Journal of Space Philosophy 10, no. 1 (Spring 2021)

Recursive Distinctioning and the Basis of Distinction

By Louis H. Kauffman and Joel Isaacson

I. Introduction

This paper is a short introduction to recursive distinctioning (RD).[1] We first give a model for RD that is faithful to all our ideas that an RD should satisfy. We then discuss other partial RDs and we discuss the role of Spencer-Brown's *Laws of Form*[2] in the articulation of distinctions and recursions. We end with a reflective epilogue.

II. What is RD?

RD means just what it says. A pattern of distinctions is given in a space based on a graphical structure (such as a line of print or a planar lattice or given graph). Each node of the graph is occupied by a letter from some arbitrary alphabet. A specialized alphabet (SA) is given that can indicate distinctions about neighbors of a given node. The neighbors of a node are all nodes that are connected to the given node by edges in the graph. The letters in the SA are used to describe the states of the letters in the given graph and at each stage in the recursion, letters in the SA are written at all nodes in the graph, describing its previous state. The recursive structure that results from the iteration of descriptions is called RD. Here is an example (Figure 1). We use a line graph and represent it just as a finite row of letters. The SA is {=, [,], O} where "=" means that the letters to the left and to the right are equal to the letter in the middle. Thus, if we had AAA in the line then the middle A would be replaced by =. The symbol "[" means that the letter to the left is different. Thus, in ABB the middle letter would be replaced by [. The symbol "]" means that the letter to the right is different. Finally, the symbol "O" means that the letters both to the left and to the right are different. SA is a tiny language of elementary letter-distinctions. Here is an example of this RD in operation where we use the proverbial three dots to indicate a long string of letters in the same pattern. For example,

```
… AAAAAAAAAABAAAAAAAAAA … is replaced by
… =========]O[========= … is replaced by
… ========]OOO[======== … is replaced by
… =======]O[=]O[======= ….
```

Figure 1: The First Few Steps of RD

[1] Joel Isaacson, Autonomic string-manipulation system, US Patent 4286330, filed April 26, 1979 and issued August 25, 1981.

[2] George Spencer-Brown, *Laws of Form* (London: G. Allen and Unwin, 1969).

Note that the element]O[appears from the simple difference between B and its neighbors, and that]O[then replicates itself in a kind of mitosis or DNA replication activity.

RD is the study of systems that use symbolic alphabetic language that can describe the neighborhood of a locus (in a network) occupied by a given icon or letter or element of language. An icon representing the distinctions between the original icon and its neighbors is formed and replaces the original icon. This process continues recursively.

In Figure 2, we illustrate further steps in the recursive process (with a fixed boundary condition). Note the dialectical flavor of the continued patterning. In this model, we have used synchronous processing so that each row is fully worked out before becoming the next row. It is convenient, particularly for pattern investigation, to use synchronicity, but it is not necessary. Many asynchronous variations are possible, and we encourage the reader to explore these on her own.

```
*AAAAAAAAAAAAAAAAABAAAAAAAAAAAAAAAAA*
*                ]O[                *
*               ]OOO[               *
*              ]O[ ]O[              *
*             ]OOOOOOO[             *
*            ]O[     ]O[            *
*           ]OOO[   ]OOO[           *
*          ]O[ ]O[ ]O[ ]O[          *
*         ]OOOOOOOOOOOOOOO[         *
*        ]O[             ]O[        *
*       ]OOO[           ]OOO[       *
*      ]O[ ]O[         ]O[ ]O[      *
*     ]OOOOOOO[       ]OOOOOOO[     *
*    ]O[     ]O[     ]O[     ]O[    *
*   ]OOO[   ]OOO[   ]OOO[   ]OOO[   *
*  ]O[ ]O[ ]O[ ]O[ ]O[ ]O[ ]O[ ]O[  *
* ]OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO[ *
*]O[                             ]O[*
```

Figure 2: An Extended RD Recursion with Boundary Conditions

RD processes encompass a very wide class of recursive processes in this context of language, geometry, and logic. These elements are fundamental to cybernetics and cross the boundaries between what is traditionally called first- and second-order cybernetics. This is particularly the case when the observer of the RD system is taken to be a serious aspect of that system. Then the elementary and automatic distinctions within the system are integrated with the higher-order discriminations of the observer. The very simplest RD processes have dialectical properties, exhibit counting, and exhibit patterns of self-replication. Thus, one has in the first RD a microcosm of cybernetics and perhaps, a microcosm of the world.

III The Concept of RD

We ask the reader to examine the chart in Figure 3, taken from Isaacson's patent document.[3] The chart is a description of the RD process. Note that at a certain point we see described the appearance of a dialectical process, and then the repetition of this dialectic throughout the continued recursion of description begetting description in an endless round. Distinctions are made between surface structure and deep structure and at a certain point in the chart, it is indicated that, in the perfect dialectical triad, the idea of RD occurs. Indeed, the idea of the RD is very much the idea of dialectical process, and in these models, we see the automatic working out of a dialectical process in its most elementary form.

For us, the observers of the simple RD, there is an experience of recognition in seeing that this simple process mirrors the elementary processes of our own thought and discrimination. At that point of recognition, the most fundamental problem arises: What is the source of the distinctions that we perceive?

On the one hand, one can recognize that for a human observer a distinction is always accompanied by an awareness or consciousness of that distinction. Furthermore, it is often the case that what is seen to be distinct depends upon the entire context of the event. A good example is the detection of the blind spot in the eye. This hole in our vision is normally not seen at all, but it can be revealed by looking in a direction to the left of a right thumb with the right eye (left eye closed). Then the thumb can disappear in the visual field, indicating the blind spot, but there is never a hole in the visual field. Some distinctions are distinctions for one modality of perception but not another. All distinctions that humans have are supported by their nervous system, the biology and physics of the organism, and the context in which these distinctions are framed. The context almost always involves a language of description, and that language itself is composed of distinctions.

[3] Isaacson, Autonomic string-manipulation system.

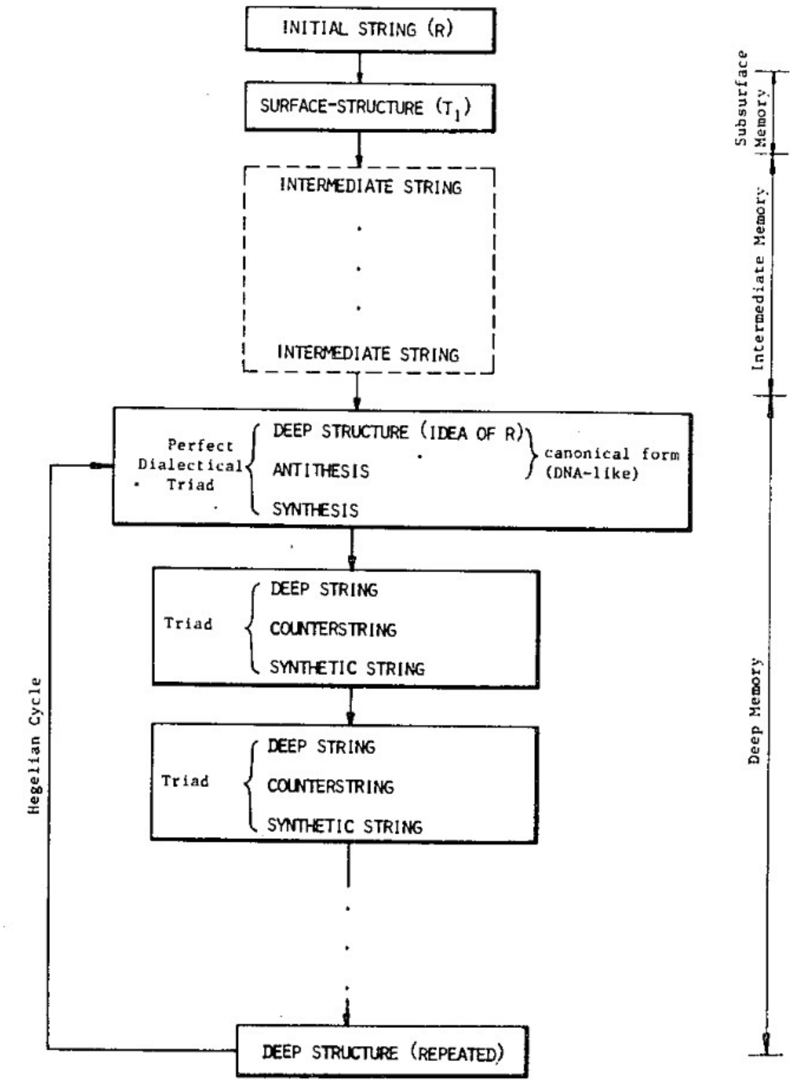


Figure 3: RD Structure and Deep Structure

Figure 3 can be interpreted as a particularly apt and process-oriented description of the action of distinctions in cognition for a human being. It is also a particularly apt description of the simple RD automaton we described in Section 2. Our intent is not to confuse these two domains but to point out the analogy between them. In a certain sense, the RD automaton engages in a form of cognition and the difference between its cognition and ours is worth contemplating.

By the same token, the RD automaton is based in distinctions that arise in the contiguity of simple elements. In this case, the elements are characters in symbol strings. The analogy can be carried forth to situations in cellular biology where the interactions are those of cells or constituents of cells, and the distinctions have to do with the direct interactions of molecules or with the making and breaking of cellular boundaries. In this arena, significant distinctions are seen to be in operation, apparently independently of our individual cognition and awareness.

This leads to the inevitable discussion of the notion of distinctions independent of human awareness. We understand that such distinctions occur in other organisms and indeed within our own organism. So, the digestive system makes its distinctions with regard to the food we give it, and thereby enables the continuance of the body. So, my computer does its operations, independent of my possible understanding of its programming.

The RD automaton can suggest, in this field of analogies, that certain processes of distinction and indeed language precede the consciousness that we take to be the locus of distinctions for our understandings. Some reflection may convince the person who thinks about these ideas that the conception of distinction is circular. Distinctions beget distinctions in an endless round. And once again the RD automaton is a simple model of that dialectical process.

IV. Synchronicity

RD processes, as we have discussed them, are synchronous processes in the sense that several variables (the characters in a string for example) are replaced at the same time by a globally defined rule. It is possible to also discuss and investigate asynchronous processes where the updating occurs locally and in different orders than the simultaneity we have imposed. Nevertheless, we shall in this discussion adhere to synchronous processes and leave the asynchronous for another time (*sic*).

The general synchronous process is described very succinctly in mathematical terms. Let there be given a set of variables $x_1, x_2, \ldots, x_n$ and a collection of functions $F_1, F_2, \ldots, F_n$ where each F_k is a function of these variables. We may write $F_k(x_1, x_2, \ldots, x_n)$. Then we define a synchronous process where the variables are updated by the equations

$$x_1' = F_1(x_1, x_2, \ldots, x_n)$$
$$\ldots$$
$$x_n' = F_n(x_1, x_2, \ldots, x_n)$$

If we let x denote the vector of values $x = (x_1, x_2, \ldots, x_n)$, then the system can be written concisely as $x' = F(x)$ where F is the vector of F-values. The crux of a synchronous process depends on the choice of the rules of $F(x)$. In the case of the RD process described in Section 2, we have an F that is defined on triples of values (a character and its neighbors to the left and to the right). The possible values are the symbols], [, =, and 0 and any other distinguishable character symbol. We have:

F(A,B,C) = 0
F(A,A,C) = [
F(A,B,B) =]
F(A,A,A) = =

where A, B, and C denote signs so that A, B, and C are distinct.

The output of the function of three variables is the new character that replaces the middle variable. In a long string of characters, this computation is performed for each triple and the results are stored until all computations are complete. Then the new row of characters replaces the original row. This is the synchronous model for the one-dimensional RD.

Note that the new characters (0, =, [,]) are iconic for the distinctions they represent. Thus, this orthodox RD has the characteristics of distinctions involving adjacency and iconicity. Each new character is an icon for the distinction that it connotes. This property singles out this RD from the vast collection of possible recursions that involve only four values and three variables.

It is interesting to examine the simplest examples of the RD recursion. For example, we can use strings of length three with the boundary condition that the end characters are always seen to be different from the emptiness on their right or their left. Then we have a period two oscillation as shown below.

ABC
000
[=]
000
[=]
000
...

If we use strings of length four, then we can attain a period three oscillation.

A B C D
0 0 0 0
[= =]
0 [ ] 0
0 0 0 0
[= =]
0 [ ] 0
...

There is a range of periods and behaviour to be explored in this simplest RD.

If we consider functions at this same level of simplicity, then some analogous behaviour can be observed. For example, let 0 and 1 denote the two basic Boolean values and let <*x*> denote the negation of *x* so that <0> = 1 and <1> = 0. Then we can define a simplest recursion by *x*′ = <*x*>, leading to a period two oscillation ...01010101... A next simplest example that leads to a period four oscillation is

$$x' = y,$$
$$y' = <x>.$$

Beginning here, we can construct many oscillators and many patterns. The RD phenomenon occurs very near the beginning of this hierarchy of mathematical possibilities.

In fact, all the processes of the form $x' = F(x)$ can be seen as RD. It is a matter of investigation of the details of the recursion $F(x)$ to find out how the rules of these distinctions operate. A good arena for examining this is the field of cellular automata where experimentation with rules has led to a vast zoo of phenomena. Not all such recursions take part in the dialectical process of the RD, but all can be seen as the consequence of making distinctions and expressing them in a recursive domain.

V. The Audioactive Recursion

1
11
21
1211
111221
312211
13112221
1113213211
...

Illustrated here is a pattern of *recursive description*. Each line is a description of the previous line. To see this, read the lines aloud. The second line says, "one one," and that is a description of the first line. The third line says, "two ones," and that is a description of the second line. The next line says, "one two, one one," then "one one, one two, two ones," and so on. The full alphabet for this recursion is the set of numerals {1, 2, 3}, and they are alternately signs and elements of the description of a pattern. This "audioactive sequence" was extensively investigated by John Horton Conway,[4] and it has many mathematical properties.

A variant on the above recursion that is quite interesting is to start with the number three rather than one. Then we have

3
13
1113
3113

[4] John H. Conway, "The Weird and Wonderful Chemistry of Audioactive Decay," *Eureka* 46 (1986): 5-16.

132113
1113122113
311311222113
...

It is not hard to see that if the rows are r_1, r_2, r_3, ... then r_{n+3} is an extension of r_n. This means that we can build three infinite rows A, B, C that are in dialogue with each other in the sense that B describes A, C describes B, and A describes C.

A= 311311222113...
B= 13211321322113...
C = 1113122113121113222113...

There is much to explore in this recursion. A description is of course certainly a distinction, but the distinctions made by this form of description are of a more complex nature than the adjacencies in the first RD that we have discussed.

Remarkably, the audioactive sequences shown here are based on a very small alphabet of numerals (1, 2, 3). It is a bit mysterious what can come from only one, two and three.

VI. Formal Arithmetic

Here we give an example of *formal arithmetic,* governed by a very simple recursive distinctioning with contiguity of characters.

Here are the formal arithmetic rules for changing a string of characters consisting in the characters “*,” ”<,” and “>.”

** is replaced by <*>
>< is replaced by (nothing).

Note that in this recursion, we rely on adjacency to detect the patterns that are to be replaced. Detection and replacement of pattern is the form of distinction in this model.[5]

If we start with a row of five stars, then the following recursion will occur.

*****
<*><*>*
<**>*
<<*>>*

If you interpret * as the number 1, <*X*> as 2*X* for any number *X*, and *XY* (adjacent strings) as *X* + *Y*, then you will see that result of the string replacement will be a coding

[5] Louis H. Kauffman, “Arithmetic in the Form,” *Cybernetics and Systems* 26, no. 1 (1995): 1-57.

of the number of stars in the first row. In this example, <<*>>* = 2(2(1)) + 1 = 4+1 = 5. In fact, the result of the recursion can be interpreted as the binary coding for the original number of stars. Here is another example.

****

<*><*><*><*><*><*><*><*>*

<********>*

<<*><*><*><*>>*

<<****>>*

<<<*><*>>>*

<<<**>>>*

<<<<*>>>>*

The result tells us that there are $2^4 + 1 = 17$ stars in the first row.

Here is the method to convert the result of the recursion to binary notation. Start with the result. Remove the left pointing arrows. Replace the stars by instances of 1. Place a 0 in between each >> and place a 0 at the right if there is no star. Then remove all the right pointing arrows.

For example:

<<<<*>>>>*

>>>> 1>>>>1

1>0>0>0>1

10001.

This recursion is a simple automaton that does arithmetic and converts numbers into binary. Everything proceeds from two forms of distinction. One form recognizes pairs of stars and replaces them by a bracketed star. The other recognizes oppositely pointing pairs of brackets and erases them. At first, it is not obvious that these two forms of distinction are a basis for calculations in arithmetic. Just so, there are recursive processes behind our familiar actions that would seem unfamiliar until we examine them. Consider an everyday action such as speech and ask yourself how you produce the highly patterned sounds that constitute your voice. It is a long story in new territory to articulate what happens in that domain.

Recursion in arithmetic is itself unknown territory for most mathematicians and scientists at this time. For example, consider the following *Collatz Rule:*

If n is even, replace n by $n/2$.
If n is odd replace n by $(3n+1)/2$.
If $n = 1$, STOP.

For example, 7 > 11 > 17 > 26 > 13 > 20 > 10 > 5 > 8 > 4 > 2 > 1. It is conjectured that for any natural number *n*, this process will, after a finite number of steps, terminate at 1. The problem has been known since the 1940s. It remains unsolved at the time of writing. Many adventures can be had in exploring the Collatz Recursion. It is based on little more than elementary arithmetic and the distinction between even and odd. Kauffman has used the arithmetic automaton based on a star and bracket to explore the Collatz problem, but it has not yielded up its secrets yet. This problem indicates the depth of simple recursions in the structure of elementary mathematics. Mathematics itself is built from distinctions. We are often surprised by the phenomena that emerge just from mathematics itself in the face of recursion.

VII. Laws of Form

This example is different than the previous ones. Here we start with distinction, but we do not institute rules for a synchronous recursion. The system we describe is due to G. Spencer-Brown in his book *Laws of Form*.[6]

Here the sign $\overline{\;\;}|$ stands for the distinction that it makes between inside the sign and its outside. Spencer-Brown calls $\overline{\;\;}|$ the mark, and allows it to refer to any given distinction, including itself. The inside of the mark is unmarked. The outside of the mark is marked (by the mark).

The mark $\overline{a}|$ can be interpreted as an instruction to cross the boundary of a distinction. In that mode, we have denoted the value obtained by crossing from the state a. Thus $\overline{\overline{\;\;}|}|$ is unmarked, since we have crossed from the marked state, and $\overline{\;\;}|$ is marked since we have crossed from the unmarked state. An extra mark in the space outside the mark is redundant since that space is already marked. Consequently, we may write $\overline{\;\;}|\overline{\;\;}| = \overline{\;\;}|$. Thus, we have two basic replacement rules:

$$\text{Crossing: } \overline{\overline{\;\;}|}| =$$
$$\text{Calling: } \overline{\;\;}|\overline{\;\;}| = \overline{\;\;}|.$$

A calculus arises from this so that one can reduce or expand arbitrary expressions in the mark. For example,

$$\overline{\overline{\overline{\;\;}|\,\overline{\;\;}|}|\,\overline{\;\;}|}|\,\overline{\;\;}| = \overline{\overline{\overline{\;\;}|}|\,\overline{\;\;}|}|\,\overline{\;\;}| = \overline{\overline{\;\;}|}|\,\overline{\;\;}| = \overline{\;\;}|.$$

One can prove that the simplification of an expression is unique and go on to consider the algebra that is related to this arithmetic.

[6] Spencer-Brown, *Laws of Form*.

In the algebra, we have identities such as AA = A for any expression a, and $\overline{\overline{A}\rceil}\rceil$ = A for an expression A. Remarkably, the algebra is quite non-trivial and leads to a new construction for Boolean algebra and new insights into the nature of logic.

Here, a great deal of structure comes to light if we decide not to use synchronicity immediately and to elicit designs that are asynchronous and have behaviour that is independent of choices of time delay. In this way, the timeless structure of such asynchronous structures enters and supports the creation of the rhythm and temporality of recursive computation. In this way, one can consider recursive structures related to the calculus of indications (as this calculus of the mark is called).

An elementary structure of great significance appears from these equations:

$$M = \overline{aN}\rceil$$
$$N = \overline{bM}\rceil$$

To see what happens here, let *a* and *b* be unmarked. Then we have

$$M = \overline{N}\rceil$$
$$N = \overline{M}\rceil$$

If M = $\overline{\ }\rceil$ and N = $\overline{\overline{\ }\rceil}\rceil$, these values satisfy the equations, and so the system is in a stable state. Similarly, if M = $\overline{\overline{\ }\rceil}\rceil$ and N = $\overline{\ }\rceil$, then the system is in a stable state. We see from this that M and N together form a memory. In a possible world of recursions, the memory can maintain a particular pair of values. In this way, the binding of structure across time emerges from the timeless eternity of forms.

Furthermore, if in this memory we were to change *a* or *b* to the marked state, we could influence the memory to change state. A momentary change in the inputs *a* and *b* can reset the memory. In this way, circular systems of equations can be made that correspond to circuitry at the base of computing, and the essential design of digital computers can be accomplished in the language of the mark and its algebra.

A key function that can be described in this algebra is the operation of exclusive or. We denote exclusive or of *A* and *B* by *A* # *B*. It is expressed in the algebra of the mark as:

$$A \# B = \overline{\overline{A\overline{B}\rceil}\rceil\,\overline{\overline{A}\rceil B}\rceil}\rceil .$$

The reader will note that *A* # *B* is marked only when one of *A* or *B* is marked, but not both. Thus *A* # *B* can indicate whether *A* and *B* are distinct or not. If *A* = *B* then *A* # *B* is unmarked, but if *A* ≠ *B*, then A # B is marked. It is this ability of the logical algebra to make distinctions that gives it the capacity to be the underpinning for models of recursive distinguishing.

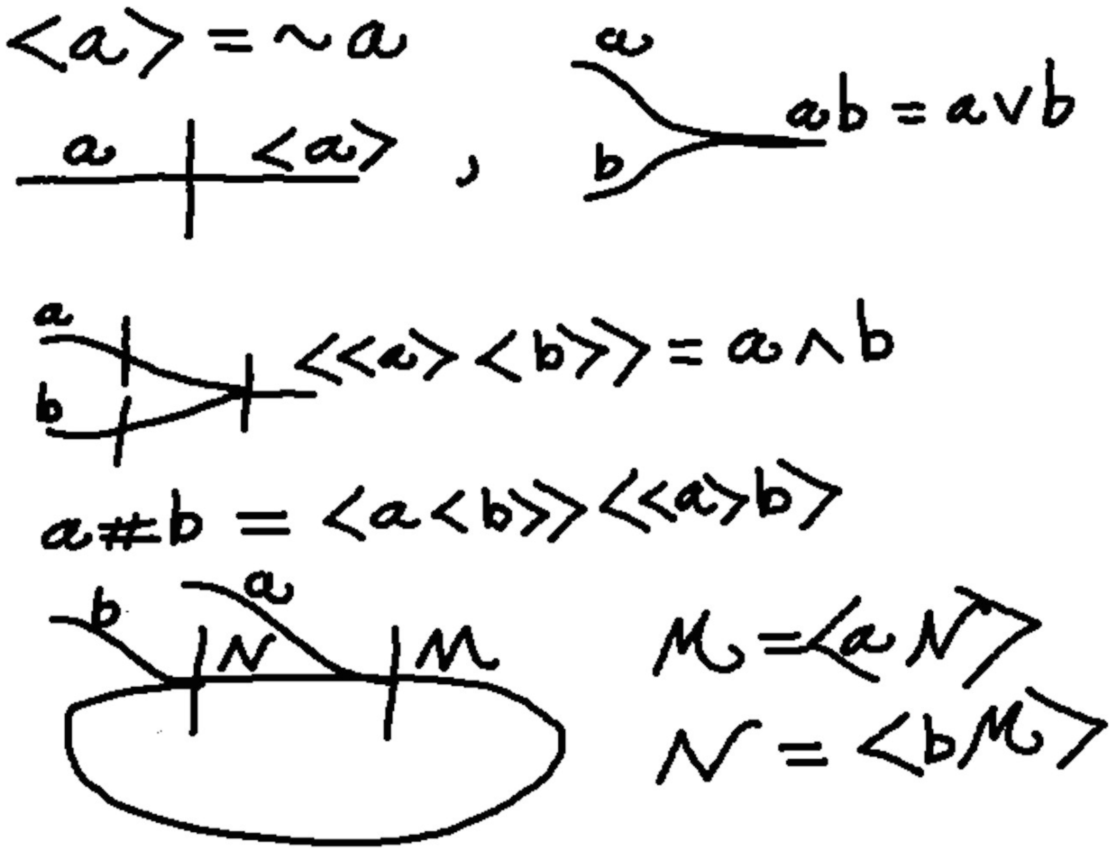


Figure 4: Diagrams for Distinction Operators

In Figure 4, (using < > for the mark) we indicate the bare bones of diagrams for these distinction operators and for the memory. It should be apparent that the memory can be regarded as a graph with a special even cycle that can be labeled with states so that there is an evaluation balance at each node. The node with the vertical marker is the distinction operator, and it is an analog of a NOR gate in electronics. Systems composed of such diagrams can be used to model the basic workings of any digital computer, and so make a Turing complete structure. In this way, we see that all computation can be seen to be based on distinctions and recursions. This way of creating a basis is not quite in the mold of RD where all distinctions are created in relation to contiguities and the formation of alphabets. In this circuit paradigm the distinctions act as states and transmissions of elementary information in the cycles and trees of graphical structures that are themselves seen as patterns of distinction operators.

By regarding the distinction operators as graphical carriers of information, the structure of these graphs in Laws of Form can be shaped as models of automata that can be built in hierarchical fashion and so concatenate into full blown designs for digital computers and information systems. By the same token, these designs can support the operations of any RD of the type that we have described in this paper. We can enfold the RD concept and designs into a full context of computation and communication.

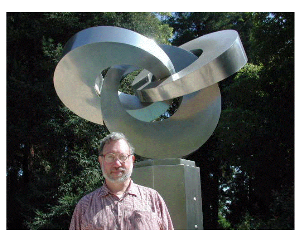
In this way, we come full circle for the structure of RD in that the consideration of a distinction and the evolution of an algebra and operations of distinction creates the platform on which RD can be constructed. But the process by which we have evolved this algebra and logic is, in fact, the already given RD capabilities of our organism and our abilities to make engineering and mathematical design.

We can reach deeper into the biological and physical world to find sources that underpin the emergence of distinctions. This will inevitably happen in the future development of RD and the understanding of distinction.

VIII. Epilogue

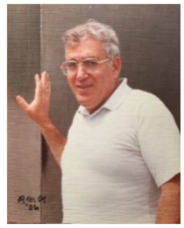
We have discussed in this essay the structure of recursive distinctioning and variants of it that are based on some or all its themes. There is a need in thinking to find simple basic principles and constituents from which all other apparent phenomena can be built. Here it is proposed that distinctions are such elementals. But this assertion must be taken with a grain of salt, for there can be no definition of distinctions. Distinctions escape the net of the conceptual exactly because the conceptual is based upon certain fundamental distinctions. And distinctions escape the simplicity of the physical for the same reasons. No one has ever isolated a distinction in nature that is not dependent upon some particular system of observations that give rise to such distinctions for given observers. Nevertheless, we insist that at the bottom of any investigation there will be fundamental distinctions being produced in an endless round of recursion that is essential RD.

****



About the Authors:

Louis H. Kauffman is Emeritus Professor of Mathematical Physics and Cybernetics at the University of Illinois, Chicago. He has degrees from MIT and Princeton. He has 170 publications. He was the founding editor for the *Journal of Knot Theory and its Ramifications*, and he writes a column entitled "Virtual Logic" for the journal *Cybernetics and Human-Knowing*. He was president of the American Society for Cybernetics from 2005-2008. He introduced and developed the Kauffman Polynomial. He was the recipient of the 2014 Norbert Wiener Award of the American Society for Cybernetics.



Joel D. Isaacson has pioneered in RD cellular automata since the 1960s. RD was rooted in studies relating to the analysis of digitized biomedical imagery. He utilized NASA's computing facilities at the Goddard Space Flight Center in Greenbelt, MD for the initial stages of this research. His research has been supported over the years by DARPA, SDIO, NASA, ONR, USDA and a good number of NIH institutes. He is Professor Emeritus of Computer Science, Southern Illinois University, and Principal Investigator for the IMI Corporation.

Editors' Notes: Recursive Distinctioning (RD) is the term applied by Dr. Joel Isaacson, in his 1981 patent titled "Autonomic String Manipulation System." This essay is the latest Kauffman and Isaacson publication providing a short introduction to RD. Readers will find extensive references to their more detailed publications in the Special Issue of the *Journal of Space Philosophy* 5, no. 1 (Spring 2016). Dr. Isaacson concentrates on intelligence in his Chapter 24 of "The Intelligence Nexus in Space Exploration: Interfaces Among Terrestrial, Artifactual and Extra-Terrestrial Intelligence," in *Beyond Earth: The Future of Humans in Space*, edited by Bob Krone (Burlington, ON: Apogee Space Books, 2006). He later published "Nature's Cosmic Intelligence" in the Fall 2012 issue of the *Journal of Space Philosophy*. The overview statement for this 2021 publication is that "The discovery that our universe contains information and intelligence in a process that is basic also to human perception and cognition," is a contribution to the understanding of the cosmos by Drs. Isaacson and Kauffman. ***Bob Krone and Gordon Arthur.***