Discrete dynamical networks, basins of attraction, and content addressable memory

Andy Wuensche
Discrete Dynamics Lab (www.ddlab.org)
and Faculty of Computing, Engineering and Mathematical Sciences, UWE

A key notion underlying the collective behaviour of discrete dynamical networks is that state-space is organized into a number of basins of attraction, connecting states according to their transitions, and summing up the network's global dynamics[4, 5].

Discrete dynamical networks consist of a set of elements (cells) taking inputs from each other, and changing each cell-state according to some logical function of its inputs. The connectivity is usually sparse and cells are updated synchronously in discrete time-steps. Examples of discrete dynamical networks are cellular automata, and the more general case, random Boolean networks, where the Boolean attribute may be extended to multi-value.

In these finite, deterministic (though unpredictable) dynamical systems, an initial state sets off a train of successor states (a trajectory) by the iteration of the logical functions relating to each cell's inputs. Any trajectory must encounter a repeat state (because state-space is finite); this defines a state cycle (an attractor). So the dynamics organizes state-space into transient states flowing to attractor cycles (basins of attraction) analogous to the same concept in continuous dynamical systems, but with the important difference that in discrete dynamical networks transients can merge outside the attractor. Basins of attraction therefore consist of transient trees rooted on attractor cycles, where the leaves of the trees are unreachable states that can only be introduced from outside the system.

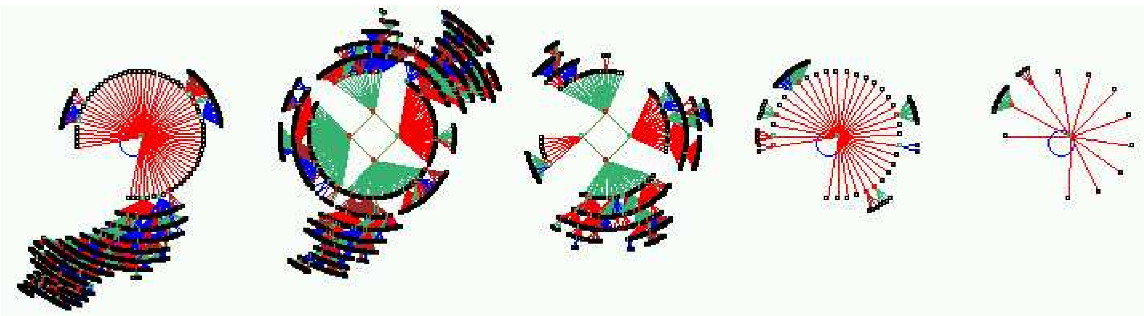

Figure 1: The basin of attraction field (state transition graph) of a small random Boolean network, with 13 cells. Cells have between 1 and 5 inputs, which were randomly assigned together with their logical functions. State-space (2^{13}=8192) is partitioned into 5 basins of attraction, size: 3998, 3294, 774, 98 and 29, with attractor period: 1, 4, 4, 1, 1. The density of leaves (unreachable states) is 0.992. Each node represents a different state in state-space. Time flows inwards to the attractor and then clockwise.

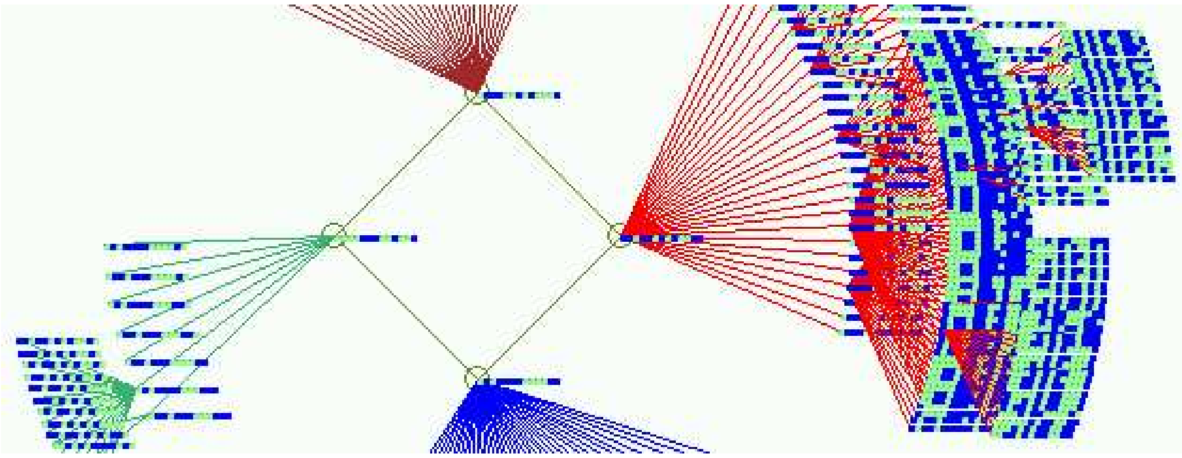

Figure 2: A detail of the third basin in figure 1, showing the states as bit-strings.

Subtrees and basins of attraction (represented as state transition graphs as in figures 1 and 2) can be computed with algorithms which directly generate the predecessors of each state[4, 5], driving the dynamics backwards in time along every valid reverse path. The exact graph depends on the exact connections and logic for each cell in the network. The state transition graph is to some extent stable to small perturbations/mutations in the network, but even the most minimal mutations (if they hit a “sensitive spot”) may drastically alter the graph. In general mutations to connections have more effect than mutations to the logic.

Figure 1 shows all the basins of attraction (the basin of attraction field) for a small random Boolean network. Figure 2 shows a detail of the third basin. Time flows inwards to the attractor and then clockwise. Note that the degree of convergence in the flow relates to notions of order-chaos in space-time patterns, high convergence implies order, low convergence implies chaos. Asimple measure of convergence is the leaf density of unreachable states.

How does this relate to memory? A network has “content addressable” memory from its ability to categorize (thus recognize) input, by following the resulting flow to particular attractors, Hopfield’s classic idea[1]. In discrete dynamical networks, there is also categorization outside attractors making a hierarchy of sub-categories far from equilibrium, because the root of each subtree is a sub-category, a richer system of content addressable memory than attractors alone[6]. These ideas provide an approach to understanding memory in networks of neurons in the brain[5, 6], and in genetic regulatory networks where basins of attraction are said to correspond to cell types[2, 3, 7, 8].

Work in progress addresses learning, how to change the network architecture to make desired categories and sub-categories, and the related “inverse problem” in genetics, how to infer the network that results in observed patterns of gene expression, cell types, in the organism.

This research is carried out with the help of the software Discrete Dynamics Lab (DDLab), available at `www.ddlab.org`[9]. Papers by the author in the references are also available at this site.

References

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

[2] Kauffman,S.A, *"The Origins of Order"*, Oxford University Press, New York, 1993.

[3] Somogyi,R., and C.Sniegoski, "Modeling the Complexity of Genetic Networks", COMPLEXITY, Vol.1/No.6, 45-63, 1996.

[4] Wuensche,A., and M.J.Lesser. *"The Global Dynamics of Cellular Automata"*, Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley, Reading, MA, 1992.

[5] Wuensche,A., "The Ghost in the Machine; Basin of Attraction Fields of Random Boolean Networks", in *"Artificial Life III"*, ed C.G.Langton, Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley, Reading, MA, 1994.

[6] Wuensche,A., "The Emergence of Memory", in *"Towards a Science of Consciousness"*, eds. S.R.Hameroff, A.W.Kaszniak, A.C.Scott, MIT Press, 1996.

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

[8] 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.

[9] Wuensche,A., "Discrete Dynamics Lab: Tools for investigating cellular automata and discrete dynamical networks", (updated for multi-value). To appear in *"Artificial Life Models in Software"*, eds. A.Adamatzky and M.Komosinski, Springer, 2004.