Recursion Of Binary Space As A Foundation Of Repeatable Programs

Jeremy Horne
RhinoCorps
Las Cruces, New Mexico 88002 USA

ABSTRACT

Every computation, including recursion, is based on natural philosophy. Our world may be expressed in terms of a binary logical space that contains functions that act simultaneously as objects and processes (operands and operators). This paper presents an outline of the results of research about that space and suggests routes for further inquiry. Binary logical space is generated sequentially from an origin in a standard coordinate system. At least one method exists to show that each of the resulting 16 functions repeats itself by repeatedly forward-feeding outputs of a function operating over two others as new operands of the original function until the original function appears as an output, thus behaving as an apparent homeostatic automaton. As any space of any dimension is composed of one or more of these functions, so the space is recursive, as well. Semantics gives meaning to recursive structures, computer programs and fundamental constituents of our universe being two examples. Such thoughts open inquiry into larger philosophical issues as free will and determinism.

Keywords: Recursion, Binary Logic, Cosmological Semantics, Quantum Semantics, Computational Space, Lattice Theory, Automata, Free Will, Epistemology, Information Theory

BACKGROUND

Recursion is a mechanical operation, but its significance is founded in the deep structure of the universe. Philosophers, physicists, psychologists, and others have suggested that this micro world is the substratum of the entire macro world as we experience it, the substratum sought by Descartes [1], Aristotle [2], Wheeler [3], Wolfram [4], Piaget [5], and Horne [6], among others. John Locke argued that a deeply fundamental substance must logically exist if we have sense impressions. In short, these impressions must emanate from something, i.e., a substratum [7]. The "smallest of the small" is the subject of Leibniz's Monadology and the atoms of Democritus.

If one applies Descartes' method of repeatedly subdividing anything in what is presumed to be the physical world, they will arrive at the smallest unit - Planck area (or even strings). Herein exists at least a part of the basis of the observations concerning the substratum made by Wheeler, Wolfram, Piaget, and Horne. Yet, Planck areas must have a world, or context, in which to exist, and vacuum fluctuations may be the candidate. Proceeding with the logic even further, we may see that the common denominator of Planck area and vacuum fluctuations merely are two aspects of the same phenomenon – displacement, or the substratum. This micro world, discovered by the epistemology of Cartesian reductionism, contains the deep binary structure that permeates all computations, including recursion.

We will symbolize our binary micro world by the convenient and familiar symbolization that is most immediately relevant to the science of binary computation, 0 and 1. (Of course, other symbols may be used, but the familiarity of 0 and 1 will expedite understanding how this philosophy has a direct application to the electronic computation that permeates our lives.) Yet, the space occupied by that which the symbols represent and the processes occurring within that space have major import. This import holds not just for the field of computer science and cybernetics. It also holds for the genesis of thought and natural computation as properties of being itself. Our trail of discussion will take us to logical space, major dynamics within that space and its essence, and the semantics of the syntax of logical space. We will exit this discussion by paying attention to the implications for the future.

GENERATING BASIC LOGICAL SPACE

Logical space is generated with various conventions. My method is intuitive, based on the way humans think about ordering based on increasing quantity. The unit of functionally complete logical space is a four by sixteen matrix, the columns (permutations of the values, taken four at a time), demarcated by the sixteen functions, f_0 through f_{15} [8]. The *p* and *q* rows are permutations of the two values available in a binary world, symbolized by 0 and 1. Base two values increase sequentially, as by counting, originating at the X-Y intersection of a standard two-dimensional coordinate system and radiating outward and upward toward the upper-right-hand part of space. A segment of that space for close-up viewing follows. The next smaller image displays the complete space.

p	q	f_0	f_1	. . .	f_{13}	f_{14}	f_{15}
1	1	0	1	. . .	1	0	1
1	0	0	0	. . .	0	1	1
0	1	0	0	. . .	1	1	1
0	0	0	0	. . .	1	1	1

Table 1 Partial Table of Logical Space

II I

p	q	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

0 (origin)

III IV

Table 2 Table of Logical Space

Because of the p/q placeholder permutations, this generation operates within specific bounds and is deemed a deductive, or closed space. Each of the sixteen columns of binary values that serve as functions, or operators, act on dyadic relations of functions to produce another function, also found within the deductive space.

The binary logical space syntax can be instantiated with any set of binary values, such as:

p	q
0	1
False	True
Large	Small
Up	Down
Strong	Weak

Table 3 Example of binary values

For any binary space, all the processes applying to the syntax, such as functional recursion, apply to all the instances. That is, the instantiation of the variable carries with it the properties and processes relating to the variable, including the semantics of Planck areas and vacuum space fluctuations of the micro world.

For the computer scientist, the 0s and 1s are all too familiar symbols of binary enumeration. In doing logical calculations, one displays a problem of semantics, such as

p (f_3)	$\supset$ (f_{13})	q (f_5)
1	1	1
1	0	0
0	1	1
0	1	0

Table 4 The containment function, f_{13}

Three functions are involved: f_3, $f_{5,}$ corresponding to the first and third columns of logical space in Table 2, as the permutations of the 0s and 1s, and the function (f_{13}) representing containment, or "material implication". A function operating on two other functions is the manner in which one moves from one position in logical space to the next. Each function has a semantic, where the function represents a relation or process, such as "and". The function f_{13}, perhaps is the most important, for it describes processes occurring within a presumed closed space. The 0 represents a larger space and the 1 a smaller one. 0 contains 0; 1 contains 1; 0 contains 1. In these three cases, there is information, or 1. It is not the case that entropy, or 1, yields potential, or energy; hence there is no information, or the result is something unknown to us (such as there being the possibility that, indeed, entropy could result in energy"). In cosmological semantics, the 0 represents chaos, the inchoate, or that which is the potential from which emerges information, or entropy (that which consists of the things of which we are aware in our universe).

It may be noted that with variables more than two, the resulting space is merely a complex version of the basic dyadic space. (With three variables, p, q, and r, one starts counting in the lower left-hand corner, 000, vertically to 111, as permutations of the variables, and horizontally, column 00000000 through 11111111, each as an eight-place function in a 256-row space. The number of variable (n) permutations is 2^n, the same as the number of rows.)

FUNCTIONAL RECURSION

Entities that maintain themselves without changing in an environment are said to be homeostatic. One may say that the re-emergence of the same entity after contact (and possible temporary and immediate changes) with the environment is a self- maintaining (homeostatic) automaton. It is recursive in that it self-repetitive. The outputs of a function operating over two others are repeatedly forward-fed into the preceding dyadic relationship as new operands of the original function until the original function appears as an output, thus evidencing the homeostatic, or self-maintaining, character of the function, i.e., recursive. Functions as operators may become operands, or functors; thus, process may become object and vice versa. Graphs can be constructed, with edges pointing to the re-appearance of the function [9]. A critical problem arises as to how the recursion can be initiated. Of course, a function, or object space, cannot simply repeat itself without an instruction set, even if the set were embedded in the space. Therefore, the algorithm, or process, must lie outside of that object space. This issue will become clearer later in our discussion of larger binary spaces and processes governing the larger scale structure of the universe, itself. After all, the processes innate in the universe are manifested in all aspects of the macro world.

BINARY SPACE RECURSION

Numerically ascending binary space is created from elements within two binary placeholders and a simple sequencing rule. Both of these result in space originating from a point on a standard four-quadrant coordinate system. Perforce, since each of the 16 functions in basic logical space, of which compound general binary space is composed, is recursive, the compound space, itself is recursive [10]. A multiply dimensioned matrix displays the relationship of functionally recursive cycles of the space. The following illustrates a sample of logical space occupied by eight functions, each function represented by a rectangular solid. Each solid varies in height according to the number of iterations or cycles it takes for the outputs to be fed into the function before the function repeats itself. The figure depicts the three-dimensional character of the space after each of the eight functions has repeated itself once.

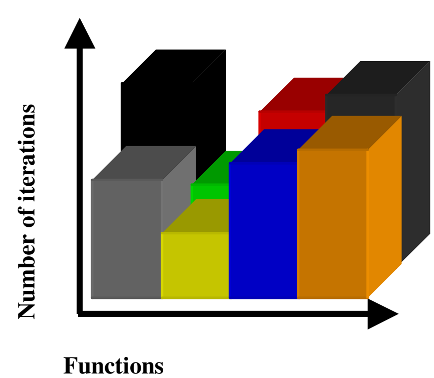

Table 5 Hypothetical iteration of functions– one cycle. Each block represents a function.

This is important, as some functions repeat themselves with fewer cycles than others. Later we will discuss the computation of these larger spaces. For the moment, some technical aspects of recursion need attention. As to computer programs being able to repeat themselves, it should be noted that each row of a machine language array of 0s and 1s, to which all programs are reducible, contains a number of "complete" functions, i.e., each function has four places.

However, for any space (real or logical) reduced to binary functions, there is an issue of "partial functions". Consider two lines of concatenated horizontal spaces: "Irregular spaces" may arise in expressing any phenomenon by binary space.

Line 1: 0100 0010 00 and

Line 2: 1010 0001 0011 10 (underlines inserted for visual convenience)

The first binary space is 10 digits long. The second is 14. The first consists of two and a half functions: $f_4 + f_2$ + 00. This last portion could be any function ranging from 0000 through 0011. For the second, the first four digits are 1010, or f_5. The second set of four is 0001 is f_1. The last two, 10, could be any function between 1000 through 1011. In such cases of ambiguity, it remains to be developed a model that accounts for the range of possible outcomes, perhaps a probability of the next being based on what has occurred before. However, the range of outcomes is discrete. Perhaps the edge of binary spaces with partial function can be deemed probability space recursive within a specific range.

Now that the discussion of recursive logical functions is complete, a final technical question remains of arriving at algebra of recursive spaces. A space composed of an i by j matrix of 0s and 1s is composed of subspaces of n four-place elements (functions), and those elements, themselves, are recursive. The same method for generating recursion in each of the 16 binary functions also may be applied to spaces. We start with placeholders derived from the table reflecting binary possibilities, f(p,q), given the binary logical space syntax described above, where p = f_x and q=f_y. Yet, the functions, of f_x and f_y refer to spaces, rather than individual functions. Furthermore, there most likely will be more than 16, depending upon the agreed-upon size of the space. In any event, the size of each space used as a function must be equal. We take a function space f_* in a syntax of special recursion.

For p:

$f_*(f_x,f_y) \rightarrow f_n$

$f_*(f_n,f_y) \rightarrow f_{n*}$ - Substituting the output from the previous into the p placeholder.

. . .iterations. . .

$f_*(f_{n*},f_y) \rightarrow f_x$

Thus, the first placeholder containing f_x has been shown to be recursive. Now, we advance to the second placeholder, q that contains f_y, in order to show that the whole of f_* is recursive.

For q:

$f_*(f_x,f_y) \rightarrow f_n$

$F_*(f_x,f_n) \rightarrow f_{n*}$. Substituting the output from the previous into the q placeholder

. . .iterations. . .

$F_*(f_x,f_{n*}) \rightarrow f_y$

Where:

n = first output

n* = successive outputs.

Certain configurations would have the same recursive outcomes. In theory, one would take a function of an existing space and a designated equally sized subset of a space, and forward feed the output as a replacement function of the *existing* space until the existing space re-appears. Again, as in operations with single four place functions, it is to be noted that results, or objects are processes, as well. Further, they are homeostatic. It should be an object of research to ascertain if there are "spatially recursive families". Blocks of certain 0s and 1s (uniformly sized spaces) could be represented by an algebra that would serve as shorthand to computing and assessing the outcome of larger spaces. For example, in a four-by-20 block space consisting of one four-place function by five four-place functions, we may have:

f_x:

01000100010010101111
11110001101000101010
00011111000101010100
01010100010101000001

f_y

010001110000101011011
111110001101000101010
000111101000101000100
010100011100101000001

f_*

010001110000101011011
111111110001000101010
001111101000101000100
010100011100111000001

f_n = ? Left for research.

At this juncture, the computational method allows for only dyadic relations, but this does not exclude n-yadic ones. Additionally, it may be remarked that a more complete model of binary special recursion would not have to be confined to two or three dimensions. There might even be n-operators in a multidimensional binary environment. While functions and uniform spaces composed of functions may be recursive, it remains as an exercise to assess the significance of these operations.

IMPLICATIONS OF RECURSIVE BINARY SPACE

Many of the philosophical and subsequent "practical" aspects of recursion point to a revelation about the nature of our universe, our place in it, and ultimate outcome of it all. This section will address some possible practical uses of the recursion algorithm and suggested research areas. The philosophical ramifications will be discussed later.

A potential application of functional recursion involves any entity expressible in binary values being repeatable. For example, machine language programs (the basis of all computer programs) offer themselves as obvious candidates. That is, a block of machine code acting as an operator over two other equally sized blocks would produce a fourth. After a fashion, an algebra of computations would act as a shortcut to producing code. It has been suggested to this author that another potential application is in cryptography. This method concerns a message being translated into a binary form and encoded by the recursion devices, with the decryption algorithm being either embedded in the original or sent separately. An example of an encryption key would be informing the receiver of the number of iterations the space has recursed from its origination point. The space would be "run in reverse" to produce the original.

In terms of pattern recognition applications, work has been done to generate binary spaces by a random concatenation of binary functions [11]. The result yields intriguing designs with no philosophical interpretation beyond the view that patterns emerge from seemingly random phenomena. These "basins of attraction" illustrate a theory of fractals, or self-similarity independent of scaling. We need to ask, instead of random generation, what pattern of spaces might emerge with an ordering principle or algorithm (such as one based on the intellectual complexity of functions), using individual functional recursion or spaces (as functions) that contain computer programs? Would fractals or basins or attraction appear? Here, the basins of attraction resulting from such an intentional concatenation of functions would be mappable to the phenomena producing them. In turn, the basins might be used as indicators of those phenomena. Research in pattern recognition of such basins would be needed to understand their nature. For example, would certain shapes signify certain types of phenomena?

In the worlds of artificial consciousness automata theory, any machine language program is recursive as a homeostatic automaton, given the above. In addition, massively distributed parallel processing programs could be represented in such an environment and be amenable to analysis and further development, based on approaches discussed in this paper. Two or more programs acting as automatons (recursive spaces) may be creating world of their own, not unlike Conway's "Game of Life" or multi-tiered cellular automatons. It is to be kept in mind however, that while the state space of any program is recursive, there is no excluding the situation that an apparently self-organizing system is seeking a different space in which it deems to be more stable. Naturally, if the system is self-adapting in a constantly changing environment (as with the case of a real-time interaction with a "natural" environment – synthetic system) the re-organization may be forever continuous. One may extend the discussion to modeling and simulation, in that subsystems interacting in a larger context may result in the whole simulation assuming its own identity.

On a grander scale of applications, loop quantum gravity theorists say the universe, itself, can be subdivided into Planck areas ($1.6x10^{-35}$ meters) [12]. These areas may be seen in terms of what they are not (vacuum space fluctuations), thus establishing a binary semantics. Indeed, the four-fold nature of describing the state of Planck areas by loop quantum gravity theorists appears to be amenable to a four-place binary descriptor. This descriptor, by itself, is a recursive function of logical space. For example, point, area, space, and displacement as parameters existing in space-time may be designated as existing or not existing (0 or 1), these descriptors and possible values being mappable to Smolin's geometric representations of Planck area qualities. At this juncture, an algebra of binary spaces has the potential of being the framework for describing an apparent autopoietic, or self-organizing, space, each function being a descriptor of a Planck area and its opposite (vacuum space).

On an earthly level, if everything is reducible to or expressible by a binary system, then, logically, anything is recursive. So, too, all the processes described above apply. It is useful to contemplate that any space identified for research would represent an innate universal process. Further, the larger number of samples taken, the greater understanding there would be of the nature of the universe. If the semantics of Planck area are correct, does there exist here the basis of a "theory of everything"? Currently, these ideas appear not to be testable, given available thinking and methods. However, we can start thinking about what has been demonstrated to be pure logic in terms of philosophy, that is, logic as the language of innate order in the universe.

TOWARDS A PHILOSOPHY OF RECURSION

Wheeler, Misner, Thorne, and Piaget, among others, do suggest that the universe has a binary structure. Wheeler, Misner, and Thorne even say, "...a machinery for the combination of yes-no or true-false elements does not have to be invented. It already exists" [13]. Jean Piaget argued:

> There exist outline structures, which are precursors of logical structures. It is not inconceivable that a general theory of structures will...be worked out, which will permit the comparative analysis of structures characterizing the outline structures to the logical structures characteristic of the higher stages of development. The use of the logical calculus in the description of neural networks on the one hand, and in cybernetic models on the other, shows that such a programme is not out of the question (emphasis included) [14].

The observations above apply to both the macro and micro world. Computer science is a discipline primarily concerned with computation, and, more recently, artificial and synthetic intelligence (artificial intelligence involving real-time human intervention). Attempts to model and simulate what we perceive to be reality perforce, by virtue of the fact they are computer based, must incorporate the deep binary structure we have examined and the parameters of its existence. In fact all artificial and synthetic programs (those involving human intervention in real-time), as a matter of logic, present the same issues, ones that we may refer to as the "philosophy of binary information systems". The micro world consisting of Planck areas, their opposites, and the processes relating these two is the most fundamental binary information system and is the "building block" of all information systems. Otherwise stated, the parameters allowing for its existence and processes governing (those outside the system, itself) it are imminent in all computations and their results.

The significance of recursion extends beyond merely demonstrating that computer programs can be self-replicating. We must again re-visit modeling and simulation and address the Herculean assumptions and "machinery" upon which rests this world. Whatever processes exist at the base level permeate that which emanates from the binary world. If indeed any universe (ours or any that we create) as a binary space is recursive, then, the instruction set now becomes a critical object of inquiry.

A natural philosophy of the cosmological world in terms of the micro world of Planck areas may offer some resolution in that area of conundrums, such as the "halting problem", Church's Theorem, and "the set of all sets" [15]. That is, these apparent paradoxes are manifestations of a law permeating the universe, not unlike the law of gravity. Any universe, including our own, is a subset of another. Recursion cannot occur on its own. A system or pattern can emerge only as a result of an algorithm. Algorithms for creating structures do not simply organize themselves; they need parameters and an instruction set for creation. Something must determine the nature of that set. From nothing, nothing comes. A collection of elements, such as binary functions or Planck areas (as semantics for the binary syntax) merely is a data set incapable of autopoiesis.

Within the binary-based universe exists both object and process - a function being an operator, as well as an operand. Still, the "instruction set" governing which will be what and when exists as the" cradle" (and, hence, outside) of that universe. Parallel dimensions, gravitons, and brane theory, one might argue, are attempts to find that instruction set. In absence of theories such as these, we are left with a disturbing view: our recursive universe is just a collection of data (deductive), and free will is illusory.

CONCLUSION

Our excursion into the realm of a phenomenological world has been reduced to binary space. We have seen that it is possible for that world to be self-repeating. While recursion drumbeats for the mundane and our presumed status as mere objects, that there may be free will is supported by the fact that the instruction set cannot lie within object space. Otherwise, everything, including our lives, would be fixed and pre-determined for all space-time. Induction truly lives at the most fundamental level and is embedded in the parameters that define our universe. Deduction really does not exist, except as constrained by the issues raised by Heisenberg, Church, et. al. We impose "contamination" on the parameters (including those identifying elements and the rules governing them) defining the set within which deduction occurs. While Descartes the reductionist surely can give us run for our money, the astute gambler would be wise to bet on Hume, Mill, and Russell the "inductionists".

REFERENCES

[1] Rene Descartes, **Discourse on the Method of Rightly Conducting the Reason, and Seeking Truth in the Sciences** in **The Philosophical Works of Descartes**, Vol. 1, Elizabeth S. Haldane and G.R.T. Ross, translators, Cambridge: Cambridge University Press, 1977, P. 121 and 168.

[2] Aristotle, **Metaphysics** in **The Basic Works of Aristotle,** Richard McKeon, editor,. Book I, Ch. 2, New York: Random House, Pub., 1941.

[3] C.W. Misner, K.S Thorne, J.A Wheeler, "Pregeometry as a Calculus of Propositions", **Gravitation**, Chapter 8, New York: W.H. Freeman and Company, 1973.

[4] Stephen Wolfram, **A New Kind of Science**. Champaign, IL: Wolfram Media, Inc., 2002, pp. 6-7, 18, 259, and 317.

[5] Jean Piaget. **Logic and Psychology,** New York: Basic Books, Inc., Pub., 1958.

[6] Jeremy Horne. "Logic as the Language of Innate Order in Consciousness", **Informatica**, Vol. 22, No.4, 1997, pp. 675-682.

[7] John Locke, **An Essay Concerning Human Understanding,** Book II, Chapter 23, par. 3-6, New York: Dover Pub., Vol. 1, 1959.

[8]. Jeremy Horne "The General Theory and Method of Binary Logical Operations", **Journal of Applied Science and Computations1,** 3 December 2000.

[9] Jeremy Horne "Recursion of logical operators and regeneration of discrete binary space" **(Informatica No. 24 (2000),** Pp. 275-279).

[10] Ibid. p. 278.

[11] Wuensche, A., "The Ghost in the Machine: Basins of Attraction of Random Boolean Networks," University of Sussex at Brighton: Cognitive Science Research Papers, CSRP 281, 1993, and see also more detail at http://www.cogs.susx.ac.uk/users/andywu/ (7 April 2005)

[12] Lee Smolin, **Three Roads to Quantum Gravity**, New York: Basic Books, pp. 106-124, for example.

[13] Wheeler et. al,, p. 1209.

[14] J Piaget. **Logic and Psychology** New York: Basic Books, Inc. (Piaget 1958, p. 48).

[15] Alonzo Church, **Introduction to Mathematical Logic**. Princeton University Press, 1944, P. 23.