# A New Three Dimensional Bivalent Hypercube

**Description, Analysis, and Prospects for Research**

**Jeremy Horne**
**Avenida Moreras 131**
**San Felipe, Baja California C.P. 21850**
**mindsurgeon@hotmail.com**

**Keywords: 3-D Hypercube, Logic, Pattern Classification, Binary Systems, Innate Order**

## Abstract

A three dimensional hypercube representing all of the 4,096 dyadic computations in a standard bivalent system has been created. It has been constructed from the 16 functions arrayed in a table of functional completeness that can compute a dyadic relationship. Each component of the dyad is an operator as well as a function, such as "implication" being a result, as well as an operation. Every function in the hypercube has been color keyed to enhance the display of emerging patterns. At the minimum, the hypercube is a limited "multiplication table" or table of dyadic computations and values that shorten the time to do operations that normally would take longer using conventional truth table methods. It also can serve as a theorem prover and creator. With the hypercube comes a complete system without the need for axioms. The main significance of the 3-D hypercube at this point is that it is the most fundamental way of displaying all dyadic computations in binary space, thus serving as a way of normalizing the rendition of uninterpreted, or raw, binary space. The hypercube is a dimensionless entity, a standard by which in binary spaces can be measured, analogous to a meter stick.

## Introduction

A three dimensional hypercube representing all of the 4,096 dyadic computations in bivalent systems has been created. There are 16 functions that can compute a dyadic relationship, each component of the dyad, as well as its operator, being a function. These functions are arrayed in a table of functional completeness that reflects a binary counting from 0000 to 1111. Each function in the hypercube has been color keyed as an aid to make any patterns more visible. The hypercube is a new canonization of three-dimensional binary space. At the minimum, the hypercube and the canonization underlying it serve as a "multiplication table" or table of computations and values that shorten the time to do operations that normally would take longer using conventional truth table methods. There are other uses, such as various hypercubes consisting of binary spaces used to compute optimal communications paths ( "hamming distances" ). The main significance of the hypercube at this point is a description of the most fundamental three-dimension space in the binary world and a standard by which there can be a classification and analysis of patterns in binary space, be it randomly generated or from known process. Patterns, or displays of regularity may be produced by a regular process. Patterns emerge from deep innate structures in the universe. The hypercube is a structure created from a known process, and gauging a pattern generated from reputedly random processes against it may be a way of understanding randomness. Currently the hypercube is being presented here for research purposes.

### Construction of binary logical space and functional notation

Zeros and ones and permutations of those as successive quantities present themselves as ordered logical space. These semantics are in keeping with a fundamental aspect of mathematics discussed by Giuseppe Peano in 1898 concerning postulates describing ordering based upon increasing quantity. Peano's Postulates lack a critical postulate concerning a definitive association between succession and

increasing quantity by a regular increment. Yet, the notion of succession, or ordering, exists. Mathematics and logic are co-joined by order based on succession marked by increase, or mounting quantity of binary space, and there is a philosophy underpinning it [1].

Our semantics reflects that philosophy of order. The four rows of permutations of existent relationships yield a 16-column space, the Table of Functional Completeness (TFC). It is called "complete", as all possibilities, or permutations, of 0s and 1s appear for the placeholders p and q. This is generated by the same method as with the above tables – serially and in ascending order (binary counting) in the same manner as the previous tables and in this case from 0000 to 1111, every column being vertically read. Columns are headed by an "f" with subscripts ranging from 0 through 15, each designating a particular function. In computer language, bytes consist of eight bits, and half a byte is a nybble. The TFC consists of 16 columns of nybbles; a function is a nybble. The notation is consistent with that presented by Irving Copi in his *Symbolic Logic* [2]. However, his functions are discussed only in terms of completeness of the binary system. Nothing is written about the nature, philosophy, or the use of the functions as discussed in this paper.

While the TFC includes the p and q generators, or placeholders, they could be omitted, leaving the functions. Philosophically, it can be said that process (a function) is object (result of computation), and object is process. (Notice, also, that in keeping with our ontological commitment of having only two existents, 0 and 1, that stripped of the letters and the function designators, all that remains in the TFC are those 0s and 1s, or only bits) We will see more of this shortly, where a function is an operator, as well as a result of a computation. This makes the logical space an entirely closed and complete space. One function is always the result of two other functions being computed via an operator function. The TFC showing all the permutations of relationships between existents as functions is the following:

| **p** | **q** | $\mathbf{f_0}$ | $\mathbf{f_1}$ | $\mathbf{f_2}$ | $\mathbf{f_3}$ | $\mathbf{f_4}$ | $\mathbf{f_5}$ | $\mathbf{f_6}$ | $\mathbf{f_7}$ | $\mathbf{f_8}$ | $\mathbf{f_9}$ | $\mathbf{f_{10}}$ | $\mathbf{f_{11}}$ | $\mathbf{f_{12}}$ | $\mathbf{f_{13}}$ | $\mathbf{f_{14}}$ | $\mathbf{f_{15}}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

**Table 1 -** Table of Functional Completeness (TFC)

For example, referring to simple existents, row 3 for function 4 ($f_4$), is read, "function 4 relates p=1 to q=0 to yield 0."

With more than two variables, or existents, the TFC is expanded to $2^n$ rows of permuted values, where n=number of variables and $(2^n)^n$ columns in the table. With three variables the display of zeros and ones is simply doubled, where "0" becomes "00", and so forth. With four variables, it becomes "0000".

From the TFC those teaching propositional logic create notational standards for expressing relationships between two existents. Four functions – nybbles - normally are taught: and (&), or (∨), equivalence (≡), and implication (⊃) - but there are 16 functions, and each can be given an operator symbol, viz:

| | |
|---|---|
| $\mathbf{f_0}$ | X - Contradiction |
| $\mathbf{f_1}$ | &, and, conjunction |
| $\mathbf{f_2}$ | >, p is greater than q |
| $\mathbf{f_3}$ | 1>, 1 precedes, or, simply "p" |
| $\mathbf{f_4}$ | <, p is less than q |
| $\mathbf{f_5}$ | >1, 1 follows (or simply "q") |
| $\mathbf{f_6}$ | ≠, p or q is true (1) but not both (XOR);exclusive “or” |
| $\mathbf{f_7}$ | ∨, p or q is true or both are true; inclusive “or”, disjunction |
| $\mathbf{f_8}$ | NOR, neither p nor q or both is/are true |
| $\mathbf{f_9}$ | ≡, p is equivalent to q in truth value |
| $\mathbf{f_{10}}$ | >0, 0 follows (or simply "not q") |
| $\mathbf{f_{11}}$ | ⊂, ← q contains p |
| $\mathbf{f_{12}}$ | 0>, 0 precedes (or simply "not p") |
| $\mathbf{f_{13}}$ | ⊃, or → p contains q (often called “IMP”) – **defines deduction** |
| $\mathbf{f_{14}}$ | NAND, not both p and q are true |
| $\mathbf{f_{15}}$ | T, tautology |

**Table 2 -** Functions, Symbols, and Their Names

Negation (~) is a unary operation, and the TFC implicitly defines it, with functions $f_8$ through $f_{15}$ being opposite or “mirroring”, reflections of $f_0$ through $f_7$. (Although functional completeness is discussed commonly, names often are not given to the functions or standardized. Operator symbols are not standardized; “&” is the same as “^”, “·”, and “and”.)

## Truth tables and the new canonization

Truth tables are in functional form, e.g., 0011 (third column in the TFC) is $f_3$. A calculation like $f_{10}(f_{12}, f_4) \rightarrow f_{11}$) follows the same procedure as with the standard four operators. For example, $f_{10}$ means:

| p | q | p >0 q |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**Table 3 -** Truth Table for $f_{10}$

The resulting value is 1 when 0 is the second value in the relationship. Otherwise, the result is 0. Any function operating over p - ($f_3$), and q - ($f_5$), or the four permutations of 0 and 1, will yield itself. So, the function defines itself, as in $f_{10}(f_3, f_5) \rightarrow f_{10}$ and $f_9(f_3, f_5) \rightarrow f_9$ . The “ → “ will be used interchangeably with “⊃” for typographical convenience. Now, replacing the $f_{12}$ and $f_4$ values for the ones in p and q, p = 1100 and q = 0100, respectively, we have the following:

| p | q | p >0 q |
|---|---|---|
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 0 | 0 | 1 |
| 0 | 0 | 1 |

**Table 4 -** Truth Table for $f_{10}(f_{12,}\, f_4) \rightarrow f_{11}$

To illustrate the rapidity of space saving, the new truth table canonization is:

| p | q | p >0 q |
|---|---|---|
| $f_3$ | $f_5$ | $f_{11}$ |

**Table 5 -** New Truth Table Canonization

The syntax for dyadic computations is $f_n(f_x, f_y) \rightarrow f_{p,}$, where $f_n$ represents a binary operator, such as $f_7$, or 0111. The $f_x$ and $f_y$ represent the operands, and the $f_p$ is the result of the computation. N-adic computations take the form $f_q(f_n(f_x, f_y) \rightarrow f_{p,})) \rightarrow f_r \ldots f_*$ , with the fq and fn being operators. The fp and fr are computational results. Again, any function, as in $f_r$ and f*, can be an operator, or an operand, depending upon its placement in the syntax. Such is one of the factors making binary logical space closed, each function being in dialectical relationship with the others (one in terms of the others), where it can serve in an opposite capacity – operator or operand. Process as an operator becomes the object of an operator (a result of a computation), and object as a function becomes a process. For example, $f_{13}$ is the material implication operator, but it also can be the result of a computation.

Evaluation is standard, working from the innermost parentheses to the outermost. It is optional whether to re-iterate the $f_3$ and $f_5$ underneath the formula being evaluated, as these values already exist in the permutation table. They have been left in to demonstrate that many computations can be done simply by inspecting the function.

To appreciate the space saving nature of the canonization, we have a standard truth table, such as:

| p | q | r | s | (p | & | q) | → | [(r | ≡ | s) | v | (p | → | r)] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Table 6 -** Standard four variable table

rendered as:

| | | | | | 1 | | 3 | | 1 | | 2 | | 1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **p** | **q** | **r** | **s** | **(p** | **&** | **q)** | **→** | **[(r** | **≡** | **s)** | **v** | **(p** | **→** | **r)]** |
| | | | | | $f_1$ | | | | $f_9$ | | $f_7$ | | $f_{13}$ | |
| $f_0$ | $f_0$ | $f_3$ | $f_5$ | $f_0$ | $f_0$ | $f_0$ | $f_{15}$ | $f_3$ | $f_9$ | $f_5$ | $f_{15}$ | $f_0$ | $f_{15}$ | $f_3$ |
| $f_0$ | $f_{15}$ | $f_3$ | $f_5$ | $f_0$ | $f_0$ | $f_{15}$ | $f_{15}$ | $f_3$ | $f_9$ | $f_5$ | $f_{15}$ | $f_0$ | $f_{15}$ | $f_3$ |
| $f_{15}$ | $f_0$ | $f_3$ | $f_5$ | $f_{15}$ | $f_0$ | $f_0$ | $f_{15}$ | $f_3$ | $f_9$ | $f_5$ | $f_{11}$ | $f_{15}$ | $f_3$ | $f_3$ |
| $f_{15}$ | $f_{15}$ | $f_3$ | $f_5$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{11}$ | $f_3$ | $f_9$ | $f_5$ | $f_{11}$ | $f_{15}$ | $f_3$ | $f_3$ |

**Table 7 -** Four variable table in only terms of functional notation

The canonization explanation is published in *The Journal for Systemics and Informatics* [3].

## Composition of the hypercube

The hypercube represents the 4,096 permutations of dyadic (two place) computations of the sixteen functions in the Table of Functional Completeness (TFC), i.e. $f_n(f_x, f_y)$, where fn is a selected operator and the ordered pair x, y as operands. The hypercube is to computational completeness for dyadic relationships as the TFC is to the permutations of 0s and 1s as a four place number. The first is three-dimensional, the second two-dimensional. The hypercube contains the smallest volume that can be occupied in Euclidean space. This fact ensures that the resulting permutation space is optimal. There are 16 plates in the hypercube, each corresponding to one of the 16 functions. Each plate displays a Cartesian coordinate form of a particular function operating over the 16 functions, including itself and shows the complete permutation of computations for a function. There are $16^2$ computations, in each plate or 256 results. In reading the hypercube one starts from the top left of each plate, reads downward and then across the top to arrive at an an answer. A number of function pairs are not commutable, i.e., yield the same result if the functions are switched. Thus, in Plate $f_6$, for $f_6(f_9, f_{12})$ to get $f_5$, read down the left-hand most column to $f_9$ and then across to the column headed by $f_{12}$ in the manner of a distance chart on a highway map to get the $f_5$. The same plate shows $f_6(f_8, f_{11}) = f_3$. The plates are like TFC generation but with ascending functions. Two samples appear in the *Appendix*.

## Computational significance of the hypercube

The hypercube acts as a multiplication table for doing dyadic computations in binary space. Rather than displaying a full truth table one can do a chained calculation, such as $f_{13}(f_7(f_4, f_8), f_1(f_{12}, f_3)), f_9)$ simply by starting with the innermost parentheses, as in standard logical calculations, and working to the outermost function, $f_{13}$. In this case, using the hypercube, the final result is:

$f_7(f_4, f_8) \rightarrow f_{12}$ $f_7$ plate – first set of innermost parentheses

$f_1(f_{12}, f_3) \rightarrow f_0$ $f_1$ plate – second set of innermost parentheses

$f_7(f_{12}, f_0) \rightarrow f_{12}$ $f_7$ plate – $f_7$ function operating over the results of the previous two calculations

$f_{13}(f_{12}, f_9) \rightarrow f_{11}$ $f_{13}$ plate – final calculation.

## Towards pattern recognition in binary space with the hypercube

A pattern is a display of repetition. A pattern can indicate of an ordered process that is emerging and we should be able to determine the nature of the process generating it and predict how the display will appear in the future. Patterns display order, and it is order through the lens of the hypercube as an observing and measuring device that we hope to see what produces that order in a binary space. Binary space may be able to display patterns resulting from proofs and demonstrations, machine language programs, Turing machine programs, and cellular automata, among others. Each is an ordered process, hence potentially being able to exhibit a pattern in the hypercube. Much of the following is speculative but offers a foundation for future research in ascertaining the efficacy of the hypercube as a way of detecting and analyzing regularity in binary spaces.

There are two types of spaces, one purposeful, where we know what generated it, and the second, one resulting from an unknown generator. The question about emerging patterns can be answered in a straight-forward way. Create a number of theorems, or deductive arguments, and generate a truth table to exhibit an arrangement of 0s and 1s, some or all of which may exhibit a regularity. Theorems are structures, some of which are of thinking itself. They may represent types of abstraction. Certain modes of thinking present certain types of binary spaces. It may be the case that two or more theorems present the same arrangement. Each arrangement can be classified, as Wolfram has done with his cellular automata [4]. Once a "dictionary" of pattern types resulting from these theorems is produced, we can use it to compare regularity generated randomly or from unknown processes. Theorems can be created by the methods of proof, conditional proof, short truth tables, or by the hypercube. We see may a pattern in the hypercube or that the cube can analyze it. If so, then, we will be on the way of discovering the origin of patterns in a reputedly random space.

## Tracing the origin of a valid relationship using the hypercube

We ask what ultimate two functions will give tautology ($f_{15}$) as a result for a theorem's corresponding conditional, that is, $f_n(f_p, f_q) \rightarrow f_{15}$. We go to Plate $f_{13}$ and look inside the plate for the $f_{15}$ s. Each $f_{15}$ is the result of one function, or conjunct of them implying another to give a valid relationship, or theorem. Then we determine what two functions $f_{13}$ operate over to give the $f_{15}$, and there are many. We see $f_{15}(f_0, f_{0)}$ as the first. Remember, read the intersection of the first row and the first column to see the $f_0$, just like a distance finder between two cities on a road map. Thus, $f_{13}(f_0, f_0) \rightarrow f_{15}$ . Similarly, we see all across the $f_0$ row a series of $f_{15}$ s, i.e., $f_{13}(f_0, f_n) \rightarrow f_{15}$ or $f_{13}(f_n, f_n) \rightarrow f_{15}$, for that matter, n being any function. For the $f_1$ row, every other function yields $f_{15}$, as in $f_{13}(f_1, f_1) \rightarrow f_{15}$ and $f_{13}(f_1, f_3) \rightarrow f_{15}$. Another example is $f_{13}(f_6, f_{14}) \rightarrow f_{15}$. Now, we ask what functions can produce each of the functions used by $f_{13}$ to produce $f_{15}$. Such an exercise is rather extensive, but as has been demonstrated, the hypercube shortens this research considerably, as only a look-up is required. For example, we have $f_{13}(f_8, f_{12}) \rightarrow f_{15}$ (in fact $f_8 - f_{15}$). Then, how we can get an $f_8$? Just about any of the hypercube plates will tell us quickly. Taking one at random, let us say $f_4$, we can locate many, such as $f_4(f_7, f_{8\text{-}15}) \rightarrow f_8$. Plate $f_9$, as another example, shows that $f_9(f_{13}, f_{10}) \rightarrow f_8$. Hence, the task of determining exactly what computational result gave rise to a function is not possible, because of their being at least two paths to the same result. One should realize, however, that theorems can be created by such backtracking in the hypercube, just so long as one preserves the relationships set forth in the corresponding conditional. Here, in addition to the the long and short truth tables method, the hypercube serves as a useful short-hand tool for creating theorems. It also can act as a theorem prover. Simply backtrack by inspection from a result to see if what reputedly gave rise to it actually did. A worthy research project would be to create a computer program to generate theorems from the hypercube, classify them according to emerging patterns or methods of generation, and create the dictionary alluded to earlier.

## The Hypercube as foundational for pattern analysis

To date, there has been no method to find exactly all the functions generating a function or a space, save for a nearest neighbor analysis of cellular automata using a Turing-style tape of 0s and 1s [5]. Contrary to this, the analyzable spaces here are not necessarily produced by using a cellular automaton or neighborhood space method. The 0 and 1values may be inserted from anywhere by any method, thus adding force to the term “raw space”.

We are not trying to say what a pattern represents, such as a face or language character but merely that a regularity exists, and we should ask what gave rise to it. The whole theory underlying this paper is that all binary patterns have their origin in some place in the hypercube, as the hypercube is the most fundamental 3-D building block of binary logic, given what the above discussion about theorem generation illustrates. From that origin, there is built from various functional relations expressed by the cube the emerging pattern. Patterns don't emerge from nowhere. There is an origin and the progression is orderly. For example, in examining the hypercube closely, it may be asked whether certain groupings, such as the assemblage of $f_{15}$s on plate $f_{11}$ constitute a core, or a “seed” for “gliders”, the image so popular in discussing cellular automatons [6]. Aside from any reference to order emerging from chaos in the binary world or whether there are Lorentz attractors, there are already in each of the sixteen plates emerging patterns of color, as with Plate $f_7$, where, with $f_7$, there is a string of $f_7$s, followed by eight $f_{15}$s, both vertically and horizontally, suggesting boundary conditions of some type. The diagonals in each plate show a functional counting, one diagonal ascending, the other descending. By coupling all the plates together, a three-dimensional view may reveal more. Already can be found emerging grouping, or clustering, of functions, as in Plate $f_4$, where $f_2$ seems to congregate in groups of three at various places along the top of the diagonal. The hypercube also may contain patterns classified in the dictionary of theorems mentioned above.

We also question whether there is a family of patterns or an algebra of spaces. For example, there is $f_{13}(f_n, f_{15}) \rightarrow f_{15}$, $f_{13}(f_n, f_n) \rightarrow f_{15}$, and $f_{13}(f_0, f_n) \rightarrow f_{15}$, where n is any function, and from this we may say that $f_7[f_{13}(f_n, f_{15}), f_{13}(f_0, f_n)] \rightarrow f_{15}$, and list equivalent relationships with appropriate substitutions. Theorems are algebraic expressions.

When we see a space for the first time, it can be designated as a “raw binary space”, meaning that if we discern any regularity of the values, we may not know what generated them. Let us say a pattern has been generated “randomly”. One may argue that there is “emergence”, but of what? Patterns displayed resulting from allegedly autopoietic, or self-organizing processes present a challenge of determining what the organizing principle is. For any regularity in binary space, we should ask not only what generated it but what its significance is. In Wolfram's automata, what do any of the patterns generated by the automatons mean, save for the design being associated with a rule? Rules are specific thought patterns, but is there a more general observation about thinking we can make? Later, we will touch on this subject in discussing a correlation between binary spaces and activity in brain structures. We won't analyze patterns (as many ways exist for doing so [7]) but suggest a way of normalizing a space and propose the hypercube as a measuring device for the raw spaces. To work with patterns in binary space, one can consider a way of making the hypercube a measurement and classification device. Raw spaces should be normalized, such as padding sets of bits less than a nybble (four bits) either before or after with 0s, as spaces can be only of complete functions. One compares the normalized space using the “dictionary of theorems”, mentioned above. A second means of classifying spaces is with the 3-D hypercube; what in that “raw space” looks like the regularity inside any of the plates in the hypercube? The hypercube is a baseline against which to measure deviation of any set of blocks of binary space.

## A system without axioms

To this point we have not identified any axioms but yet have been able to generate theorems by the definition of deduction, truth table methods, and corresponding conditional rule. All are based on the rule that a statement must be derivable from the previous, and that derivation can be found through inspection by using the hypercube. This means that in terms of zeros and ones, the “and”, as well as “material implication” operator/function a deductive relationship holds. Validation that a statement follows from the previous is done by inspection of the hypercube. This means that our system is without axioms, or naturally and intuitively deductive.

All of the functions emerge from juxtaposition of numbers in an ascending fashion. The table of logical space is set, and the functions are only naming devices for particular sets of 0s and 1s in that space. It is not necessary to use axioms to derive the functions. Only an ordering principle is needed, coupled with the primitives, definitions, and so forth. It is proposed that Peano's Postulates form the basis of such a system, plus a postulate that asserts mounting quantity based succession. A start would be focusing on the concept of number based on a fundamental Cartesian cut.

## Philosophical significance of the binary structures

The two-valued, or binary, system is foundational in deduction, as it uses the lowest number of variables possible to construct a system of relations. The simple observation, other than of the whole, is of two. This observation stems from an extreme division of any object in the three dimensional universe; ultimately, it will be reduced to the smallest of the smallest, or Planck volume in terms of not-Planck volume, or vacuum space. Of course this sub-quantum world must be apprehended in terms of the whole in order to place matters in proper perspective. The three dimensional world is syntactically binary [8].

This author's assertion is that logic is a language that describes innate order in the 3-D universe and that it is the basis upon which mathematics rests. Logic is discovered, rather than invented; "...a machinery for the combination of yes-no or true-false elements does not have to be invented. It already exists [9]." Jean Piaget asserts, “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 ... .[10]”. This idea is not new, as it extends as far back as 5000 years ago in South Asia [11], and with the Chinese 4,000 years ago and the I Ching. Leibniz [12], the first modern scientist to formalize the binary arithmetic in 1703 wrote of it.

When one attempts Cartesian subdivision at the quantum level, the world of uncertainty is met, and one cannot measure position, except statistically. Yet, computationally, we can go to the Planck scale, where all symmetries are broken. To divide something beyond a Planck volume would require more energy than exists in the universe. What exists in terms of not-Planck volume is vacuum space, which is penetrated periodically with energy fluctuations. At the Planck scale, nothing is discrete, so one has to identify what makes a description of reality binary [13]. We certainly do not know what is at the Planck scale level, as this degree of granularity is theoretical and merely computational. At the Planck scale, the very nature of the binary world is transformed, where bivalency transforms to a four dimensional world and a four-valued system [14]. This system is the superset of the three-dimensional bivalent system.

The unfolding structure of the most basic logic in the three dimensional world - binary relationships - comes from natural ordering [15]. Logical space is generated in an ascending fashion and ultimately contains all the relationships possible in this world. Recall, everything is reducible to Planck volume

and non-Planck volume, a duality. From the singular and planar logical spaces comes the three dimensional hypercube.

## Future direction in research

Much of what follows is speculative, but it was stated at the outset in this paper that the hypercube is being presented for research purposes. Aside from lofty considerations of the quantum world [16], the hypercube has more prosaic applications such as an algebra of spaces. Consider commutativity as error checking device. Commutativity is symmetric, where $f_C(fp, fq)=f_C(fq, fp)$. This goes for C=0, 1, 6, 7, 8, 9, 14, and 15. Functional computation may be expressed in algebraic form, such as the simple example in the $f_5$ plate, where $f_5(fn, fp) \rightarrow fp$, and where fn and fp represent any two distinct functions. Similarly, $f_3(fn, fp) \rightarrow fn$ exists for the $f_3$ plate. For plate $f_8(f_{15}, fq) = f_0$ and $f_8(fp,f_{15}) = f0$, so $f_8(f_{15},fq) = f_8(fp,f_{15})$. Numerous and more complicated relations may be developed, but such is work for further research.

We have seen where the new canonization with the functional notation can result in an algebra of functions to generate inference and equivalence rules. For example, *modus ponens* is $p \supset q, p, \therefore q$. In our canonization, this is $f_{13}$(a conjunct of functions resulting from others) $\rightarrow$ $fc_{,}$ (a conclusion, or derived function) where $f_{13}$ must result in $f_{15}$, or tautology. Research might produce a computer program to generate not only acceptable rules, but these might be used to help produce an algebra of spaces. Already, we have seen where the hypercube has aided us in finding theorems using the corresponding conditional and a computer program can be written to do this.

We said above that each of the 16 functions is recursive, i.e., the outputs forward fed as inputs into the function cause the function to reappear [17]. Thus, each function acts as a self-maintaining, or homeostatic, automaton. Of course, all binary spaces are composed of one or more of these functions, or partial functions (less than a nybble). Starting with a set of set of formulas demarcating an initial space, it would be interesting to see how that space evolves until it repeats itself. No entity at whatever level is static, so tracing the dynamism of an initial state of binary functions would give an insight to pattern generation and possibly shed light on how basins of attraction form.

As a longer term project, one may map each function to a sound or color to see what patterns may emerge. Newton, following an idea by the ancient Greeks, suggested that there may be a correlation between color and sound [18]. Correlating sound to color is not novel these days [19]. In various computer programs designed to play CDs, such as Windows Media, one can view colored patterns emerge when playing music.

Consciousness studies can be explored with processes applied to binary spaces. According to Tononi, consciousness is integrated information, "...the amount of information generated by a complex of elements, above and beyond the information generated by its parts." [20]. Consciousness arises from the condition of neural systems, and these can be represented in a binary manner, i.e., on-off switches, or as Tononi refers says, "photodiodes" [21]. Of course, to represent anything approaching what people think is consciousness would involves enormous complexity, as Tononi admits, but his serves a model for research. Perhaps the the 3-D hypercube developed in this paper could be overlaid on to the binary space generated by Tononi's model, much in the same manner as discussed earlier with respect to binary spaces in general. The theorems and their corresponding patterns generated by the hypercube might have neural correlates and such would involve Tononi's research. This world is just beginning.

## Conclusion and outlook

The most basic binary logical space is generated from a single square to two squares, one containing a value and the other a second value. The permutations of this two-squared space yield four permutations of the two values. These, in turn, produce the sixteen basic functions displayed in the Table of Functional Completeness (TFC). From the TFC is developed the three-dimensional hypercube. Functions also are results of computations and vice versa. A new canonization has been presented that allows for a simplified way of computing dyadic relationships, as well as traditional truth tables. The hypercube is color coded to help display the relationships of functions to each other and identify patterns. At the outset the hypercube can be used as a look-up table to yield the results of any dyadic computation involving any of the 16 functions.

The hypercube allows for more rapid and simplified dyadic computations in bivalent space, but also may enable enhanced methods for computing hamming distances. There are indications that Lorenz attractors may exist within the hypercube, these possibly indicating seeds from which order is generated from what was thought previously to be chaos. Something (a pattern) doesn't come from nothing. The universe at the third dimension has innate order, described by binary structures, the hypercube being one. While there is evidence of randomness (inability to predict), such as Brownian movement and pi ($\pi$), there is an innate order in the universe, and chaos contains encoded order that can be untangled by logical analysis.

Numerous research areas stem from the development of the basic hypercube, all centering on pattern analysis of structures expressed in binary space. Once an undefined binary space is mapped onto the ordered one, the same analytical process of pattern recognition can be applied, thus leading to a uniform way of looking at reality in many of its diverse but reducible forms.

## References (all web links accessed 27 March 2011)


[1] J. Horne, "The General Theory and Method of Binary Logical Operations", **Journal of Applied Science and Computations,** 3 December 2000.

[2] I. Copi, **Symbolic Logic**, 5th ed., New York: Macmillan, 1979, pp. 220-222.

[3] J. Horne, "Recursion Of Binary Space As a Foundation Of Repeatable Programs", ***The Journal of Informatics and Systemics***, Vol. 4, No. 5, 2006.

[4] S. Wolfram, **A New Kind of Science**, http://wolframscience.com/nksonline/toc.html.

[5] A.Wuensche, "The Ghost in the Machine: Basins of Attraction of Random Boolean Networks, University of Sussex at Brighton: **Cognitive Science Research Papers***,* CSRP 281, 1993, https://docs.google.com/viewer?url=http://www.informatics.sussex.ac.uk/users/andywu/downloads/papers/ghost_in_machine.pdf&embedded=true&chrome=true .

[6] Wolfram, *passim*.

[7] Hamann, "*Definition and Behavior of Langton's Ant in Three Dimensions", ,* http://www.springerlink.com/content/m75033714447106g/, http://videolectures.net/kdd09_ye_mdpbmf/, and http://www.exploringbinary.com/what-a-binary-

counter-looks-and-sounds-like/ .

[8] J. Horne, “Is Reality Digital or Analog?”, 2011 **FXQI Essay**, http://www.fqxi.org/community/forum/topic/862 .

[9] C.W. Misner , K.S. Thorne, J.A. Wheeler **Gravitation,** New York: W.H. Freeman and Company, 1973.

[10] J. Piaget, **Logic and Psychology**, New York: Basic Books, Inc.; 1958.

[11] **Rig Veda**, Hymn CXXIX – **Creation**, Book the Tenth – **Oxford Companion to Philosophy**, http://www.answers.com/topic/ancient-philosophy#Indian_philosophy.

[12] Leibniz, **Explication de l'Arithmétique Binaire**, https://docs.google.com/viewer?url=http://ads.ccsd.cnrs.fr/docs/00/10/47/81/PDF/p85_89_vol3483m.pdf&embedded=true&chrome=true, 1703.

[13] J. Horne, “Is Reality Digital or Analog?”, 2011 **FXQI Essay**, http://www.fqxi.org/community/forum/topic/862 .

[14] A. Stern, A., **Matrix Logic,** Amsterdam: North-Holland, 1988.

[15] J. Horne, **Logic as the Language of Innate Order in Consciousness,**. Informatica, 1997; 21: 675-682.

[16] D. Rapoport, **Surmounting the Cartesian Cut Through Philosophy, Physics, Logic, Cybernetics, and Geometry:Self-reference, Torsion, the Klein Bottle, the TimeOperator, Multivalued Logics and Quantum Mechanics,** Found Phys, 22 August 2009, A. Stern, A., **Matrix Logic,** Amsterdam: North-Holland, 1988.

[17] J. Horne, “Recursion of Logical Operators and Regeneration of Discrete Binary Space”, **Informatica,** 2000;24:275-279.

[18] Newton I. **Opticks, or, a Treatise of the Reflections, Refractions, Inflections, and Colours of Light**. London:Printed for William Innys at the West-End of St. Paul's, 1730.

[19] For example, see http://www.jacmuse.com/artisticconcepts/newpage14.htm, http://www.harpcenter.com/page/SWHC/PROD/BestMeditCD/3212R.

[20] G.B. Tononi, “Consciousness as Integrated Information: a Provisional Manifesto,;Bull. 215**:** 216-242, 2008.

[21] Ibid.

The author wishes to thank Asia Flood, a professional artist located in Gold Canyon, AZ for her invaluable assistance in color coding the hypercube. Without her work, it would have been difficult, if not impossible to discern the colored patterns.

# Appendix

## Sample Plate of The Three-dimensional Hypercube

### $f_1$ - AND – conjunction p & q

| $f_1$ | $f_0$ | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ | $f_{12}$ | $f_{13}$ | $f_{14}$ | $f_{15}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $f_0$ | $f_0$ | $f_0$ | $f_0$ | $f_0$ | $f_0$ | $f_0$ | $f_0$ | $f_0$ | $f_0$ | $f_0$ | $f_0$ | $f_0$ | $f_0$ | $f_0$ | $f_0$ | $f_0$ |
| $f_1$ | $f_0$ | $f_1$ | $f_0$ | $f_1$ | $f_0$ | $f_1$ | $f_0$ | $f_1$ | $f_0$ | $f_1$ | $f_0$ | $f_1$ | $f_0$ | $f_1$ | $f_0$ | $f_1$ |
| $f_2$ | $f_0$ | $f_0$ | $f_2$ | $f_2$ | $f_0$ | $f_0$ | $f_2$ | $f_2$ | $f_0$ | $f_0$ | $f_2$ | $f_2$ | $f_0$ | $f_0$ | $f_2$ | $f_2$ |
| $f_3$ | $f_0$ | $f_1$ | $f_2$ | $f_3$ | $f_0$ | $f_1$ | $f_2$ | $f_3$ | $f_0$ | $f_1$ | $f_2$ | $f_3$ | $f_0$ | $f_1$ | $f_2$ | $f_3$ |
| $f_4$ | $f_0$ | $f_0$ | $f_0$ | $f_0$ | $f_4$ | $f_4$ | $f_4$ | $f_4$ | $f_0$ | $f_0$ | $f_0$ | $f_0$ | $f_4$ | $f_4$ | $f_4$ | $f_4$ |
| $f_5$ | $f_0$ | $f_1$ | $f_0$ | $f_1$ | $f_4$ | $f_5$ | $f_4$ | $f_5$ | $f_0$ | $f_1$ | $f_0$ | $f_1$ | $f_4$ | $f_5$ | $f_4$ | $f_5$ |
| $f_6$ | $f_0$ | $f_0$ | $f_2$ | $f_2$ | $f_4$ | $f_4$ | $f_6$ | $f_6$ | $f_0$ | $f_0$ | $f_2$ | $f_2$ | $f_4$ | $f_4$ | $f_6$ | $f_6$ |
| $f_7$ | $f_0$ | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_0$ | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ |
| $f_8$ | $f_0$ | $f_0$ | $f_0$ | $f_0$ | $f_0$ | $f_0$ | $f_0$ | $f_0$ | $f_8$ | $f_0$ | $f_8$ | $f_8$ | $f_8$ | $f_8$ | $f_8$ | $f_8$ |
| $f_9$ | $f_0$ | $f_1$ | $f_0$ | $f_1$ | $f_0$ | $f_1$ | $f_0$ | $f_1$ | $f_0$ | $f_9$ | $f_8$ | $f_9$ | $f_8$ | $f_9$ | $f_8$ | $f_9$ |
| $f_{10}$ | $f_0$ | $f_0$ | $f_2$ | $f_2$ | $f_0$ | $f_0$ | $f_2$ | $f_2$ | $f_8$ | $f_8$ | $f_{10}$ | $f_{10}$ | $f_8$ | $f_8$ | $f_{10}$ | $f_{10}$ |
| $f_{11}$ | $f_0$ | $f_1$ | $f_2$ | $f_3$ | $f_0$ | $f_1$ | $f_2$ | $f_3$ | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ |
| $f_{12}$ | $f_0$ | $f_0$ | $f_0$ | $f_0$ | $f_4$ | $f_4$ | $f_4$ | $f_4$ | $f_8$ | $f_8$ | $f_8$ | $f_8$ | $f_{12}$ | $f_{12}$ | $f_{12}$ | $f_{12}$ |
| $f_{13}$ | $f_0$ | $f_1$ | $f_0$ | $f_1$ | $f_4$ | $f_5$ | $f_4$ | $f_5$ | $f_8$ | $f_9$ | $f_8$ | $f_9$ | $f_{12}$ | $f_{13}$ | $f_{12}$ | $f_{13}$ |
| $f_{14}$ | $f_0$ | $f_0$ | $f_2$ | $f_2$ | $f_4$ | $f_4$ | $f_6$ | $f_6$ | $f_8$ | $f_8$ | $f_{10}$ | $f_{10}$ | $f_{12}$ | $f_{12}$ | $f_{14}$ | $f_{14}$ |
| $f_{15}$ | $f_0$ | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ | $f_{12}$ | $f_{13}$ | $f_{14}$ | $f_{15}$ |

### $f_{13}$ - ⊃, p contains q, p ⊃ q, – defines deduction

| $f_{13}$ | $f_0$ | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ | $f_{12}$ | $f_{13}$ | $f_{14}$ | $f_{15}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $f_0$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ |
| $f_1$ | $f_{14}$ | $f_{15}$ | $f_{14}$ | $f_{15}$ | $f_{14}$ | $f_{15}$ | $f_{14}$ | $f_{15}$ | $f_{14}$ | $f_{15}$ | $f_{14}$ | $f_{15}$ | $f_{14}$ | $f_{15}$ | $f_{14}$ | $f_{15}$ |
| $f_2$ | $f_{13}$ | $f_{13}$ | $f_{15}$ | $f_{15}$ | $f_{13}$ | $f_{13}$ | $f_{15}$ | $f_{15}$ | $f_{13}$ | $f_{13}$ | $f_{15}$ | $f_{15}$ | $f_{13}$ | $f_{13}$ | $f_{15}$ | $f_{15}$ |
| $f_3$ | $f_{12}$ | $f_{13}$ | $f_{14}$ | $f_{15}$ | $f_{12}$ | $f_{13}$ | $f_{14}$ | $f_{15}$ | $f_{12}$ | $f_{13}$ | $f_{14}$ | $f_{15}$ | $f_{12}$ | $f_{13}$ | $f_{14}$ | $f_{15}$ |
| $f_4$ | $f_{11}$ | $f_{11}$ | $f_{11}$ | $f_{11}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{11}$ | $f_{11}$ | $f_{11}$ | $f_{11}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ |
| $f_5$ | $f_{10}$ | $f_{11}$ | $f_{10}$ | $f_{11}$ | $f_{14}$ | $f_{15}$ | $f_{14}$ | $f_{15}$ | $f_{10}$ | $f_{11}$ | $f_{10}$ | $f_{11}$ | $f_{14}$ | $f_{15}$ | $f_{14}$ | $f_{15}$ |
| $f_6$ | $f_9$ | $f_9$ | $f_{11}$ | $f_{11}$ | $f_{13}$ | $f_{13}$ | $f_{15}$ | $f_{15}$ | $f_9$ | $f_9$ | $f_{11}$ | $f_{11}$ | $f_{13}$ | $f_{13}$ | $f_{15}$ | $f_{15}$ |
| $f_7$ | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ | $f_{12}$ | $f_{13}$ | $f_{14}$ | $f_{15}$ | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ | $f_{12}$ | $f_{13}$ | $f_{14}$ | $f_{15}$ |
| $f_8$ | $f_7$ | $f_7$ | $f_7$ | $f_7$ | $f_7$ | $f_7$ | $f_7$ | $f_7$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ |
| $f_9$ | $f_6$ | $f_7$ | $f_6$ | $f_7$ | $f_6$ | $f_7$ | $f_6$ | $f_7$ | $f_{14}$ | $f_{15}$ | $f_{14}$ | $f_{15}$ | $f_{14}$ | $f_{15}$ | $f_{14}$ | $f_{15}$ |
| $f_{10}$ | $f_5$ | $f_5$ | $f_7$ | $f_7$ | $f_5$ | $f_5$ | $f_7$ | $f_7$ | $f_{13}$ | $f_{13}$ | $f_{15}$ | $f_{15}$ | $f_{13}$ | $f_{13}$ | $f_{15}$ | $f_{15}$ |
| $f_{11}$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_{12}$ | $f_{13}$ | $f_{14}$ | $f_{15}$ | $f_{12}$ | $f_{13}$ | $f_{14}$ | $f_{15}$ |
| $f_{12}$ | $f_3$ | $f_3$ | $f_3$ | $f_3$ | $f_7$ | $f_7$ | $f_7$ | $f_7$ | $f_{11}$ | $f_{11}$ | $f_{11}$ | $f_{11}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ | $f_{15}$ |
| $f_{13}$ | $f_2$ | $f_3$ | $f_2$ | $f_3$ | $f_6$ | $f_7$ | $f_6$ | $f_7$ | $f_{10}$ | $f_{11}$ | $f_{10}$ | $f_{11}$ | $f_{14}$ | $f_{15}$ | $f_{14}$ | $f_{15}$ |
| $f_{14}$ | $f_1$ | $f_1$ | $f_3$ | $f_3$ | $f_5$ | $f_5$ | $f_7$ | $f_7$ | $f_9$ | $f_9$ | $f_{11}$ | $f_{11}$ | $f_{13}$ | $f_{13}$ | $f_{15}$ | $f_{15}$ |
| $f_{15}$ | $f_0$ | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ | $f_{12}$ | $f_{13}$ | $f_{14}$ | $f_{15}$ |