

Mapping Knowledge to Boolean Dynamic Systems in Bateson's Epistemology

Thomas E. Malloy, Department of Psychology, University of Utah

Gary C. Jensen, Department of Psychology, University of Utah

Timothy Song, School of Computing, University of Utah

Correspondence should be directed to: Thomas E. Malloy, University of Utah, Department of Psychology, 380 S. 1530 E., Room 502, Salt Lake City, UT 84112-025; email: malloy@psych.utah.edu.

RUNNING HEAD: Mapping Knowledge to Boolean Systems

The citation for this article is:

Malloy, T. E., Jensen, G. C., & Song, T. (2005). Mapping knowledge to Boolean dynamic systems in Bateson's epistemology. *Nonlinear Dynamics, Psychology, and Life Sciences*, 9, 37-60.

Abstract: Gregory Bateson (1972, 1979) established an epistemology that integrates mind and nature as a necessary unity, a unity in which learning and evolution share fundamental principles and in which criteria for mental process are explicitly specified. E42 is a suite of freely available Java applets that constitute an online research lab for creating and interacting with simulations of the Boolean systems developed by Kauffman (1993) in his study of evolution where he proposed that self-organization and natural selection are co-principles "weaving the tapestry of life." This paper maps Boolean systems, developed in the study of evolution, onto Bateson's epistemology in general and onto his criteria of mental process in particular.

KEYWORDS: Epistemology; Bateson; knowledge; Boolean networks; mental process; logical operators; emergence; description; tautology; explanation; representation.

E42 is a program for constructing, editing, representing, and finding the characteristics of Boolean systems. Kauffman (1995, p. 99) developed Boolean network models (which are distinct from the related idea of NK fitness landscapes, e.g., 1995, p. 171 ff) to explore questions about the creation of “genetic circuits of a variety of logics and complexities” (p. 98). He argues that such networks are useful idealizations for studying biological processes (1993, pp. 174, 182, 183) and that attractors in dynamic systems, which are susceptible to systematic study in “ensembles of Boolean networks,” are pervasive in “genomic cybernetic systems... immune systems, neural networks, organ systems, communities and ecosystems.” While his thinking extends well beyond the boundaries of Boolean systems, he uses them as a critical tool for exploring important theoretical proposals, “I shall suggest that there are general principles characterizing complex systems able to adapt: They achieve a ‘poised’ state near the boundary between order and chaos, a state which optimizes the complexity of tasks the systems can perform and simultaneously optimizes evolvability,” (1993, p. 173).

FUNDAMENTALS OF BATESONIAN EPISTEMOLOGY

Differences that Make a Difference

A primary focus of Bateson’s work is a sophisticated and highly elaborated epistemology based on difference. Using Korzybski’s (1958) map (knowledge) and territory (what is to be known) metaphor, he argues that what gets onto maps from the territory are differences, whether these be “a difference in altitude, a difference in vegetation, a difference in population structure, difference in surface, or whatever,” (e.g., Bateson, 1972, p. 457). Differences are “elementary ideas,” (1972, p. 315, 463), or to be more complete, “the transform of a difference traveling in a circuit is an elementary idea” (1972, p. 460). Difference, which language so facilely converts to a noun, is more properly a relational process whose outcome can be represented as the binary distinction, e.g., bits of information or the true/false designations of a logical truth table. Only those differences in the territory which make a difference in the neurology of the knower qualify as knowledge. A dog whistle, for example, produces air pressure differences that make a difference in the auditory system of a dog but not in the auditory system of a human. Knowing is a relational process which requires that differences in the world make differences in the knower (1972, p. 272).

Multiple Descriptions

A fundamental premise in Bateson’s epistemology is that knowledge is generated from integrating multiple versions or multiple descriptions of the world: “What bonus or increment of knowing follows from combining two sources of information?” (Bateson, 1979, p. 63; note that numbers for *Mind and Nature* correspond to the 2002 Hampton Press edition). His use of the term “description” in this context is quite general; it can refer to any form of knowing, a visual image, an auditory system response, the method section of a scientific paper, and so on. He argues that having multiple descriptions that apply to the world is a crucial tool for generating knowledge about the world. Whenever two independent sources are combined (in a principled way) new knowledge is generated; indeed, we think that Bateson meant this new knowledge as radically new in the sense that Goldstein (2002) uses “radical novelty” as a criterion for emergent phenomena. Thus we will hazard that, in modern terms, Bateson meant that knowledge emerges from combining different descriptions of the territory. For example, combining the slightly different images available in the right and left eyes allows the direct

perception of depth; depth is not directly experienced by one eye alone (although there are monocular depth cues).

Mental Process

Bateson (1979, pp. 89, 102, 106) characterizes mental process as the transformation of differences as they pass through a network. More specifically, he lists six criteria of mental process:

(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. (6) The description and classification of these processes of transformation discloses a hierarchy of logical types immanent in the phenomena.

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)

As the last part of the quote indicates, Bateson considers both evolution and thinking to be mental processes which share characteristics that make them deeply parallel. Both Bateson's characterization of difference as the elementary mental process and his proposing that "both evolution and learning must fit the same formal regularities" (1979, p. 4) suggest an intriguing relationship to Kauffman's work in general and to Boolean systems specifically as a way for further exploration of his epistemology. The basis of Bateson's epistemology in the detection of difference and in the transmission and transformation of difference in a network readily allows correspondences to be made with such flows of difference in networks generated by Boolean models. Moreover, the modern proposal that, in a theory of evolution, self-organization has a co-role with natural selection in the molding of life, particularly as this proposal is specified with Boolean models (e.g., Kauffman, 1993, p. 182), has potential implications for an extension of Bateson's theory that knowledge and evolution are parallel systems so that the self-organization of knowledge from experience can have a co-role with trial and error learning (1979, p. 274, 300, 301) in his description of the levels of learning.

FUNDAMENTALS OF BOOLEAN NETWORKS

In general terms, E42 allows an investigator to build relatively small discrete dynamic systems consisting of N ($4 \leq N \leq 400$) binary nodes (0, 1). On any iteration, each node accepts input (either 0 or 1) from K ($2 \leq K \leq 5$) other nodes in the system. Each node has a logical truth table which determines what its value will be on iteration $T+1$ as a function of the inputs it has received on iteration T . Other major structural variables that determine the dynamics of systems created in E42 are the connections (wiring) that determine which nodes give input to each node, the truth tables each node uses to decide, based on its inputs, if it will be on or off on the next iteration, and the initial state (on or off) of each node when the system starts on iteration $T=1$. While N and K must be user-specified to generate a discrete dynamic system, the wiring, truth tables, and initial states are generated pseudo-randomly for each node but can be afterwards altered. E42 has other user-defined control variables that will not be discussed in this paper. Next we will turn to a summary of the basic logic of how an NK Boolean system works.

Nodes

As a specific case for explaining how E42 works, consider a didactic example, 4-Node Standard, which is an NK Boolean system that has N=4 nodes and K=2 inputs to each node (see Fig. 1). 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.

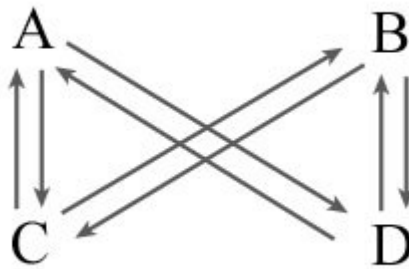


Fig. 1. 4-Node Standard: A simple NK Boolean dynamic system with N=4 nodes and K=2 connections per node. The arrows indicate direction of input among the four nodes

Wiring

Wiring refers to how the four nodes are connected to each other—which nodes give input to which other nodes. The wiring of 4-Node Standard 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 direct 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 and so even this tiny system meets Bateson’s criterion that there be complex chains of determination.

Table 1. Logical relations among nodes in the 4-Node Standard Boolean 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 at T+1 | A at T | B at T | D at 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 |

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. 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 by OR respectively. The operators used here are arbitrary.

State Vectors, State Space and State Transitions

To keep track of the changing states for all four nodes we define a state vector. At time T, the state vector, $\mathbf{S}(T)$, 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 $\mathbf{S}(1) = \{1100\}$ means that, at time T=1, A = 1, B = 1, C = 0, and D = 0. At any point in time, T_i , the state of the system (the combination of all four states of all four nodes) is fully described by $\mathbf{S}(T_i)$. For any different point in time, T_j , the state of the system (that is the states of all of its nodes) would be describe by $\mathbf{S}(T_j)$.

The state space is a matrix of all possible state vectors that can occur. Every possible state vector is shown in the leftmost four columns of Table 2. NK Boolean systems are deterministic; this flow of deterministic causality can be understood by examination of the transitions from state to state in Table 2. The left four columns define the full state space, that is, for any given time T, they list every possible state vector from {0000} to {1111}.

Table 2. State Transition Table for the 4-Node Standard system.
Leftmost four columns list all possible state vectors at time T.
Rightmost four columns constitute the state vector at time T+1
which deterministically flows from the prior state vector at time T.

| Node | Time T | | | | Time T + 1 | | | |
|------|--------|--------|--------|--------|------------|--------|--------|--------|
| | Node A | Node B | Node C | Node D | Node A | Node B | Node C | Node D |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

For each row in Table 2, the state vector in the leftmost four columns at time T deterministically flows to the corresponding state vector in the rightmost 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. 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.

The Flow of process

Notice that the transformation of differences immanent in state vectors at T and T+1 is a relational process. This is consistent with Bateson's formal view of knowledge in general:

... while I can know nothing about any individual thing by itself, I *can* know something about *relations between things*. If I say the table is "hard,"... what I know is that the... relationship between the table and some sense organ or instrument has a special character... It is always the relationship between things that is the referent of all valid propositions. It is a man-made notion that "hardness" is immanent in one end of a binary relationship. (Bateson & Bateson, 1987, pp. 157, 158.)

Tributaries, State Cycles and 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 which we will call basin 3 (the name is arbitrary). Since basin 3 cycles back to the same state vector on each iteration its cycle (or period) is said to have length one, $L=1$. Alternatively it can be said to be a fixed point attractor or described as having period one. Basin 3 simply moves from {0000} to {0000} to {0000}, and so on, in a single point loop as portrayed in Fig. 2 (c). 4-Node Standard 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 attractor basin, basin 2, that has period four or length $L=4$, i.e., it repeats any state vector every four iterations (panel (b) in Fig. 2). Basin 2 can be written as: [{1000}, {0001}, {1100}, {0011}, {1000}, etc...]. A final basin, basin 1, can be found using Table 2 and is explicated in Fig. 2 (a).

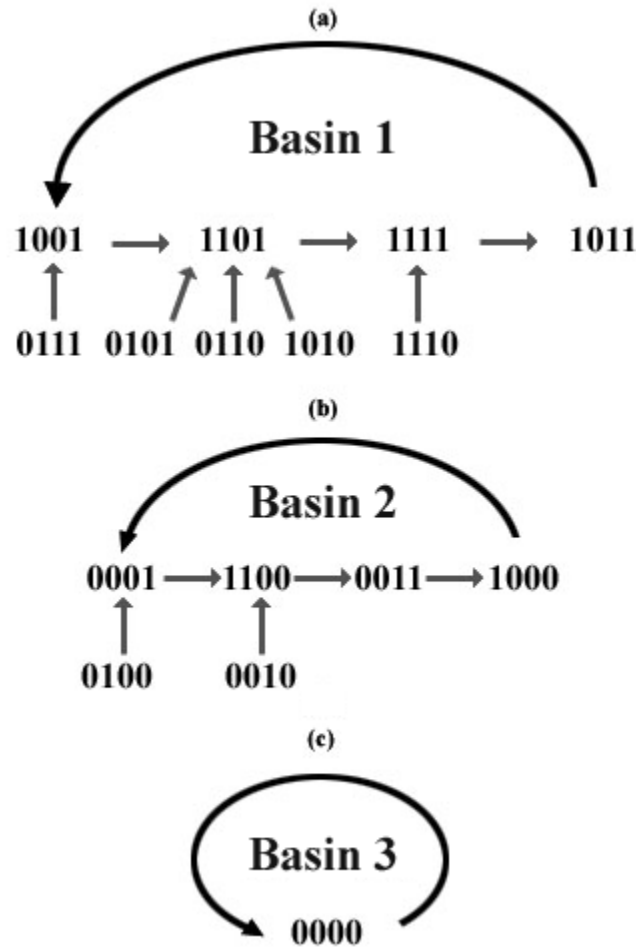


Fig. 2. Basin Structure of the 4-Node Standard example of an NK Boolean system.

If the same vector occurs twice, once at T and again at $T+L$, that means that the identical sequence of state vectors that fell between time T and time $T+L$ must repeat again between $T+L$ and $T+2L$ and yet again between $T+2L$ and $T+3L$ and so on forever or until the system is perturbed. The system is in a basin and L will be the basin length.

Figure 2 (a, b, c) illustrates the three basins of the 4-Node Standard system. Notice that the two period-four basins (basins 1 and 2) both have one or more tributaries. A tributary is a state vector that leads directly (or indirectly through other state vectors) into a basin. A system passes through a tributary state vector only once on its way to a state cycle (attractor). Once in an attractor, the system’s behavior cycles through the same set of state vectors endlessly. Strictly speaking the term “basin” is best used to refer to the looping state cycle (attractor) along with all the tributaries which lead into the attractor. But many people, the authors included, use “basin” to refer to the attractor loop when the meaning is clear.

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 (see Fig. 2). Now it will cycle endlessly through the four state vectors of basin 2. Similarly, if the system is in basin 2 at {1000} and something perturbs node C from 0 to 1, the resulting state vector {1010} is a tributary of basin 1, see Fig. 2 (a). 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 can be perturbed by an external influence, that is, by the user or it may perturb itself. In the discussion below we will describe how in searching for its own basin structure, E42 perturbs itself. Presumably, complex living systems both can be perturbed by external influences and can perturb themselves. Note also that there are more drastic ways of perturbing a Boolean system than simply changing the states of nodes. The truth tables or the wiring may be changed, even the number of inputs per node (K) and the number of nodes (N) can be changed. It becomes a definitional question as to whether the system is the same system after such changes. In any event, such changes typically affect the behavior of the system in profound ways which will not be considered in this paper.

A basin matrix (**B**) is the ordered list of all state vectors that occur in a basin's attractor cycle. For example, the basin matrix for basin 1 is [{1001}, {1101}, {1111}, {1011}]. Since a basin is a circle, cycling repeatedly through all its state vectors, there is no natural state vector to call the first state vector. We have therefore taken the convention that basin matrices are rotated until they start with (in row 1 of the basin matrix) the vector with the lowest Boolean value. There is no reason for that choice except that it is easy to calculate, unique, and every basin must have one such vector. Having a standardized starting point makes finding the same basin in an archive of basin matrices much easier. It also allows us to adjust phase transitions among basins for future analyses, such as those by the TAO tool described below.

Circular Chains of Determination

While state cycles, attractors, and basins are concepts that, as far as we know, are nowhere explicitly mentioned by Bateson, these concepts are intriguingly akin to the to Bateson's fourth criterion that mental process requires chains of determination that are at least circular in complexity. His work is replete with examples of circular causality such as his categorization of symmetrical, complementary, and reciprocal social relations as early as 1939 (reprinted in Bateson, 1972, see particularly pp. 68, 69), his characterization of the evolution of ecosystems (e.g., 1972, pp. 155, 338, 339), and his characterization of the structure of an ecological crisis (1972, p. 499).

XOR and the Qualities of Change over Time

Bateson's epistemology is based upon a living system's ability to detect difference. The ears of a dog can hear higher pitches than human ears; that is, their ears can detect differences in air pressure that ours cannot. We now turn to the importance of difference in detecting those dynamic patterns called basins. The logical XOR function can be said to detect difference in the sense that it returns a 1 if two inputs are different and a 0 if they are same (see Table 1). TAO is a suite of tools for describing and analyzing change over time (hence the name); its primary basis for doing so is the XOR operator.

In the following discussion, we will lay out a conceptual design of how basins are found; sometimes the actual computer code deviates from this conceptualization in minor ways for reasons of speed and efficiency. First, while the system is running in E42, the TAO tool creates a list of state vectors for a long series of iterations (the number of iterations in such a series is a variable specified by the user). Then, beginning with the second state vector, it compares each state vector backwards in time with all preceding vectors. When TAO detects that a state vector is *not* different from some prior state vector it "knows" that the system has fallen into an attractor cycle (Fig. 2 makes apparent why the repetition of a state vector means the system is looping in a cycle). This is the discrete analogue of looking for a first derivative equal to 0. So the first function of TAO is to detect basins in system dynamics. For each found basin TAO extracts a copy of the basin matrix (the list of state vectors in that basin). It then indexes (names) and archives the basin matrices for all found basins. After the search is completed TAO rotates all basins in the archive to begin with the state vector that has the lowest Boolean value. Creating an archive of basins, named and rotated, allows the dynamics of a specific basin to be found and studied on demand without the tedious, sometimes impossible, wait for the running system itself to repeat that basin. Once there is an archive of basins, the TAO tool suite has the capacity to examine what we call the higher-order derivatives of the dynamic flow (i.e., the changes in the changes over time). Finally, TAO has advanced features that categorize basins by their discrete derivatives; such categories have been found to correspond to hierarchies of perceptual groupings in humans that well may be a well-defined case of emergence in perceptual processes (Malloy, Bostic St Clair, & Grinder, in press).

TAO-1: The Discrete First Derivative

The discrete analogue of the first derivative uses the XOR operator to compare successive state vectors. Suppose, in the 4-Node Standard system, two successive state vectors are $\underline{S}(T) = \{0101\}$ and $\underline{S}(T+1) = \{1101\}$. Using XOR to compare each corresponding node in the two vectors yields TAO-1 = {1000} because the first node is different in the two state vectors and the last three are the same. Table 3 extends this idea to a series of state vectors for six iterations from the flow of changes in the 4-Node Standard system. It is relatively easy to confirm the computation of TAO-1 by examining the rightmost column in Table 3. Simply compare the corresponding nodes in two consecutive state vectors. If they are different TAO will return a 1; if they are the same it returns a 0. Consider the functional notation, e.g., TAO-1 (4, 3), from the third row of the table. The "1" in TAO-1 means refers to the first derivative (higher order derivatives will not be discussed in this paper). The parentheses (4, 3) means we are comparing the state vectors for T=4 and T=3. Finally, of course, TAO is the symbol of the function that returns a 0 when there is no change in the elements of state vectors over time and a 1 where there is. The elaborate indexing (4, 3) of iterations usually is not necessary, but in the discussion below of how TAO-1 is used to discover basin structure, it will be useful.

Table 3. TAO-1 applied to a sequence of state vectors from the dynamic flow of the 4-Node Standard system. TAO-1 compares the state of corresponding nodes at T and at T+1 and returns a **0** if the states of a node are the **same** at T and T+1 or returns a **1** if the states of a node are **different** at T and T+1

| Time (T) | State Vector at time (T) | First Derivative (TAO-1) |
|------------|--------------------------|---|
| T=1 | S(1) = {0101} | Non-applicable |
| T=2 | S(2) = {1101} | $\underline{\mathbf{S}}(1) = \{0101\}$ $\underline{\mathbf{S}}(2) = \{1101\}$ TAO-1 (2,1) = {1000} |
| T=3 | S(3) = {1111} | $\underline{\mathbf{S}}(2) = \{1101\}$ $\underline{\mathbf{S}}(3) = \{1111\}$ TAO-1 (3,2) = {0010} |
| T=4 | S(4) = {1011} | $\underline{\mathbf{S}}(3) = \{1111\}$ $\underline{\mathbf{S}}(4) = \{1011\}$ TAO-1 (4,3) = {0100} |
| T=5 | S(5) = {1001} | $\underline{\mathbf{S}}(4) = \{1011\}$ $\underline{\mathbf{S}}(5) = \{1001\}$ TAO-1 (5,4) = {0010} |
| T=6 | S(6) = {1101} | $\underline{\mathbf{S}}(5) = \{1001\}$ $\underline{\mathbf{S}}(6) = \{1101\}$ TAO-1 (6,5) = {0100} |

Using Backward Comparisons to find Basins

The rational for finding basins analytically with TAO is simple—at any moment, T, compare the current state vector backwards in time not just with the previous vector but with *all* prior state vectors since the system started either until a comparison yields the zero vector, **0**, or until iteration T=1 is reached. A zero vector, of course, means that two state vectors are identical which means that they are the same vector and when a deterministic system repeats the same vector it is in a basin.

Let us examine this rational in more detail with all the computational details of the procedure. Table 4, below, expands the example started in Table 3. In the T-1 column we compare, as in Table 3, the current state vector, $\underline{\mathbf{S}}(T)$ with the state vector one iteration back in time, $\underline{\mathbf{S}}(T-1)$. In the T-2 column we compare the current state vector, $\underline{\mathbf{S}}(T)$ with the state vector from two iterations back in time, $\underline{\mathbf{S}}(T-2)$. And so on. Let the system begin running at T=1 in state vector $\underline{\mathbf{S}}(1) = \{0101\}$, which, from visual examination of Fig. 2, we know is a tributary that will lead into basin 1. Begin comparing backwards at T=2. The only prior time is T=1, so there is only one backward TAO comparison, which yields $\{1000\}$ which not **0**. Therefore move to the state vector for the next iteration, T=3, and make the TAO comparisons between the current state vector, $\underline{\mathbf{S}}(3)$, and the two previous state vectors, $\underline{\mathbf{S}}(2)$ and $\underline{\mathbf{S}}(1)$; those two comparisons are found in the T-1 and T-2 columns of Table 4. Neither of these produces a TAO = **0**. Therefore, move to T=4 and repeat all the TAO comparisons backwards starting with one iteration back (T-1), then moving to two iterations back (T-2) and finally to three iterations back (T-3).

Examination of the T-1, T-2, and T-3 columns of the T=4 row of Table 4 indicates that none of these yields $\underline{0}$. Move on to T=5 and then to T=6. At T=6, in the T-4 column of Table 4, note that the TAO comparison between $\underline{S}(6)$ and $\underline{S}(2)$ produces $\underline{0}$, that is, TAO-1 (6,2) = {0000} = $\underline{0}$. At this point TAO has found a basin. The difference in the iteration number (T=6 minus T=2) is 4 which is the basin length, L = 4.

Table 4. Finding Basin 1. At every iteration, T, use TAO to compare backwards to all previous iterations until you reach T=1 or until TAO yields $\underline{0}$

| | T - 1 Detect Basins with L=1 | T - 2 Detect Basins with L=2 | T - 3 Detect Basins with L=3 | T - 4 Detect Basins with L=4 |
|-----|--|--|--|--|
| T=2 | S(2) = {1101} S(1) = {0101} TAO-1 (2,1) = {1000} | | | |
| T=3 | S(3) = {1111} S(2) = {1101} TAO-1 (3,2) = {0010} | S(3) = {1111} S(1) = {0101} TAO-1 (3,1) = {1010} | | |
| T=4 | S(4) = {1011} S(3) = {1111} TAO-1 (4,3) = {0100} | S(4) = {1011} S(2) = {1101} TAO-1 (4,2) = {0110} | S(4) = {1011} S(1) = {0101} TAO-1 (4,1) = {1110} | |
| T=5 | S(5) = {1001} S(4) = {1011} TAO-1 (5,4) = {0010} | S(5) = {1001} S(3) = {1111} TAO-1 (5,3) = {0110} | S(5) = {1001} S(2) = {1101} TAO-1 (5,2) = {0100} | S(5) = {1001} S(1) = {0101} TAO-1 (5,1) = {1100} |
| T=6 | S(6) = {1101} S(5) = {1001} TAO-1 (6,5) = {0100} | S(6) = {1101} S(4) = {1011} TAO-1 (6,4) = {0110} | S(6) = {1101} S(3) = {1111} TAO-1 (6,3) = {0010} | S(6) = {1101} S(2) = {1101} TAO-1 (6,2) = {0000} |

Archiving and Finding the Next Basin

At this point the TAO tool will archive the basin. It will record the four state vectors that constitute the basin, rotate them so that the state vector with the lowest Boolean value is placed first, and name the basin matrix. Since it is the first basin found, it will be called basin 1. The number (e.g., "1") in the basin name has no numerical or counting function; it is simply a name, that is, it functions like the numbers on athletic jerseys which are simply a way of identifying players.

Once TAO has found and archived a basin, it moves on to another search for other basins. To do so, TAO examines the latest state vector, (e.g., $\underline{S}(6) = \{1101\}$) and perturbs the system by pseudo-randomly selecting 50 percent of the nodes and changing their states (0 to 1 or 1 to 0). TAO then begins the process again. Using the 4-Node Standard example of Fig. 2, suppose the pseudo-random change of $\underline{S}(6)$ changed $\underline{S}(6)$ from {1101} to {1100}. TAO then starts the next basin search with {1100}. Fig. 2, above, indicates that this starting point is in basin 2, so that is the basin that will be found and archived. If the next perturbation produces {0000} the system falls into basin 3 (see Fig. 2) and that basin will be found and archived.

Obviously, a pseudo-random perturbation of the state of a system will not conveniently perturb the system into the next possible basin. If after a given search, the pseudo-random perturbation provokes the system into a basin that has been found before, which TAO "knows" by checking its archives, TAO will not create a new basin but simply note that the prior basin has occurred again. It keeps track of the relative frequency of the occurrence of any basin it finds, which later allows users to gain a sense of that basin's ability to exhibit homeostatic return after

perturbation (Kauffman, 1993, pp. 208, 209). In our simple example, when a TAO search was performed 1000 times the relative frequencies for basins 1, 2 and 3 were .561, .376, and .063 respectively. These relative frequencies are consistent with dividing the number of state vectors that contribute to each basin by 16 which is the total number of vectors in the state space (see Fig. 2).

The [TAO interface](#) has two user defined parameters: The number of perturbations and the number of iterations per perturbation. The first of these specifies the number times the user is willing to perturb the system in a search for basins. The second parameter gives the maximum basin length (in number of iterations) for which the user is willing to search. The current, very simple and fully analyzed, example would make that seem superfluous, but for complex systems it may be necessary to perturb the system 1000 or 10,000 times to get a sense of its basin structure. Even then, after 10,000 perturbations, it may be more a sense that enough basins have been discovered to describe the behavior the system than it is a sense that all the basins have been found. Once TAO has completed a search and archived basins, the tool can be used to search again, perhaps with larger values of the user-defined parameters. Such multiple searches for basins will result in different numbers (names) being applied to the same basins because basins will be found in different pseudo-random orders; also, different searches don't necessarily find the same set of basins.

We introduce here the phrase pragmatic chaos. TAO may not find a \emptyset first derivative within the user-defined parameters. Perhaps the user has TAO search 10,000 times (that is make 10,000 perturbations) and for each search allows it to look for basins up to $L = 1000$ (i.e., 1000 iterations per perturbation) and under these conditions TAO does not find a single basin. Such a result in no way proves the system is in chaos; the very next search may produce a basin, or, alternatively, there may be many basins, but they have lengths greater than 1000. But at some point, the user may be willing to define the system for practical purposes to have no discoverable pattern in its behavior within the parameters that have been set. These parameters can be set very high; ten thousand searches for basins of length 1000 can take about five to fifteen minutes, depending on computer speed. If the user is willing to leave a computer running all night or over the weekend the parameters can be set to very large values. We refer to systems that exhibit no basins within our parameters as in pragmatic chaos. Since we are focusing on the nature of the human ability to apprehend pattern in the behavior of complex systems, and since the span of apprehension is relatively small, this is a useful epistemological category.

Knowing the Fundamental Patterns of Creatura

Loops (feedback, feedforward, etc.) are fundamental to the foundational premises of systems theory—the flow of process somehow returns to itself. In an E42 Boolean system there are loops at the level of wiring (see Fig. 1) whereby difference flows among the nodes. But there are other loops—the attractor cycles shown in Fig. 2. The distinction between these types of loops is critical and cannot be overstated. The attractor cycles in Fig. 2 emerge out of the wiring structure (Fig. 1) and relational structure (Table 1) of the system. The loops apparent in Figures 1 and 2 are at different levels of analysis. Our interpretation is that the loops in Fig. 2, which we call the behavior of the system, emerge from the structure to the system.

Bateson and Bateson (1987) make clear that difference, loops, and an immanent hierarchy of loops are fundamental to the mental process of living systems (Creatura):

When this recognition of difference was put together with the clear understanding that *Creatura* was organized into circular trains of organization, like those that had been described by cybernetics, and that it was organized in multiple levels of logical typing, I had a series of ideas all working together to enable me to think systematically about mental process... (Bateson & Bateson, 1987, p. 14)

An E42 Boolean system parallels the above description of the mental process of *Creatura*; it has a central role for difference and it exhibits both loops and an immanent hierarchy of loops.

In terms of epistemology, if loops are a fundamental characteristic of living systems, then a critical function of the process of knowing would be the ability to detect such loops (attractor cycles) in the behavior of natural systems. The TAO tool detects such cyclic patterns. And it does so in a manner that is consistent with Bateson's description of the mental process of living systems: It detects differences in the transforms of differences. Bateson proposes that such news of difference is the fundamental act of knowing and we have shown that through that act—detecting differences in differences—a system can “know” the most basic pattern of living systems—cyclic behavior. Malloy, Bostic St Clair and Grinder (in press) show that by further refinement of the detection of difference a hierarchy of cyclic pattern can be known by the model in ways that correspond to human knowledge.

REPRESENTING DYNAMIC RECURSIVE LOGIC

Logical Calculations and Proofs

As the discussion above of the 4-Node Standard example makes clear, the dynamic behavior of a Boolean system is the result of the sequential application of logical operations. To unfold the behavior of such a system requires, in essence, a logical proof. These systems use truth tables to generate their behavior, moving through tributaries into basins and thus the occurrence of a basin is essentially a proof. Conversely, observing the behavior of such a system is to comprehend, in some form, the steps of a logical proof as it unfolds. Comprehend in what form? How specifically are these logical steps to be perceived?

The motivation for E42 is epistemological: If we construe the input from the universe as a dynamic system, how do humans apprehend its dynamic patterns? Can they distinguish when a system is in an ordered basin or in a tributary or in chaos? Can they distinguish one basin from another basin? Can they specify the length of a basin or other characteristics that might matter to their ability to act effectively or even survive? Step for a moment into a metaphorical application; suppose we think of people as dynamic systems and their moods as one set of attractor basins that change with various perturbations. With a good friend we can perceive, nearly instantly, which mood (basin) s/he is in. How do we do that? We won't answer that question here, of course; we simply want to highlight that for human knowledge the question of how we perceive basins and other characteristics of dynamic systems is a crucial and interesting issue.

E42 eases the massive cognitive load and saves the time and effort of deriving complex proofs of the recursive, dynamic moves of a logic-based system. Even so, the results of the derivations are themselves complex. Tables of 0's and 1's, like Table 2 but hundreds of columns wide and thousands of rows deep, don't easily lead to insight or even comprehension, and so the question of representation is crucial. Representation will be considered in depth elsewhere in the future, but will be briefly addressed here because some knowledge of how the results of

logical derivations are represented is required to understanding the epistemological significance of the output.

One solution available in E42 to the question of representation is the Node Frame which represents the theoretical nodes as a matrix of dark and light green squares twinkling on and off as the logical calculations are made in real time. Interactive examples of systems represented by twinkling nodes can be found at www.psych.utah.edu/dynamic_systems. This closely follows the representational strategy of Kauffman (1993). Kauffman was searching for critical characteristics of a dynamic system such as ordered, chaotic, or edge of chaos regimes (e.g., 1995, p. 86), and representing his systems as nodes was useful for those purposes. But for other forms of knowledge, such as identifying individual basins among many basins, the twinkling nodes are less effective. For all basins greater than length one, the patterns in the twinkling nodes change on every iteration in complex ways, and they leave no history. So, for a small system with 25 nodes, distinguishing one basin of length 10 iterations from another of the same length means remembering ten sequential patterns (each pattern made up of 25 on-off elements) and being able to say that such a sequence of ten patterns is different from a second sequence of ten patterns. If there are many basins, remembering one from another is nearly impossible. Humans need some other way to do it.

Historical Trace: Smilie 3

The name, Smilie 3, is due historically to a cross language miscommunication from English to Chinese and back to English, and it is buried so deeply in the code that changing it would be heavy work; you can think of it as an arbitrary name. Smilie 3, a second option for representing dynamics, leaves a trace of the history of the state vectors through which the system has recently passed. It replaces a 0 with a white space and a 1 with a black space on a grey grid. In written text state vectors are naturally expressed as rows; in contrast Smilie 3 represents the state vectors as columns. That is, the states of a system's nodes constitute the vertical dimension of the 2-D grid. Time (iteration) constitutes the horizontal dimension.

Smilie 3 output for two basins from the 4-Node Standard system are shown in Fig. 3. Panels (a), (b), (e), and (f), all represent basin 1, but in slightly different ways while panels (c), (d), (g) and (h) represent basin 2. For now look at panel (a) which corresponds to the four state vectors for basin 1. The state vectors, rotated to start with the lowest Boolean value, are $\{1001\}$, $\{1101\}$, $\{1111\}$, $\{1011\}$. Unlike the vertical axis of a mathematical graph which runs from low values at the bottom of the axis to high values at the top, Smilie 3, in representing state vectors as columns, starts with the first node at the top of the column rather than at the bottom. In Fig. 3 (a) look below the grey line at the first (leftmost) column. Notice that, reading from the top, the grid squares are black-white-white-black; this corresponds to $\{1001\}$, but with black substituted for 1 and white substituted for 0. Moving rightward in panel (a), the second column is black-black-white-black, which corresponds to $\{1101\}$, the second state vector. The third column is black-black-black-black which corresponds to $\{1111\}$, the third state vector. Finally, the fourth column and the fourth state vector are black-white-black-black and $\{1011\}$, respectively. Panel (b) of Fig. 3 also represents basin 1 but the basin cycle starts on a different, arbitrary, iteration; in effect it is the same basin as panel (a) but out of phase. A comparison of panels (a) and (b) reveals that phase relations matter in perception; the visual patterns form

different gestalts. Fig. 3 panels (c) and (d) show a historical trace of basin 2 in similar manner. Again, for basin 2, phase relations matter in perception.

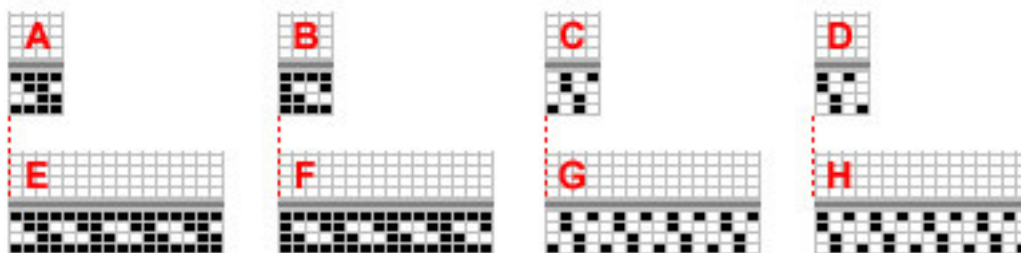


Fig. 3. Various visual representations of logical derivations showing two basins from the 4-Node Standard system. Node states are along the vertical axis. Iterations are along the horizontal axis.

We now turn to a perceptual phenomenon we call tiling effects. We frequently take snapshots of dynamics showing the historical trace of a basin passing four times through its cycle because of what we call the tile-floor effect. A single patterned floor tile often will not give insight to a large pattern until it is placed next to several other tiles. The same can be the case for historical traces. Figure 3 panels (e) and (f) show historical traces of basin 1 moving four times through its cycle. Notice that tiling the basins together strengthens the basin pattern and produces emergent gestalt perceptual characteristics. Figure 3, panels (g) and (h), show a similar tiling effect for basin 2. Returning to the metaphorical application, it may be useful, in a similar way, to observe a person cycle through a behavioral basin several times to get full effect of the pattern of the behavior. The Smilie 3 historical trace produces a visual representation of basin patterns that humans can easily use to recognize, distinguish, and identify; this is important when a large set of complex basins change repeatedly due to perturbations of the a dynamic system.

In application, representation involves complex issues (e.g., Bostic St Clair & Grinder, 2000, p. 164 ff). Here we have only raised some issues concerning the representation of the computer output itself and not any issues regarding how humans represent a dynamic universe. Indeed, we've just touched on computer representation; for example, E42 is able to represent system dynamics as auditory patterns and as complex patterns of visual movement.

MAPPING KNOWLEDGE TO BOOLEAN NETS

Description, Tautology and Explanation

We are here interested in the processes of knowing and in building useful ideas about the nature of knowing. Now that we have developed an explicit model toward that end, what sort of epistemologically well-defined procedures might guide its application? Bateson has offered such guidelines. As mentioned above, a fundamental process for generating new knowledge is the comparison and integration of multiple descriptions of the world: The experience of depth requires the integration of the double description offered by slightly different images in the right and left eye.

For Bateson (1979, p. 76 ff.), explanation is the mapping of description onto tautology. That is, explanation is in some abstract way like depth perception; it is a new form of knowledge that is generated by a principled mapping of a description of a phenomenon onto a tautology.

“An explanation has to provide something more than a description provides, and in the end, an explanation appeals to a *tautology*, which as I have defined it, is a body of propositions so linked together that the links *between the propositions* are necessarily valid. 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, the mathematics underlying nonlinear dynamics. Description and tautology constitute, for Bateson, a particularly potent pair of independent languages for generating knowledge.

Critical Issues in Mapping

The framing of explanation as the mapping between two distinct languages (description and tautology) brings up at least two important issues. Bateson argues that, in general, a description of any phenomenon is in no way predetermined (1979, p. 37 ff); that is, there is large number of descriptions that can be applied to any interesting phenomenon (say the processes of knowing). Furthermore, if the number of possible descriptions is large, so too is number of tautologies onto which a particular description may be mapped. All these different descriptions and tautologies have no prior claim over each other; it is left to we who explain and we who understand explanations to think critically about the choice of description and the choice of tautology. The second critical issue is the nature of the mapping between description and tautology. This, clearly, is nontrivial. We infer from the direct experience of the sense of depth that there is some sort of mapping between the images of the two eyes that produces this sense of depth. But there is currently no successful vision science model of the principles by which the brain maps the “description” from one eye to the “description” of other eye so that the perception of depth emerges; the correspondence problem (how to determine which features in the image of the left eye correspond to which features in the right eye) alone is presently insuperable (Palmer, 1999, p. 211). Most explanations of depth start with diagrams of rays initiating from a point in the environment and radiating at different angles to corresponding points in the two eyes; but the visual system is presented with the inverse problem: It cannot look down at itself in relation to the environment as typical diagrams do; it has only the two retinal images to work with and there is no *a priori* way of knowing which feature in one image corresponds to which feature in the other image. Put another way, typical explanations of depth beg the question of correspondence. This is a cautionary tale for any process of mapping from one realm to another realm and specifically for the mapping of description onto tautology. The critical issue we are describing is that the mapping from one description to another description in general can occur in many different ways, and discovering how to map in a way that is useful is nontrivial. In a chapter titled “Multiple Versions of the World” Bateson (1979) addresses these issues by offering nine cases, very different, of the kinds of mappings we are talking about here. His readers are left, presumably as Bateson himself was left, to generate principles for addressing these two issues (the choice of descriptions and the mapping between descriptions) on their own (see, Bostic St Clair & Grinder, 2000, p. 77 ff).

Mapping to Mental Process

Within the frame of explanation as a mapping between description and tautology, E42 is a computer program for creating and representing a large class of tautologies, each generated from different initial conditions by the logic of NK Boolean discrete dynamic systems. We have argued that Boolean systems are an appropriate tautology to put into relation with the kinds of

descriptions of mental process that Bateson has proposed, starting at the definitional level of the “parts” of a mental system and proceeding to his construal of difference: “the fact [is] that it takes at least two somethings to create a difference. ...there must be two entities (real or imagined) such that the difference between them can be immanent in their relationship” (1979, p. 64). The N abstract nodes in an NK Boolean system are the model’s place holders for these unspecified “somethings.” Moreover the interactions of these nodes are triggered by difference (Boolean inputs). Thus Boolean systems created in E42 correspond to the Bateson’s first two criteria of mental process. The third criterion, that mental systems store collateral energy will not be discussed here but is related in interesting ways to Kauffman’s (2000, p. 4, 98) idea that autonomous agents be able to propagate work. We’ve noted above the correspondence between E42-generated tautologies and the fourth, fifth and sixth criteria of mental process, which we have described as the fundamental patterns of living systems (Creatura). To review, Boolean networks require looping chains of determination (Criterion 4) and these loops exist at different levels of analysis (Criterion 6). Finally, E42 provides a way of tracking and representing the relational processes of taking differences and for transforming those differences as they circulate through a network (Criterion 5). Without the benefit of such computer simulations, Bateson used tautologies based on the logic of steam engine governors, doorbell circuits, and descriptions of simple ecological systems as ways of tracking the flow of the transformation of difference in a network. E42 allows the flexible construction of much more complex tautologies designed to fit epistemological descriptions of interest and to track the implications of such tautologies with more rigor than is possible with verbal examples.

OVERVIEW

The mapping we have offered between Bateson’s epistemological descriptions and Boolean tautologies is rather general at this point. Other (Malloy, Bostic St Clair & Grinder, in press) and future work will focus on more specific mappings from knowledge construed as the detection of difference and the subsequent flow of the transformations of those difference across a network onto the tautologies generated by Boolean networks. At the most general level, Kauffman’s argument that Boolean systems are a useful idealization of genetic processes maps well onto Bateson’s strong equivalence between mind and nature:

...epistemology is an indivisible, integrated meta-science whose subject matter is the world of evolution, embryology, and genetics—the science of the mind in the widest sense of the word... But epistemology is always and inevitably *personal*. The point of the probe is always in the heart of the explorer: What is *my* answer to the question of the nature of knowing? I surrender to the belief that my knowing is a small part of a wider integrated knowing that knits [together] the entire biosphere... (Bateson, 1979, pp. 81, 82).

References

- Bateson, G. (1972). *Steps to an ecology of mind*. New York: Ballantine.
- Bateson, G. (1972). *Mind and nature: A necessary unity*. New York: Bantam.
- Bateson, G. & Bateson, M. C. (1987). *Angels fear: Towards an epistemology of the sacred*. New York: Macmillan.
- Bostic St Clair, C. & Grinder, J. (2000). *Whispering in the wind*. Scotts Valley, CA: J & C Enterprises.
- Goldstein, J. (2002). The singular nature of emergent levels: Suggestions for a theory of emergence. *Nonlinear Dynamics, Psychology, and Life Sciences*, 6, 293-309.
- Kauffman, S. A. (1993). *The origins of order: Self-organization and selection in evolution*. Oxford: Oxford University Press.
- Kauffman, S. A. (1995). *At home in the universe: The search for the laws of self-organization and complexity*. Oxford: Oxford University Press.
- Kauffman, S. A. (2000). *Investigations*. Oxford: Oxford University Press.
- Korzybski, A. (1958). *Science and sanity; an introduction to non-Aristotelian systems and general semantics*. Lakeville, CT: International Non-Aristotelian Library Publishing Co.
- Malloy, T. E., Bostic St Clair, C. & Grinder, J. (In press). Steps to an ecology of emergence. *Cybernetics & Human Knowing*.
- Palmer, S. E. (1999). *Vision science: Photons to phenomenology*. Cambridge, MA: The MIT Press.