour (space-time or basins of attraction) as the phenotype [1]. Figure 14 shows CA mutant basins of attraction.

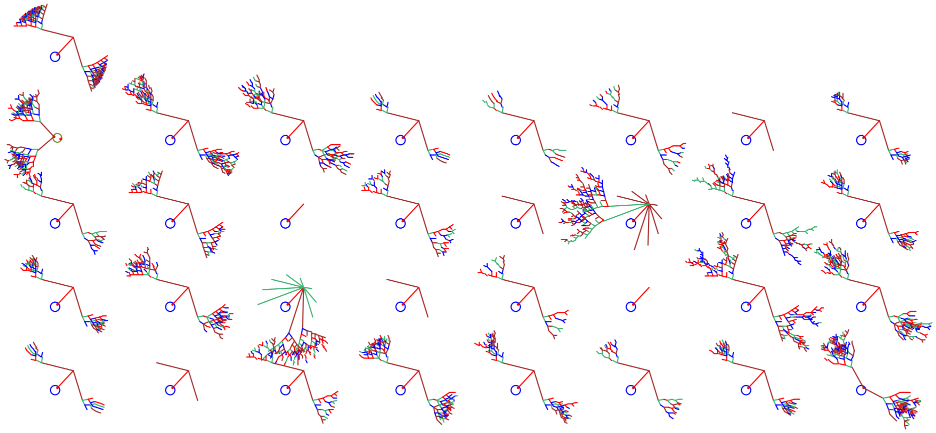

Fig. 14. Mutant basins of attraction of the $v = 2$, $k = 3$, rule 60 ($n = 8$, seed all 0s). Top left: The original rule, where all states fall into just one very regular basin. The rule was first transformed to its equivalent $k = 5$ rule (f00ff00f in hex), with 32 bits in its rule table. All 32 one-bit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

With RBN and DDN there is greater freedom to modify rules and connections than with CA. Algorithms for learning and forgetting [2,12,14] 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.

4.1. Modelling neural networks

Allowing some conjecture and speculation, what are the implications on memory in animal brains? 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 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 subnetworks flowing automatically within the subtrees of basins or 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 microcircuitry 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.

4.2. 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 [15]. 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 [3, 6] that the genetic regulatory network is a dynamical system where cell types are attractors, and which he modelled with RBN. However, this approach has tremendous explanatory power and it is difficult to see a plausible alternative [16].

A gene is regulated by proteins from other genes, that might include itself. On a molecular level, a combination of regulatory proteins, binding to a promoter sequence, turns the gene on and off. At a macro level this determines the rate at which the gene transcribes RNA to produce its specific protein, which can be measured by micro-array analysis.

In Kauffman's model based on RBN, a gene's state is either on or off, its connections are the set of genes that provide its regulatory proteins. Its rule (Boolean function) represents how the proteins combine on the binding site to determine the gene's state. Kauffman ran his model from numerous initial states to identify the main attractors — transient length and basin volume can be inferred statistically, a method also implemented in DDLab and useful for large ordered system[2]. The results showed that the number of inputs, k, was a key variable for the number of attractors [3]. The approach emphasises the order/chaos balance in the dynamics, which depends on k, or alternatively on canalising inputs, a bias for larger k inducing order [17]. Order/chaos measures include the Derrida plot, damage spread, frozen genes, and attractor distribution. The locality of random wiring, reducing the wiring spread when the network is laid out in a regular lattice, also induces order [14]. The new methods to compute basins of attraction of RBN in complete detail [2] provided further insights into the model [15,17].

In a cell type's gene expression pattern (its space-time pattern), a particular gene may spend a large proportion of its time either on or off (frozen), or twinkling madly. If too many genes are twinkling (chaotic dynamics) the cell will be unstable. Conversely, if too many genes are frozen the cell will be too stable for adaptive behaviour. Cells constantly need to adapt their gene expression pattern in response growth/differentiation factors, and to inter-cellular and other signals, then revert to their usual dynamics. A cell type is probably a set of closely related gene expression patterns, not just on the attractors, but shifting around within the basin of attraction, allowing an essential measure of flexibility in behaviour. Too much flexibility might allow a perturbation to flip the dynamics to a different basin of attraction, from a bone cell to a fat cell, or to something alien — a cancer cell.

The model indicates that evolution has arrived at a delicate balance between order and chaos — but leaning towards convergent flow and order [17]. The stability of attractors to perturbation can be analysed by the jump-graph (Fig. 15) which shows the probability of jumping between basins due to single bit-flips to attractor states [18]. 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 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 [17].

[2] Chaotic attractors are hard to find with this method because transients and attractors become too long to be identified.

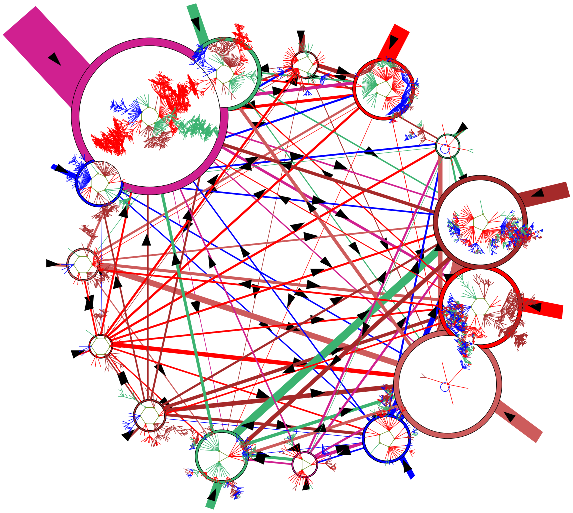

Fig. 15. The jump-graph (of the same RBN as in Fig. 2) shows the probability of jumping between basins due to single bit-flips to attractor states. Nodes representing basins are scaled according the number of states in the basin (basin volume). Links are scaled according to both basin volume and the 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 each node.

5. Conclusion

This paper has reviewed a variety of discrete dynamical networks where knowledge of the structure of their basins of attraction provides novel 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 steam 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 `www.cogs.susx.ac.uk/users/andywu/publications.html`

[1] A. Wuensche, 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.

[2] A. Wuensche, *The Ghost in the Machine: Basin of Attraction Fields of Random Boolean Networks*, In: *Artificial Life III*, ed. C.G. Langton, Addison-Wesley, Reading, MA, 1994, p. 496.

[3] S.A. Kauffman, *The Origins of Order*, Oxford University Press, 1993.

[4] A. Wuensche, Discrete Dynamics Lab (DDLab), software for investigating discrete dynamical networks, 1993–2009, `www.ddlab.org`

[5] C.G. Langton, *Physica D* **42**, 12 (1990).

[6] S.A. Kauffman, *Theoretical Biology* **22**, 439 (1969).

[7] A. Wuensche, *Complexity* **4**, 47 (1999).

[8] A. Wuensche, A. Adamatzky, *Int. J. Mod. Phys.* **C17**, 1009 (2006).

[9] S.A. Kauffman, *Investigations*, Oxford University Press, 2000.

[10] A. Wuensche, Complexity in 1D Cellular Automata; Gliders, Basins of Attraction and the Z Parameter, Santa Fe Institute Working Paper 94-04-025, 1994.

[11] A. Wuensche, *Parallel Processing Letters* **19**, 283 (2009).

[12] A. Wuensche, *The Emergence of Memory: Categorisation Far from Equilibrium, in Towards a Science of Consciousness: The First Tuscon Discussions and Debates*, eds. S.R. Hameroff, A.W. Kaszniak, A.C. Scott, MIT Press, Cambridge, MA, 1996, p. 383.

[13] J.J. Hopield, *PNAS* **79**, 2554 (1982).

[14] A. Wuensche, Attractor Basins of Discrete Networks: Implications on Self-organisation and Memory, Cognitive Science Research Paper 461, DPhil Thesis, University of Sussex, 1997.

[15] R. Somogyi, C.A. Sniegoski, *Complexity* **1**, 45 (1996).

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

[17] S.E. Harris, B.K. Sawhill, A. Wuensche, S.A. Kauffman, *Complexity* **7**, 23 (2002).

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