View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by The University of Utah: J. Willard Marriott Digital Library

Mapping Bateson's Epistemology to a Boolean Dynamic System:

1. The Emergence of Dynamic Form & 2. Hierarchies Form

International Society for Systems Sciences
50th Annual Meeting
Sonoma State University
Rohnert Park, CA
July 10-14, 2006

Thomas Mally and Gary C. Jensen
University of Utah

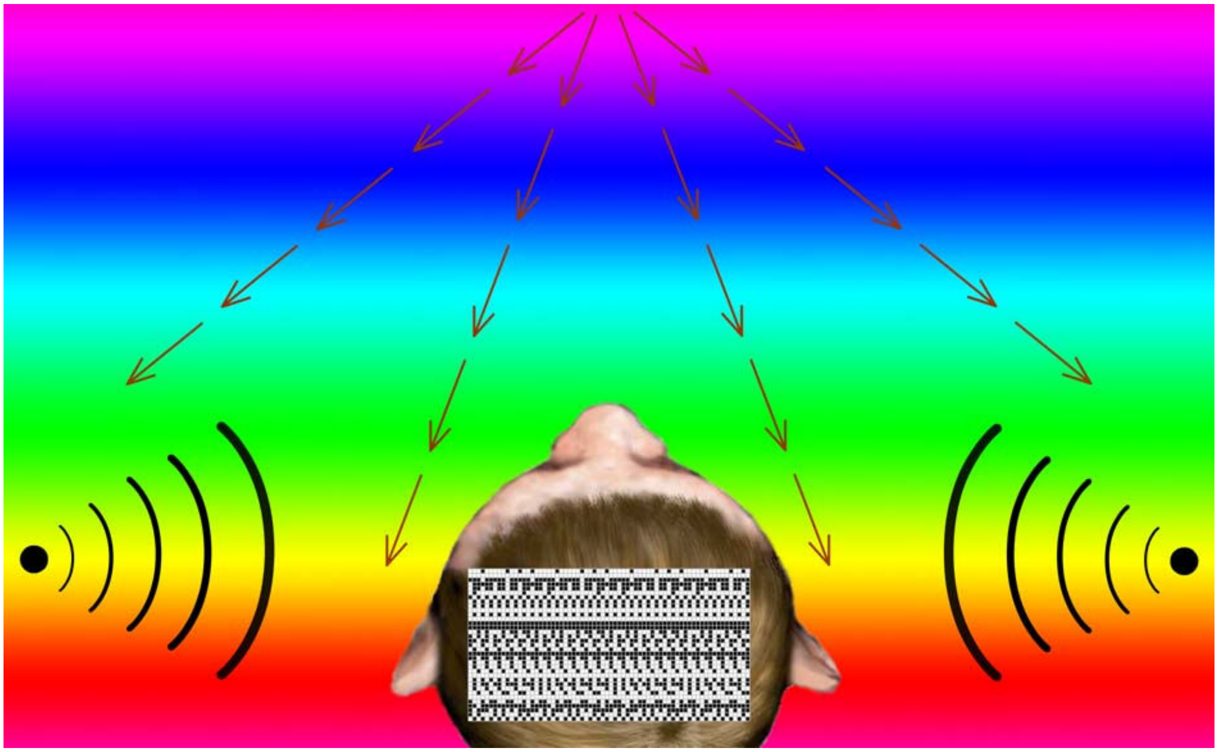
Gregory Bateson (e.g., 2002, p. 85ff; 2000, p. 457-460) construed knowledge to be the propogation of "difference" in a complex network, noting (2000 p. 460) that "the transform of difference travelling in a circuit is an elementary idea." The idea of difference is coded as 0 and 1 in Boolean systems. Kauffman (1993) developed NK Boolean computer simulations as a way explore how the structure of genomes might self-organize into emergent form (see Kauffman, 1995, p. 76 for a simple, concrete example). The N in NK Boolean systems refers to the number (N) of abstract entities called Nodes; in Kauffman's simulations N was very large, as high as 100,000 (1995, p. 83). The K refers to the number of inputs (from other nodes in the network) that each node has. Kauffman states that "While this is surely an idealization, we can extend it to networks of genes and their productrs interacting with one another in enormous webs of regulatory circuitry," (1995, p. 99). This reasoning parallels Bateson's idea of mental process. Indeed Bateson includes genetic activity as a part of mental process, stating that "the phenomena we call thought, evolution, ecology, life, learning and the like occur only in sytems that satisfy these criteria," (2002, p. 86). Mind and nature are, as his book title (Bateson 2002) states, are a necessary unity.

Bateson: Criteria of Mind

(1) Mind is an aggregate of interacting parts or components.
(2) The interaction between parts of mind is triggered by difference.
(3) Mental process requires collateral energy.
(4) Mental process requires circular (or more complex) chains of determination.
(5) In mental process the effects of difference are to be regarded as transforms (i.e., coded versions) of the difference which preceded them.
Air pressure differences ==> Fluid Pressure Diffs ==> Cilia root pressure diffs ==> Neuronal State differences ==> etc.
(6) The description and classification of these processes of transformation discloses a hierarchy of logical types immanent in the phenomena.
Logical Hierarchies vs Emergent Hierarchies?

I shall argue that the phenomena which we call thought, evolution, ecology, life, learning occur only in systems that satisfy these criteria.
(Bateson, 1979, pp. 85, 86)

"the transform of difference travelling in a circuit is an elementary idea."

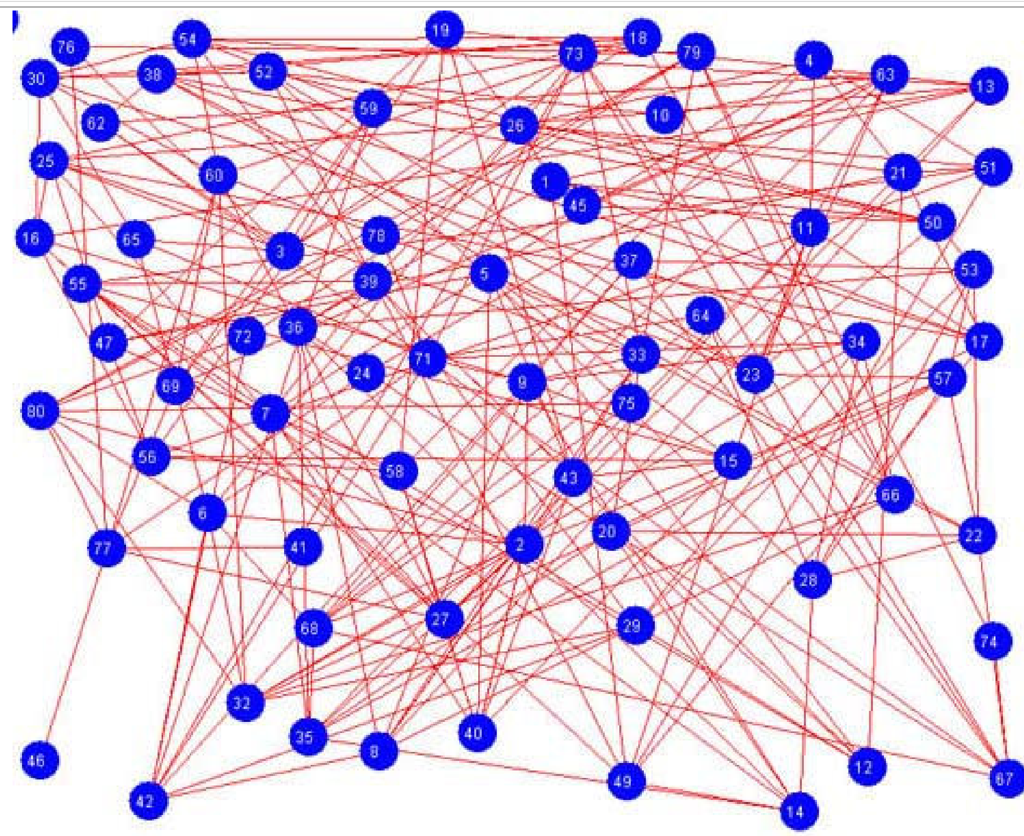


The idea of difference is coded as 0 and 1 in Boolean systems

Boolean Systems **My Tautology of Choice**
Kauffman: ***The Origins of Order*** **Mapping Boolean Systems onto Descriptions of Evolution**

Kauffman (1993) proposed that self-organization and natural selection are co-principles "weaving the tapestry of life."

Kauffman
developed Boolean network models as idealizations to explore questions about the creation of "genetic circuits of a variety of logics and complexities" (1995, pp. 98, 99)



Nodes. As a specific case for explaining how E42 works, consider an N, K Boolean system (e.g., Kauffman, 1993) that has N = 4 nodes and K = 2 inputs to each node. Name the four nodes, in order, A, B, C, D. Each node has two discrete states, 0 or 1. The state (0 or 1) of a node at time T+1 is determined by the relation between its two inputs at time T. The discrete time change from T to T+1 is called an **iteration** of the system.

Wiring. Wiring refers to how the four nodes are connected to each other--which nodes give input to which other nodes. The wiring of the 4-Node Standard system is arbitrary; its purpose is to create a simple example. Figure 1 shows the (arbitrary) connections among the four nodes of 4-Node Standard. Arrows originate in nodes that are sending input and end in nodes receiving input . Notice that node A receives input from nodes C and D, as does node B. Similarly, nodes C and D both receive input from nodes A and B. Thus, there are feedback loops between A and C, between A and D, between B and C and between B and D. A and B (as well as C and D) are not directly connected; any influence each has on the other is indirect. The same is true for C and D.

(More formally the nodes can be construed as vertices and the connections as edges.) top

Logical operators. The system is relational. The relation between a node's K=2 inputs at time T will determine that node's future state at time T+1. In Table 1 a "0" means OFF and a "1" means ON. Table 1 shows the (arbitrarily chosen) relations that determine the T+1 state of each of the four nodes in our 4-Node Standard. While the nature of these relations can be quite general (e.g., Barabasi, 2002), in N, K systems they are typically standard logical relations. Notice in Table 1 that the state of node A at time T+1 is determined by the logical OR operator between inputs from nodes C and D at time T. Similarly inputs to B are related by the logical exclusive OR (XOR), and two inputs to C and to D are related by AND and OR respectively. The set of operators used here are arbitrary except that, after working several examples by hand, this set was found to generate and example that has characteristics that are useful for learning the main points of discrete dynamic systems.

Table 1. Logical relations among nodes in the "4-Node Standard" discrete dynamic system

Node A			Node B			Node C			Node D		
OR			XOR			AND			OR		
C at T	D at T	A at T+1	C at T	D at T	B at T+1	A at T	B at T	C T+1	A at T	B at T	D T +1
0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	1	1	0	1	0	0	1	1
1	0	1	1	0	1	1	0	0	1	0	1
1	1	1	1	1	0	1	1	1	1	1	1

State Vectors. To keep track of the changing states for all four nodes we define a state vector. At time T, the state vector, **S**(T), is defined such that the first position in the vector represents the state of A, the second position the state of B, and so on.

In this way the expression **S**(1) = {1100}
means that, at time T=1,

A = 1, B = 1, C = 0, and D = 0.

At any point in time, T, the state of the system (the combination of all four states of all four nodes) is fully described by **S**(T). For any different point in time, T**"**, the state of the system (that is the states of all of its node) would be describe by **S**(T**"**).

State Space. The state space is a matrix of all possible state vectors that can occur. In this simple system, we won't write out that matrix, but every possible state vector that can occur will be shown in the leftmost four columns of Table 2, below. top

Parameters of the System. The system is specified by the number of nodes, **N**, and the number of connections per node, **K**. It is also critically specified by the **wiring diagram** (Figure 1). It is further specified by the **truth tables** (logical relations) which determine the future state of each node in relation to the K other nodes it receives input from. Later, for more advanced systems will introduce other parameters like whether a node is self-referencing (that is, checks its own state) or not.

Variables of the System. Once a system is specified, the major variables of interest are the **state vectors**, S(T) and the index indicating which **iteration** (T) the system currently resides in. top

Table 2. State Transition Table for the "4-Node Standard" system

T				T + 1			
A	**B**	**C**	**D**	**A**	**B**	**C**	**D**
0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0
0	0	1	0	1	1	0	0
0	0	1	1	1	0	0	0
0	1	0	0	0	0	0	1
0	1	0	1	1	1	0	1
0	1	1	0	1	1	0	1
0	1	1	1	1	0	0	1
1	0	0	0	0	0	0	1
1	0	0	1	1	1	0	1
1	0	1	0	1	1	0	1
1	0	1	1	1	0	0	1
1	1	0	0	0	0	1	1

We have now specified the 4-Node Standard system completely. Start the system in any combination of states (i.e., in any state vector) and all future states will be fully determined.

Deterministic State Transitions. To examine this flow of deterministic causality look at Table 2 which lists the state transitions. The left four columns (gray) define the full state space, that is, for any given time T, they list every possible state vector from {0000} to {1111}. In this simple system, there are no other combinations of states

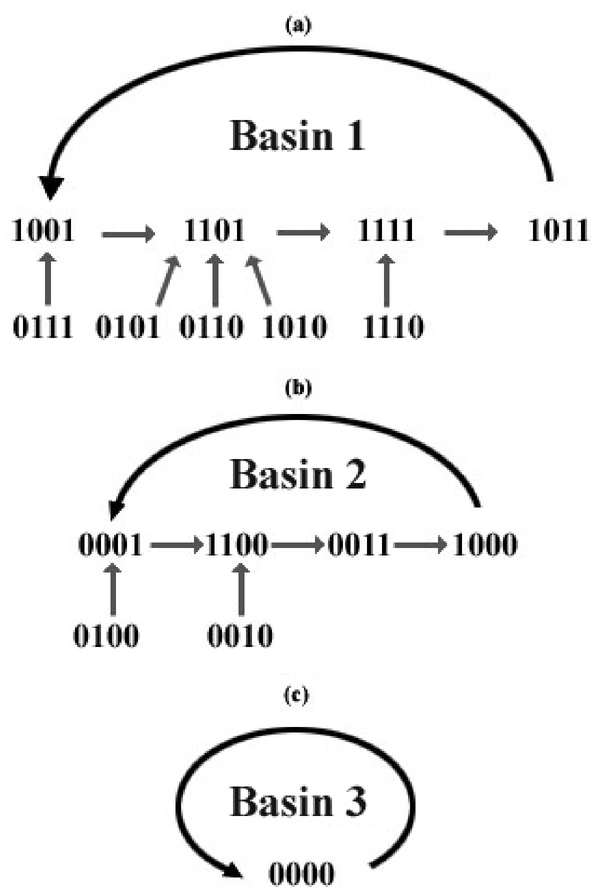
1	1	0	1	1	1	1	1
1	1	1	0	1	1	1	1
1	1	1	1	1	0	1	1

for the four nodes than those listed in the left four (gray) columns.

Pop-up Table of Logical Relations among the four nodes to help you compute the state transitions

For each row in Table 2, the state vector in the left-hand four columns at time T deterministically flows to the corresponding state vector in the right-hand four columns at T+1.
For example, Table 2 (row 9)shows that if at T the system is in state vector {1000}, where only node A is in state 1, then at T+1 the system will go to {0001} where only node D is in state 1. This transition can be derived using the logical relations in Table 1.
[The derivation might seem obvious to you; in case it is not, I will go through the steps: Given {1000} at T, at T+1 node A will change to 0 (since nodes C and D are both 0 at T). On the other hand, node D will change from state 0 at T to state 1 at T+1 because D takes on state 1 if either A or B or both are a 1, which is the case at time T.]

By similar reasoning, nodes B and C do not change states. Other state transitions in Table 2 are left to the inspection of the reader. top



Attractor Basins. Now we are in a position to describe the behavior of the system, which is characterized by three attractor basins. Examine the first line of Table 2, and note that if the 4-Node Standard system starts in state vector {0000} at T, then it remain in state {0000} at T+1, which means that at T+2 it must remain in {0000} and so on forever. The system is in an attractor basin which we will call Basin 3. The Attractor for Basin 3 has period of one: that is, it is a fixed point attractor. We also so that Basin 3 has length L=1. Basin 3 can be written as: [{0000} to repeats {0000} etc….]

Even this simple system has basins of greater complexity than Basin 3. As already noted, if the system at T is in state vector {1000} then at T+1 it moves to {0001}. Where does it go after that? Looking up {0001} in Table 2 (second row), it moves to {1100} on the next iteration, which itself transforms to {0011} on the iteration after that. But {0011} transforms into {1000} which is where we started. So the system has fallen into another basin, 2, that at has period four or length L=4, i.e., it repeats any state vector every four iterations. The Attractor Cycle of Basin 2 can be written as: [{1000} to {0001} to {1100} to {0011} to repeats {1000}etc…]. A third basin, Basin 1, can be found in Table 2 and is shown in Figure 2(a). top

Tributaries and Attractors. Figure 2 (a, b, c) illustrates the three basins of the 4-Node Standard system. Notice that the two period four basins (2 and 1) both have one or more tributaries. A tributary (transient) is a state vector that leads directly (or indirectly through other tributaries) into a basin. A system passes through a tributary state vector only once on its way to a basin. Once in a basin, the system's behavior cycles through the same set of state vectors.

Perturbations. Once in a basin, the system will remain there unless it is perturbed. Perturbation consists of changing one or more states in a state vector. For example if the system is in Basin 3 and something external to the system changes node A from 0 to 1, the resulting state vector {1000} means that the system has been perturbed into basin 2. Now it will cycle endlessly through the four state vectors of 2.

Similarly, if the system is in basin 2 at {1000}and at something perturbs node C from 0 to 1, the resulting state vector {1010} is a tributary of basin 1 (see Figure 2 c). On the next iteration, T+1, {1010} will transform into {1111} which will then transform at T+2, into {1011}, at T+3 into {1001}, at T+4 into {1101} and finally at T+5 back into {1111}.

E42 must be perturbed by an external influence, that is, the user. Presumablycomplex living systems can be perturbed by external influences and have the ability to perturb themselves.

$$\textbf{Basin(1)=} \left\{ \begin{array}{c} \textbf{1001} \\ \textbf{1101} \\ \textbf{1111} \\ \textbf{1011} \end{array} \right\}$$

Un-Transposed Attractor 1 Matrix

Nodes ==>

1	0	0	1
1	1	0	1
1	1	1	1
1	0	1	1

Transposed Attractor 1 Matrix

Time ==>

1	1	1	1
0	1	1	0
0	0	1	1
1	1	1	1

2. Replace 1's with BLACK and 0's with White

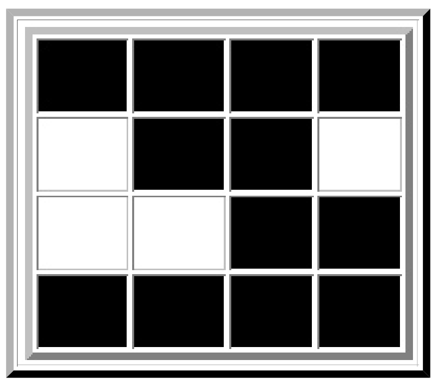
Transposed Attractor 1 Matrix

Time ==>

Node A	1	1	1	1
Node B	0	1	1	0
Node C	0	0	1	1
Node D	1	1	1	1

Transposed Attractor 1 Matrix

Time ==>



Self-organization, Emergence, and the Origins of Order. In Figure 2 we see hints of the kind of self-organization that Kauffman proposed as the organization of order in evolution. we will discuss the idea of emergence elsewhere.

This is pretty much what Bateson meant by: " (6) The description and classification of these processes of

transformation discloses a hierarchy of logical types immanent in the phenomena."

link to Apparent Motion

Link to Hiearchies

Two logics		
Logic of Logic		**Logic of Dreams**
All Humans die		**All Humans die**
***I* am Human**		**All Life dies**
***I* will die**		***I* am Life**

(Scientific) Explanation **A Special Case of Double Description**
Explanation: the ***mapping*** of **Tautology** onto **Description**
A tautology in its simplest form is 'If P is true, then P is true' (1979, p. 78)

Tautologies can be very elaborate including, for example, Nonlinear Dynamical Systems Models
Description and **Tautology** constitute, for Bateson, a particularly potent pair of independent languages for generating knowledge.
This is a form of logic of dreams: An identity of relations **Computer Science ==> Human Information Processing**
In Science **We build spider-web bridges** **across the chasm between** **a Tautology and a Description**