INDIAN OCEAN REVIEW OF SCIENCE AND TECHNOLOGY
ISSN: 2312-1874, Article 2-2015

Irreducible Statements and Bases for Finite–Dimensional Logical Spaces

Gennaro Auletta[1]

Abstract
Logic can be shown to be a branch of the combinatorial calculus. In this way we can build logical spaces on the basis of a Boolean algebra. This allows us to pick up a finite number of irreducible atomic variables for each n–dimensional space. These variables have a characteristic binary ID being the neg–reversal of themselves. A theorem is proved showing that their number must be finite. Moreover, a second theorem gives us the algorithm for building sets of generators of the space. Finally, the algorithm for computing how many alternative bases there are for any n–dimensional logical space is provided.

Keywords
Logical space — reversal operator — neg–reversal operator — atomic variable — spanning sets

[1] *University of Cassino, Italy; email: gennaro.auletta@gmail.com*

Contents

1 **Introduction** 1
2 **Irreducible sets of objects** 2
3 **Spanning spaces Logically** 5
4 **Alternative Sets and Bases** 7
5 **Symmetrization** 9
6 **Results** 9
7 **Discussion** 10
References 15

1. Introduction

Logic can be treated as a pure combinatorial calculus [1]. To this purpose, I shall introduce the notion of logical space, which can be conceived of to be an extension of the notion of Boolean algebra [5], as far as it can be considered as a vectorial space and the basic variables that span it as a kind of logical basis. This may turn out to be especially relevant for dealing with quantum computation, as far as quantum systems understood as information processors display a combinatory of possibilities [2]. The dimension of the logical space is determined by the number of those basic variables necessary and sufficient to logically span it. Variables are taken to represent sets of objects but could also be taken to represent statements; "composite" formula (expressing relations among those variables) can be taken as collections of objects (classes) or statements. This approach fits well with Category theory [4][10]. In the following, I shall denote variables with capital letters. Any collection of objects that is generated in such a way (expressing relations among those basic variables) is represented by a dot in the space (or as a 0-D subspace). Each one–dimensional (1D) space or subspace is constituted by a line connecting a variable and its negation (tautology and contradiction are extremal points and center, respectively, of this line) that we can represent as a kind of vector (where negation is represented as the opposite direction). Each 2D space or subspace is a surface (embedded in a a circle, as we shall see) constituted by all connections among the variables of two different 1D spaces or subspaces, and this can be represented in Cartesian plane, and so on. There are some fundamental numbers to consider that characterize any logical space:

- n is the number of basic variables that span the space and determine the dimension of the space.
- $m = 2^n$ is the number of truth–value assignments that determine the truth–table. In any space, all collections of objects are represented by a sequence of m 0s an 1s representing falsity and truth, respectively: the basic variables spanning an n–dimensional space have m "slots" that can be filled with 0s or 1s. I call any of these Boolean bitstrings *the binary ID* of the collection of objects in short.
- $k = 2^m$ is the overall number of collections of objects that can be generated in such a space through relations among the n variables.

Let us now introduce the basic operations. I shall make use of the usual OR, AND, and NOT. In the following, OR is represented by +, AND by absence of symbol, negation by the ordinary symbol for set complementation (X' is the negation of X). For instance, XY denotes X AND Y while $X' + Y$ denotes NOT X OR Y. Negation exchanges the 0s and 1s in a binary ID. In the 4D space (where $m = 16$, see Tab. 5) the collection of objects XY is identified by the binary ID 0000000000001111, and its negation, i.e. $(XY)' = X' + Y'$ is identified by 1111111111110000. The reader may easily check this by summing (or multiplying) binary IDs column by column, where the outputs are shown in Tab. 1. Note that I have added also the operations of subtraction (AND NOT) and division (OR NOT).

Now, I introduce two additional logical operators first formulated in [8]. The first one can be called *reversal operator* and corresponds to inversion in the theory of groups: as we shall see, the identity elements are 0 and 1 [10, Sec. 3.2]. The reversal operator transforms e.g. XY into $Y'X'$ (or $X'Y'$, due to the commutativity of AND) or $X+Y$ into $Y'+X'$ (or $X'+Y'$, due to the commutativity of OR). Let us write e.g. the first transformation as $(XY)^{-1} = Y'X'$. It owes its name to the fact that the effect of the reversal operator is to reverse the binary ID For instance, recalling that the binary ID of XY in the 4D space is 0000000000001111, the binary ID of $(XY)^{-1}$ is 1111000000000000. Note that

$$(XY)(XY)^{-1} = \mathbf{0} \quad \text{and} \quad (X+Y)+(X+Y)^{-1} = \mathbf{1}, \quad (1)$$

where **0** and **1** are contradiction and tautology represented by a string of 0s and 1s, respectively.

Another important operation is performed by the *neg–reversal operator*, that is an operation that is both negation and reversal (it does not matter in which order the two operations are executed). In such a case, we have $(XY)^{\dagger} = Y+X$, where the dag symbolizes this operator. The action of the neg–reversal operator on a binary ID is the following: it transforms e.g. 0000000000001111 into 0000111111111111. Note that

$$(XY)(XY)^{\dagger} = XY \quad \text{and} \quad (XY)+(XY)^{\dagger} = X+Y. \quad (2)$$

Both reversal and neg-reversal operations (as well as negations) are endomorphisms.

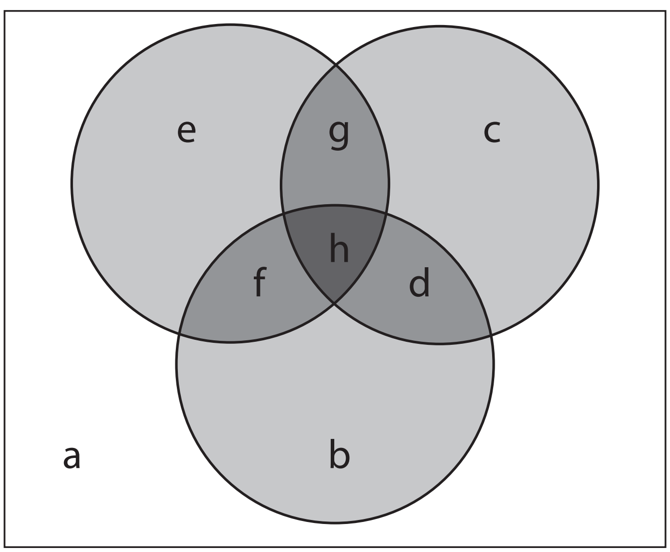

Figure 1. Venn diagram for the 3D logical space. Note that the grayscale is arranged according to the overlaps among areas: white (0% of black) is empty area, pale gray (25% of black) is no overlap, middle gray (50% of black) two sets overlap, dark gray (75% of black) three sets overlap. A convention of this kind is maintained also in the following.

2. Irreducible sets of objects

It is interesting to note that in any n–dimensional space there is a set of Collections of objects (or statements) for which negation and reversal coincide. These collections of objects are characterized by a binary ID that is divided in two halves whose the second half is the neg–reversal of the first one, what means that any of these expressions is the neg–reverse of itself, e.g., $X^{\dagger} = X$, whose binary ID in the 4D space can be taken to be 0000000011111111 (see again Tab. 5). Another way to say this is that for those variables we have $X^{-1} = X'$. In fact,

$$X = X^{\dagger} = (X')^{-1}, \quad \text{and therefore} \quad X^{-1} = X'. \quad (3)$$

Note that the application of the general endomorphic neg–reversal operation on these variables is in fact an automorphism. This can be taken as a definition of these variables. Such variables are irreducible sets (or statements). To show this, let us consider the 3D space (with $k = 256$). In this case, a suitable truth–value table is displayed in Tab. 2 (see also Fig. 1). Note that n (or $2n$, depending on whether we consider negations or not) represents the number of rows and m the number of columns. Therefore, as mentioned, in the 3D space there are $2^4 = 16$ variables that are the neg–reversal of themselves (in short, neg–reversal variables), where on each row there is a variable and its negation (which is generated by the corresponding variable by either negating or reversing it), as displayed in Tab. 3. Note that the truth–values assignment of Tab. 2 for X, Y, Z corresponds to the collections of objects 16, 13, 11, respectively.

Let us now come back to issue of irreducibility. First, note that any of the neg–reversal variables can be expanded in terms of the other similar variables following the same algorithm. For instance, in the 3D space, we can have the (among other possible) expansions of the 16 neg–reversal variables of Tab. 4. Note that transformations A/D and D/A can be considered as the inverse of each other, and this also true for couples $Y-C$, $Z-B$, $X-E$. In fact, it turns out that reiterating the same transformation, we get back the initial variable. For instance:

$$\begin{aligned} X &= E'(BC)'(B'C')' + BC = E'(B'C + BC') + BC \\ &= XYZ' + XY'Z + XY'Z' + XYZ \\ &= X(YZ' + Y'Z + Y'Z' + YZ) \\ &= X. \end{aligned} \quad (4)$$

This shows that the expansions displayed in the previous table are in fact resolutions of identity. In other words, *any* transformation of a neg–reversal variable giving rise to another neg–reversal variable, if reiterated, gives the former variable back. Thus, neg–reversal variables display a characteristic recursivity. This can be stated in a Lemma:

Lemma 2.1 *Only neg-reversal variables have such a property that makes of them sets that cannot be reduced to collections of other sets in the space they occupy and the only ones in that space.*

Proof: In fact, for a variable to be irreducible in this sense is an immediate consequence of its neg–reversibility (or also

Sum	$0+0=0$	$0+1=1$	$1+0=1$	$1+1=1$
Product	$0\times 0=0$	$0\times 1=0$	$1\times 0=0$	$1\times 1=1$
Subtraction	$0-0=0$	$0-1=0$	$1-0=1$	$1-1=0$
Division	$0:0=1$	$0:1=0$	$1:0=1$	$1:1=1$

Table 1. The four basic logical operations.

self–duality). It may be noted that the transformations of Tab. 4 and their reversals are the analogous of the unitary Hadamard transformations which express reversibility under reiteration (they are the reversal of themselves) [3, Sec. 17.7].

In other words, the set $NR(n)$ of the neg–reversal variables gives rise to an endomorphism such that the following diagram (for the 3D case) totally commutes:

$$\begin{array}{ccccc} X & \xrightarrow{F(X,Y)} & Y & \xrightarrow{F(Y,Z)} & Z \\ {\scriptstyle H(X,A)}\big\downarrow & & {\scriptstyle H(Y,B)}\big\downarrow & & \big\downarrow{\scriptstyle H(Z,C)} \\ A & \xrightarrow[F(A,B)]{} & B & \xrightarrow[F(B,C)]{} & C \end{array}$$

Thus, for all $X,A\in NR(n)$, variables $Y,Z,B,C\in NR(n)$ exist such that we have the Hadamard morphisms

$$H(X,A):X\longrightarrow A,\quad H(Y,B):Y\longrightarrow B,\quad H(Z,C):Z\longrightarrow C,$$

and the four F morphisms having the form

$$F(X,Y):X\longrightarrow Y:=X\longrightarrow XY+Y=Y.$$

Note that the inverse morphism $F^{-1}(X,Y)$ applied to Y gives $XY+X=X$. These two transformations can be generalized to any nD spaces as shown in the diagram (5). ■

$$\begin{array}{cccccccccc} A_1 & \xrightarrow{F(A_1,A_2)} & A_2 & \xrightarrow{F(A_2,A_3)} & A_3 & & \cdots & & A_{n-1} & \xrightarrow{F(A_{n-1},A_n)} & A_n \\ {\scriptstyle H(A_1,B_1)}\big\downarrow & & {\scriptstyle H(A_2,B_2)}\big\downarrow & & {\scriptstyle H(A_3,B_3)}\big\downarrow & & & & \big\downarrow{\scriptstyle H(A_{n-1},B_{n-1}} & & \big\downarrow{\scriptstyle H(A_n,B_n)} \\ B_1 & \xrightarrow[F(B_1,B_2)]{} & B_2 & \xrightarrow[F(B_2,B_3)]{} & B_3 & & \cdots & & B_{n-1} & \xrightarrow[F(B_{n-1},B_n)]{} & B_n \end{array} \tag{5}$$

variables	abcd	efgh
X	0000	1111
Y	0011	0011
Z	0101	0101
X'	1111	0000
Y'	1100	1100
Z'	1010	1010

Table 2. Truth–value assignment in the 3D space. The letters a,b, c, ... denote joint–value assignments for the 3 variables. The first and last one a denote contradiction (here, a) and tautology (here, h).

#	abcd	efgh	#	abcd	efgh
1	1111	0000	16	0000	1111
2	1110	1000	15	0001	0111
3	1101	0100	14	0010	1011
4	1100	1100	13	0011	0011
5	1011	0010	12	0100	1101
6	1010	1010	11	0101	0101
7	1001	0110	10	0110	1001
8	1000	1110	9	0111	0001

Table 3. Neg–reversal variables in the 3D logical space.

1	$X'=E(BC)'(B'C')'+B'C'$	16	$X=E'(BC)'(B'C')'+BC$
2	$A'=D(XZ)'(X'Z')'+X'Z'$	15	$A=D'(XZ)'(X'Z')'+XZ$
3	$B'=Z(XY)'(X'Y')'+X'Y'$	14	$B=Z'(XY)'(X'Y')'+XY$
4	$Y'=C(BE)'(B'E')'+B'E'$	13	$Y=C'(BE)'(B'E')'+BE$
5	$C'=Y(XZ)'(X'Z')'+X'Z'$	12	$C=Y'(XZ)'(X'Z')'+XZ$
6	$Z'=B(CE)'(C'E')'+C'E'$	11	$Z=B'(CE)'(C'E')'+CE$
7	$D'=X(BC)'(B'C')'+B'C'$	10	$D=X'(BC)'(B'C')'+BC$
8	$E'=X(YZ)'(Y'Z')'+Y'Z'$	9	$E=X'(YZ)'(Y'Z')'+YZ$

Table 4. The 16 irreducible variables in the 3D space.

Note the fractal–like generation of truth values from a $n-1$ space to a n space: each slot (either 0 or 1) of a variable's ID of the $(n-1)$D space doubles for the ID of a corresponding variable (to which we assigns the same label) in the nD space. In fact, each column of Tab. 2 is now doubled. For instance, from the 3D X whose ID is 0000 1111 we obtain the 4D X whose ID is 0000 0000 1111 1111. In other words, the whole of the collections of objects of any n–dimensional space represent a monoid $M=(\text{List}(X),++)$ without empty list that is generated by the set $X=\{0,1\}$ thanks to the operation of list concatenation $++$ [10, Secs. 3.1,4.2]; n represents the length of the list for any n–dimensional space. On the

variables	abcd	efgh	ijkl	mnop
X	0000	0000	1111	1111
Y	0000	1111	0000	1111
Z	0011	0011	0011	0011
Φ	0101	0101	0101	0101
X'	1111	1111	0000	0000
Y'	1111	0000	1111	0000
Z'	1100	1100	1100	1100
Φ'	1010	1010	1010	1010

Table 5. Truth–value assignment in the 4D space.

other hand, each n–dimensional space represents a group $(G, e^+, e^\times, +, \times, f^{-1})$ with buffer 2^n and identities $e^+ = 0$ and $e^\times = 1$ for disjunction (sum) $+$ and conjunction (product) $\times$, respectively. Both e^+ and $e^\times$ are identities for reversal f^{-1}.

Let us consider as a further example the 4D space (with $k = 256 \times 256 = 65,536$). For the 4D space, a truth–value table can be drawn as in Tab. 5. As mentioned, the number l of atomic variables for the 4D space is $2^8 = 256$ (where the fractal–like structure is again evident: note in particular that the first half of the following IDs coincides with the IDs of the 256 collections of objects of the 3D space), as displayed in Tab. 9 (see at the end of the paper).

The truth–values assignment of Tab. 5 corresponds to variables 256, 241, 205, 171 for X, Y, Z, Φ, respectively. Now, it is cumbersome but conceptually easy to verify that each of those sets is a transformation of the basic variables in a way that is again a resolution of identity. We need to generalize to n dimensions the algorithm displayed in Tab. 4 for the 3D space. This can be done in this way:

Lemma 2.2 *All Hadamard like transformations in any logical space of n dimension have the general form*

$$X_1 = X_2'(X_3 Y_4 \cdots X_n)'(X_3' X_4' \cdots X_n')' + X_3 X_4 \cdots X_n. \quad (6)$$

The lemma is self–evident. The formula immediately generates the Hadamard–like transformations H for the 4D space. For instance, the Set 129 above, let us call it variable A, can be expanded as $X'(YZ\Phi)'(Y'Z'\Phi')' + YZ\Phi$. Reciprocally, we can express X (Variable 256 above) as a combination of A (Variable 129), $B = Y'(XZ\Phi)'(X'Z'\Phi')' + XZ\Phi$ (Variable 144), $C = Z'(XY\Phi)'(X'Y'\Phi')' + XY\Phi$ (Variable 180), and $D = X(Y'Z'\Phi + YZ\Phi') + Y(Z'\Phi' + Z\Phi) + Y'Z\Phi'$ (Variable 215): $X = A'(BCD)'(B'C'D')' + BCD$: see Tab. 6.

It is well known that Boolean algebra satisfies the three requirements for a POSet, i.e.,

- Reflexivity: $\forall X, X \rightarrow X$ (where the arrow means implication),
- Transitivity: $\forall X, Y, Z$, if $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$,
- Antisymmetry: $\forall X, Y$, if $X \rightarrow Y$ and $Y \rightarrow X$, then X and Y are logically equivalent.

B	0111	0000	1111	0001	$\times$
C	0100	1100	1100	1101	$\times$
D	0010	1001	0110	1011	$=$
BCD	0000	0000	0100	0001	
$(BCD)'$	1111	1111	1011	1110	
B'	1000	1111	0000	1110	$\times$
C'	1011	0011	0011	0010	$\times$
D'	1101	0110	1001	0100	$=$
$B'C'D'$	1000	0010	0000	0000	
$(B'C'D')'$	0111	1101	1111	1111	
$(BCD)'$	1111	1111	1011	1110	$\times$
$(B'C'D')'$	0111	1101	1111	1111	$=$
$(BCD)'(B'C'D')'$	0111	1101	1011	1110	
A'	1000	0000	1111	1110	$\times$
$(BCD)'(B'C'D')'$	0111	1101	1011	1110	$=$
$A'(BCD)'(B'C'D')'$	0000	0000	1011	1110	
$A'(BCD)'(B'C'D')'$	0000	0000	1011	1110	$+$
BCD	0000	0000	0100	0001	$=$
X	0000	0000	1111	1111	

Table 6. An example of reversed Hadamard–like transformation in the 4D space.

The first two properties define a Preorder. However, it is also well known that Boolean algebra is not a linear POSet [10, Sec. 3.4], i.e. it does not satisfy

- Comparability: $\forall X, Y$, either $X \rightarrow Y$ or $Y \rightarrow X$.

The reason for that is precisely due to the existence of a collection of irreducible atomic variables and of their relations in terms of resolution of identity. However, each neg–reversal variable selects a subspace in every $n \geq 2$ logical space that is linear if we consider paths, which follow either meets (limits) or joins (colimits) [1, Chap. 1], as displayed in Figs. 2–3. (Note that such subspaces do not represent the $n-1, n-2, \ldots$ proper subspaces of each n–dimensional space: for instance, there are three 2D and six 1D subspaces in the 3D space [1, Chap. 8].)

In other words, both Preorders and finite linear orders are categories and the latter constitute some of the objects of the former [10, Sec. 4.1]. This fully justifies the notion of irreducible sets. Note that for any linear subspace, each lower level node implying a higher level node is tautology while the sum of all nodes for each level lower than the variable itself is equivalent to the latter, while the product of all nodes of each level higher than the variables gives the latter. For instance, in the 3D space we have:

$$\begin{aligned} X(YZ' + Y'Z) \rightarrow (X + Y') &= X + X', \\ XY + XZ + X(YZ' + Y'Z) + X(YZ + Y'Z') + XZ' + XY' &= X, \\ (X + YZ)(X + YZ')(X + Y'Z)(X + Y'Z') &= X. \quad (7) \end{aligned}$$

Note also that for every n–dimensional space the collections of objects of half a linear subspace (that is, from contradiction to the variable and from the latter to tautology) have the same number as the collections of objects of the $(n-1)$D space: for instance, for the 3D space, the collections of objects of the

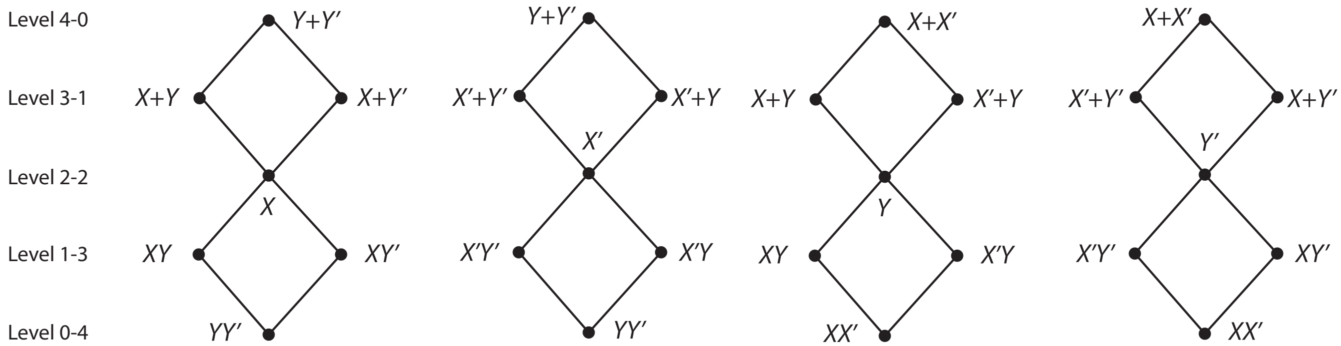

Figure 2. In the 2D logical space we can select four linear subspaces individuated by the variables X,X',Y,Y'. Note in fact that every pair of nodes being on one of the paths satisfy the comparability requirement. To help the reader, I have organized the space in levels: the first figure is the number of 1s for that level while the second one is the number of 0s for that level. For instance, Level 3-5 collects all IDs with three 1s and five 0s. Note that the whole structure has the form of a double diamond with the neg–reversal variable representing the joining point between the two. Note that here (and for every further logical space) the linear subspaces generated by the two contradictory variables (here X and X') cover all collections of objects of the two levels just above the contradiction and below the tautology (here Levels 1-7 and 7-1, respectively).

linear half–network of X and X' are 16 (and similarly for any other neg–reversal variable).

3. Spanning spaces Logically

As mentioned in the introduction, I shall introduce the notion of logical space. First, we need to set the requirements for defining what is a set of variables spanning the nD logical space. If we like to preserve some notion of independent vector in this context, each set spanning the space must satisfy the following requirements:

(i) The vectors constituting the basis share pairwise the minimal number of 0s (or 1s) that is logically possible in that space, which turns out to be $m/2$,

(ii) Due to the structure of the neg–reversal variables, they must pairwise share $m/4$ truth values among the $m/2$ numbers constituting the first half of the ID and $m/4$ among the $m/2$ numbers constituting the second half of the ID.

Now, I formulate the following lemma

Lemma 3.1 *For any n–dimensional space, sets of n irreducible atomic variables are sufficient to span the space.*

Proof: Any n-dimensional space can be spanned in the following ways: by both (i) replacing a 0 by a 1 for each level of the algebra up to the tautology (displaying m 1s), and (ii) replacing a 1 by a 0 each level for each level of the algebra down to the contradiction (displaying m 0s). Usually, it is assumed that we span the Boole–Tarski–Lindenbaum algebra by combining collections of objects, essentially making use of disjunctions of basic variables as well as their disjunctions for climbing the levels of the corresponding algebra and of conjunctions of variables as well as their conjunctions for descending the ladder of the algebra. In fact, it is an issue of pure combinatorial calculus, as the generation of all collections of objects of a logical space follows the Pascal triangle. For instance, for a 3D space, the number $k = 256$ of collections of objects is generated by the sequence 1, 8, 28, 56, 70, 56, 28, 8, 1, whose sum is 256, which can be expressed in binomial coefficients as:

$$k = \sum_{x=0}^{8} \binom{8}{x}, \tag{8}$$

where the variable number below can be taken to represent the number of 0s (or of 1s) at each level. Generalizing to any n dimensional space, we have

$$k(n) = \sum_{x=0}^{2^n} \binom{2^n}{x}, \tag{9}$$

where I have expressed the dependence of k on n. The previous equation is an instance of the general formula

$$\sum_{x=0}^{2^n} \binom{2^n}{x} = 2^{\sum_{y=0}^{n} \binom{n}{y}}. \tag{10}$$

It is now evident that only collections of objects that are the neg–reversal of themselves can span the space satisfying the requirements (i) and (ii), that is, those collections of objects are able to span their logical space. ■

Obviously, any nD space can be spanned also with other vectors. Nevertheless, neg–reversal variables are the only ones that can span the space satisfying the two properties above. Note, in particular, that the notion of logical basis is different relative to the geometric notion of basis. In fact, we can take a vectorial basis for e.g. the 3D logical space to be represented

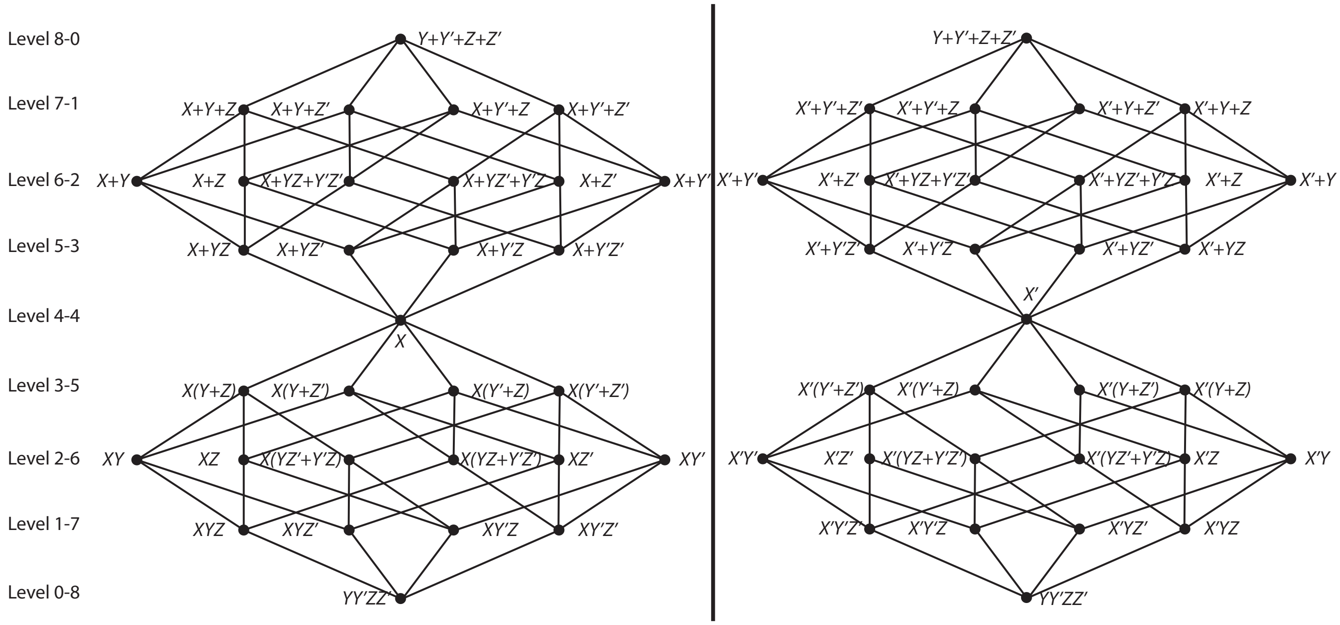

Figure 3. Two examples (X and X') out of the 16 neg–reversal variables of the 3D space. The other neg–reversal variables determine similar subspaces.

by the $m = 2^n = 8$ vectors [1, Chap 9]

$$\begin{pmatrix}1\\0\\0\\0\\0\\0\\0\\0\end{pmatrix},\begin{pmatrix}0\\1\\0\\0\\0\\0\\0\\0\end{pmatrix},\begin{pmatrix}0\\0\\1\\0\\0\\0\\0\\0\end{pmatrix},\begin{pmatrix}0\\0\\0\\1\\0\\0\\0\\0\end{pmatrix},$$
$$\begin{pmatrix}0\\0\\0\\0\\1\\0\\0\\0\end{pmatrix},\begin{pmatrix}0\\0\\0\\0\\0\\1\\0\\0\end{pmatrix},\begin{pmatrix}0\\0\\0\\0\\0\\0\\1\\0\end{pmatrix},\begin{pmatrix}0\\0\\0\\0\\0\\0\\0\\1\end{pmatrix}, \qquad (11)$$

which correspond to the 8 areas a, b, c, ..., h, as displayed in Fig. 1. At the opposite, the vectors logically spanning the space (due to the non–linearity of the space) are $n = 3$ (either X, Y, Z or X', Y', Z'), as in Tab. 2, and can be understood as particular superpositions of the latter (those giving rise to neg–reversal variables). The geometrical representation of the logical space is therefore quite different. In fact, although this basis expresses geometric linear independence of vectors, it is not the same for the truth values, as it is evident by the fact that all vectors above share pairwise six truth values. For instance, the first two vectors can be logically represented by expressions $X'Y'Z'$ and $X'Y'Z$, respectively.

Thus, the logical basis is n–dimensional while a corresponding pure geometric basis would be 2^n–dimensional on the same space. Obviously, there is a morphism between these two allowing to back–translate logical operations into traditional geometric representation. Since all possible sets of n neg–reversal variables of a n–dimensional space span the whole space, these variables may be called the *generators* of that space.

The previous lemma allows us to formulate the following theorem:

Theorem 3.1 *For any n–dimensional space the number of irreducible atomic variables is finite and is equal to* $2^{\frac{m}{2}}$.

Proof: According to the previous examination, all irreducible atomic variables need to be neg–reversal variables. This means that they are identified by half the sequence of their binary ID. An immediate consequence is that, for each n–dimensional space with $k = 2^m$ collections of objects, the number of these atomic variables is $l(m) = 2^{\frac{m}{2}}$, where l is expressed as a function of m. Both m and l can be expressed as functions of n in the following way:

$$l(n) = \sum_{x=0}^{2^{n-1}} \begin{pmatrix} 2^{n-1} \\ x \end{pmatrix}, \qquad (12)$$

$$m(n) = \sum_{y=0}^{n} \begin{pmatrix} n \\ y \end{pmatrix}. \qquad (13)$$

This implies that their number is necessarily finite. ■ The fact that the atomic variables have to be $l(m)$ will be proved below.

When the number n of the dimensions of the space grows tending to infinity, the number $2^{\frac{m}{2}}$ of irreducible atomic vari-

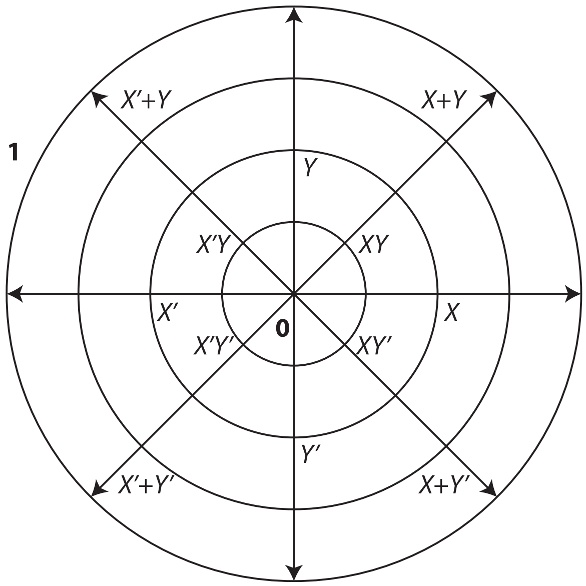

Figure 4. The way in which we can represent the logical spanning of 2D space. Note that we have here 5 circles (one of them represented by the **0** point) corresponding to the 5 levels displayed in Fig. 2.

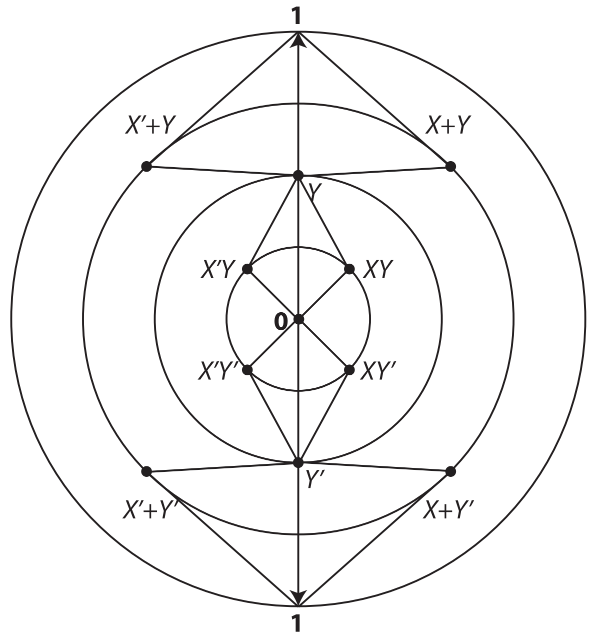

Figure 5. The linear subspace determined by Y and Y'.

ables relatively shrinks tending to 0, according to the series

$$\frac{1}{2^{\frac{m}{2}}} = \frac{1}{2^{2^{n-1}}}. \tag{14}$$

For instance, for a 3D space, the irreducible variables representing atomic sets are $1/16$ of all k collections of objects; for a 4D space the irreducible variables are $1/256$ of all k collections of objects; for a 5D space, the irreducible atomic variables are $1/65,536$ of all k collections of objects, and so on.

Up to now, I have dealt with vectorial representations of the logical variables. In fact, such a logical space can also be made isomorphic to the hypersphere of quantum–mechanical density matrices, at least for the 3D case. First of all we need to introduce vectors of different length $\lambda \leq 1$, with equality sign corresponding to tautology (symbolized by **1**). Therefore, the space is represented by a $(n-1)$–hypersphere of unitary radius with spanning vectors with length of 1/2, hyper–surface representing the tautology and center contradiction (symbolized by **0**). Note that any point can be reached from the latter and we can reach the former from any point and the expressions are always the same (a tautology). For instance, let us take the simple case of the 2D space [Fig. 4]. Note that equivalences and counter–valences are represented by bidirectional vectors. Obviously, we can add vectors of different length as well as addition of vectors of same length can give rise to a vector of different length.

In particular, we can map irreducible statements like X, X', $Y, \ldots$ to reduced density matrices in that space, while statements of the form $X+Y$ to mixtures like $\hat{P}_x + \hat{P}_y$, where the weights have no logical significance (also the phase differences are logically irrelevant). In fact, some X are true or some Y are true. Expressions like XY represent coincident events ($\hat{P}_x\hat{P}_y$) while equivalences to entangled states, where, I recall, also classical components are involved. Obviously, we deal each time only with binary projections so that X and X' represent sets $\{\hat{P}_x, \hat{P}_{x'}\}$. Note finally that tautology **1** (representing, as we shall see, a pure state) is in fact a scalar and covers the whole surface of the unitary sphere, while the contradiction **0**, as said, represents the center of the sphere. We can pack in these logical spaces the linear subspaces shown e.g. in Fig. 2 as displayed in Fig. 5. This allows us to understand a reduced state as a linear subspace of a certain variable X.

4. Alternative Sets and Bases

Note that for spaces of dimension $n > 2$ there are more sets of n neg–reversal variables that span the whole space. How to individuate this kind of basis (appropriately collecting generators)? In fact, many combinations of generators will not work. In this logical context, I have defined such variables logically spanning a logical space (generators) and constituting a logical basis as a group of vectors sharing the minimal amount of truth–values that is possible. The whole set of $l(m) = 2^{\frac{m}{2}}$ generators (all of the neg–reversal variables) of the nD space can be partitioned in subsets such that all variables pertaining to the set at least pairwise (but not all) share half of the truth values. Let us call a *spanning set* any such subset regrouping variables that at least pairwise share share half of the truth values.

The number of the variables pertaining to this subset is in general larger than the number of n variables sufficient to span the n–dimensional space. However, there are several choices of n variables among those constituting such a subset that are good for spanning the space (and the same is true for other subsets). The basis requirement is the following lemma: All n variables of a n–dimensional space constituting a basis can share only two values: the first when all of them are false and the last when all of them are true. This lemma gives us the definition of "linearly independent" vectors constituting a basis. I stress that an arbitrary number of neg–reversal variables cannot share less that these two values, and therefore this is the minimal amount that in a logical space is in general possible.

Now, I shall show that the number of variables pertaining to any spanning set of the n–dimensional space is equal to the number of shared truth–values, i.e. 2^{n-1}. The algorithm for building the number of spanning sets for dimension $n \geq 3$ is given by

$$s(n) = \frac{m(n-1)}{2} s(n-1), \tag{15}$$

where $s(n)$ denotes the number of sets for the nD logical space.

I shall proceed in a constructive and iterative way, as an instance of list concatenation. Note that here and in the following we can consider only either classes or their complements (i.e. either X or X'), what reduces to half the whole amount of computation. The first two cases (that are not covered by the general formula) are quite easy. For a 1D logical space, we have a single spanning set of a single variable (X) and its negation, whose IDs are 01 and 10, respectively.

For the 2D logical space we have again one single spanning set with two variables. These variables and their negations are built by starting with the 1D variable and its negation and multiplying all of them, generating four sequences, the two 2D variables and their negations. Since we deal with neg–reversal variables, these new variables are built by splitting the two terms to be multiplied into two parts, so that we get: 0 01 1, 0 10 1, 1 01 0, 1 10 0. In other words, we have an "external" and "internal" part of the product. Let us now establish a new and univocal convention for picking up the right variables. Let us denote with X_0 the 1D variable and with $X_1 = X$ and $X_2 = Y$ the two 2D variables. These results could be written as (where the "external" part comes first): $X_0 \otimes X_0 = X_1 = 0011$, $X_0 \otimes X_0' = X_2 = 0101$, $X_0' \otimes X_0 = X_2' = 1010$, and $X_0' \otimes X_0' = X_1' = 1100$, where $\otimes$ denotes the operation of mixing IDs of neg–reversal variables for getting IDs of higher–dimensional neg–reversal variables (and not the AND operation).

For the 3D space, we proceed in the same way, getting (where I do not consider the negations): $X_{1.1} = X_1 \otimes X_1 = 00\ 0011\ 11$, $X_{2.1} = X_1 \otimes X_2 = 00\ 0101\ 11$, $X_{3.1} = X_1 \otimes X_2' = 00\ 1010\ 11$, $X_{4.1} = X_1 \otimes X_1' = 00\ 11000\ 11$, $X_{1.2} = X_2 \otimes X_1 = 01\ 0011\ 01$, $X_{2.2} = X_2 \otimes X_2 = 01\ 0101\ 01$, $X_{3.2} = X_2 \otimes X_2' = 01\ 1010\ 01$, and $X_{4.2} = X_2 \otimes X_1' = 01\ 1100\ 01$. This gives Tab. 7, from which (satisfying the criteria imposed for logical–vector independency) we get the $m/2 = 8$ alternative bases, which are built by some kind of rotation:

$\{X_{1.1}, X_{4.1}, X_{2.2}\}, \{X_{1.1}, X_{4.1}, X_{3.2}\}, \{X_{1.1}, X_{2.2}, X_{3.2}\}, \{X_{4.1}, X_{2.2}, X_{3.2}\}$;
$\{X_{2.1}, X_{3.1}, X_{1.2}\}, \{X_{2.1}, X_{3.1}, X_{4.2}\}, \{X_{2.1}, X_{1.2}, X_{4.2}\}, \{X_{3.1}, X_{1.2}, X_{4.2}\}$.

Note that the two couples of each set that stem from either X_1 or X_2 have the "internal" part that is the negation of each other. This will be a common trait for all sets of any nD space with $n \geq 3$.

	variables	#	abcd	efgh
Set 1	$X_{1.1}$	16	0000	1111
	$X_{4.1}$	13	0011	0011
	$X_{2.2}$	11	0101	0101
	$X_{3.2}$	10	0110	1001
Set 2	$X_{2.1}$	15	0001	0111
	$X_{3.1}$	14	0010	1011
	$X_{1.2}$	12	0100	1101
	$X_{4.2}$	9	0111	0001

Table 7. The two spanning sets of the 3D logical space.

This is evident by considering Venn diagrams [see Fig. 1], but we can also use another type of diagrams that can also be applied to spaces of dimension > 3 [see Fig. 6].

We can write the number of possible choices of 3 variables for each spanning set of 4 variables (the Sets 1–2 of Tab. 7) as

$$\begin{pmatrix} 4 \\ 3 \end{pmatrix} = 4. \tag{16}$$

Note that any basis is such that the sum of the 1s (or 0s) of each column follows the binomial coefficient:

$$\begin{pmatrix} 3 \\ x \end{pmatrix}, \quad \text{with} \quad 0 \leq x \leq 3, \tag{17}$$

that is, one column with no 1, three columns with a 1, three columns with 2 1s, one column with three 1s. This is generalizable to any n–dimensional space as:

$$\begin{pmatrix} n \\ x \end{pmatrix}, \quad \text{with} \quad 0 \leq x \leq n. \tag{18}$$

Note also that in each of the sets displayed in Tab. 7, each column sums to 2 1s apart from the first and the last that sum to 0 and 4, respectively. I recall that, for any n–dimensional space, each set is indeed built in such a way that apart from the first and last column repressing all 0s and all 1s, respectively, we have $2^n - 2$ columns with 2^{n-2} 1s.

For the 4D space, we apply again the same procedure: each of the 3D sixteen variables is multiplied by all the sixteen variables, generating 256 variables with their negations. Then, we have 16 sets, which are generated in the easiest way by the variables deriving form those of the 3D space pertaining to Sets 1 and 2, i.e. $X_{1.1}, X_{4.1}, X_{2.2}, X_{3.2}$ and $X_{2.1}, X_{3.1}, X_{1.2}, X_{4.2}$, respectively, as displayed in Tab. 10.

Note that the procedure shown here proves the second half of theorem 3.1. In fact, the general algorithm is that the number $2^{m(n)}$ of variables (including their negations) for any nD space is given by

$$2^{m(n)} = 2^{m(n-1)} \cdot 2^{m(n-1)} = 2l(m). \quad (19)$$

Abstractly speaking, for the 4D space, we have for each set of 8 items out of which we need to choose quadruplets the following binomial coefficient:

$$\binom{8}{4} = 70. \quad (20)$$

However, for computing the possible alternative bases for each spanning set of the 4D space, I recall that we need to consider that all the 4 variables need to share only the first and the last digit for satisfying linear independence. This implies that we cannot have in the same basis two couples whose pairs have the 2nd and 3rd 4 numbers the negative of each other. For instance, let us take Set 12, as in Tab. 8. Then, the following six combinations are forbidden:

$$X_{6.2.1}X_{11.2.1}X_{7.3.1}X_{10.3.1}, \quad X_{6.2.1}X_{11.2.1}X_{4.4.2}X_{13.4.2},$$
$$X_{1.1.2}X_{16.1.2}X_{6.2.1}X_{11.2.1}, \quad X_{7.3.1}X_{10.3.1}X_{4.1.2}X_{13.1.2},$$
$$X_{7.3.1}X_{10.3.1}X_{1.4.2}X_{16.4.2}, \quad X_{4.1.2}X_{13.1.2}X_{1.4.2}X_{16.4.2},$$

with the result that the total number of the permissible combinations for a single spanning set is 64. Since the spanning sets are 16, this makes the total number equal to $2^7 \cdot 2^4 = 2^{11}$.

$X_{6.2.1}$	235	0001	0101	0101	0111
$X_{11.2.1}$	230	0001	1010	1010	0111
$X_{7.3.1}$	218	0010	0110	1001	1011
$X_{10.3.1}$	215	0010	1001	0110	1011
$X_{4.1.2}$	189	0100	0011	0011	1101
$X_{13.1.2}$	180	0100	1100	1100	1101
$X_{1.4.2}$	144	0111	0000	1111	0001
$X_{16.4.2}$	129	0111	1111	0000	0001

Table 8. The 8 variables of Set 12 of the 4D space.

5. Symmetrization

We could proceed for higher–dimensional spaces in the previous way. However, there is a more fruitful method. We may have noted that for the 3D space there is the "anomaly" that we have a number of dimensions that is not a multiple of 2 although it is still related to the number m of truth value assignments. We can avoid this problem by symmetrizing the space and use a 4D space (which is equal to $m/4 = 2^2$). In that case we have two alternative bases represented by the two sets of Tab. 7. Now, we can univocally map the two 2D variables to the two alternative bases and have the straightforward transformations among variables shown in Tab. 11.

By multiplying any of the above couple of statements we get the 28 statements of level 6.2, as displayed in Tab. 12.

In a similar way, we can build the other statements. For instance, statement 11111000 of Level 5-3 is given by $X'_{1.1} + X'_{4.1}X'_{2.2} = X'_{1.1} + X'_{4.1}X'_{3.2} = X'_{1.1} = X'_{2.2}X'_{3.2}$, which are in turn equal to $X'_{2.1} + X'_{3.1}X'_{1.2} = X'_{2.1} + X'_{3.1}X'_{4.2} = X'_{2.1} + X'_{1.2}X'_{4.2}$.

We can adopt this procedure for higher–dimensional spaces. In the case of the 4D space, we use again a whole spanning set to build a single basis. Therefore, we build a 8D logical space. In such particular case, the advantage of the symmetrization is less evident since 4D bases are already multiple of 2. Here, things are also a little bit more complicated. For instance, we can get (among many others) the substitutions of the Basis–1 variables displayed in Array (21).

Nevertheless, it can be helpful to proceed in this way if we think to use the same method also for other spaces. For example, we replace the 5D space by a 16D space and proceed again in a similar way. The advantage is that we avoid complex calculations of the number of alternative bases for each nD space since thewy come to coincide with the number of spanning sets, and both this number and that of variables is easily computable with the previous algorithms. Thus, for each nD space we build bases with a number of elements (dimensions) that are multiples of 2 congruent with the original (not symmetrized) dimension of the space: 2^0 for the 1D space, 2^1 for the 2D space, 2^2 for the 3D space, 2^3 for the 4D space, 2^4 for the 5D space, and so on, where the exponent for the non–symmetrized nD logical space is $n-1$.

6. Results

In short, the main results of this study are:

- For any finite logical space there is a finite number of variables representing basic sets that cannot be reduced to some collections of other sets, and their number is $l(n) = 2^{2n-1}$ for any n–dimensional space, according to Theorem 3.1.

- For any $n > 2$ logical space there are alternative sets of atomic variables and each set displays actually resolutions of identity of variables pertaining to other sets.

- These sets represent bases which can be regrouped in spanning sets, whose number is $l(n)/m(n)$ for any n–dimensional space.

- By making us of symmetrization we circumvent the problem of the calculation of bases as far as the number of bases of each nD space is $m/2$.

- The formalism of logical spaces can help us to overcome some known paradoxes in logic and set theory.

- It can be very helpful for classical and quantum computation.

- It can be helpful also in other fields of mathematics where several computations of bases are necessary.

7. Discussion

It might be noted that nobody, as far as I know, as thought about the possibility to have irreducible variables for each n–dimensional Boolean algebra. The reason is that the construction of these Boolean algebras is currently made through composition of sets of objects into new sets of objects through conjunction and disjunction and not looking at the pure combinatorial aspects dealing with pure Boolean bitstrings. There is a deep reason for that. Logic and its applications has been traditionally treated as an algebra but without the arithmetic substrate of mathematical algebra. However, it is only arithmetics (i.e. computation with numbers) that allows us to use mathematics in the powerful way that is its characteristics, from physics to engineering. Now, if we ask what is the feature that makes arithmetics so powerful, the answer is very simple: all rational numbers can be represented as dots in an arithmetic space (a line) in which we can easily pick up the successor of any arbitrary number, what allows to perform operations on these numbers (the same is true for real numbers although it is not always easy to discriminate between them). It is not by chance that Peano individuated in the relation "to be successor of" the distinctive feature of arithmetics [7]. Now, the building of a logical space allows us to individuate the "position" in the logical space of each collection of objects (through its binary ID) in a way that is univocal, thus representing a kind of logical arithmetics that establishes univocal relations among the collections of objects themselves (whether they represent propositions or classes). The first to have though about this possibility is K. Gödel [6], although his numbers only have the purpose to represent statements and not to be used for calculation: in fact are far more complex that the binary IDs (they are like "Roman" cyphers relative to decimal numbers).

This result is thus very surprising as far as it is commonly assumed that any statement that appears atomic could in fact be molecular, so that this distinction was understood to be finally only a matter of convenience. At the opposite, I have proved that there is a finite number of basic and irreducible atomic variables for each nD logical space. In other words, this sets specific limitations on the possible substitutions: only generators of a n–dimensional space and their combinations that give rise to other generators can be substituted to atomic variables of that space. In fact, only atomic variables represent sets in the logical space. Such an approach confirms the results of Category theory for solving the known paradoxes in set theory [9]. In fact, those paradoxes are built in such a way that sets of objects can built one from the other as Chinese boxes without taking care of their possible relations. If this non–logical assumption is removed, also the paradoxes disappear.

Moreover, the previous formalism is very useful for both classical and quantum computation: see also [1, Chap. 9]. In fact, we can deal with problems of computation by renouncing to implement a number of logical rules or connections in a processor but rather making use of a simple logical or dot–space. In fact, a logical space can be considered as the analogue of a network, so that any computation running free through this space will spontaneously establish connections that are all logical. In a subsequent paper I shall show the extensions of this point of view. Now, training the network by repeated use in similar conditions (in a way that is reminiscent of neural networks) allows us to establish connections that are reinforced with time and become so the privileged ones. Moreover, the above results for bases easily allows implementing quantum computation. In such a case, it would be suitable to use negation, sum and AND NOT as basic operators. Finally, the method shows here for computing generators and bases for any nD logical space can also have wider applications to mathematics.

#	abcd	efgh	ijkl	mnop	#	abcd	efgh	ijkl	mnop
1	1111	1111	0000	0000	256	0000	0000	1111	1111
2	1111	1110	1000	0000	255	0000	0001	0111	1111
3	1111	1101	0100	0000	254	0000	0010	1011	1111
4	1111	1100	1100	0000	253	0000	0011	0011	1111
5	1111	1011	0010	0000	252	0000	0100	1101	1111
6	1111	1010	1010	0000	251	0000	0101	0101	1111
7	1111	1001	0110	0000	250	0000	0110	1001	1111
8	1111	1000	1110	0000	249	0000	0111	0001	1111
9	1111	0111	0001	0000	248	0000	1000	1110	1111
10	1111	0110	1001	0000	247	0000	1001	0110	1111
11	1111	0101	0101	0000	246	0000	1010	1010	1111
12	1111	0100	1101	0000	245	0000	1011	0010	1111
13	1111	0011	0011	0000	244	0000	1100	1100	1111
14	1111	0010	1011	0000	243	0000	1101	0100	1111
15	1111	0001	0111	0000	242	0000	1110	1000	1111
16	1111	0000	1111	0000	241	0000	1111	0000	1111
17	1110	1111	0000	1000	240	0001	0000	1111	0111
18	1110	1110	1000	1000	239	0001	0001	0111	0111
19	1110	1101	0100	1000	238	0001	0010	1011	0111
20	1110	1100	1100	1000	237	0001	0011	0011	0111
21	1110	1011	0010	1000	236	0001	0100	1101	0111
22	1110	1010	1010	1000	235	0001	0101	0101	0111
23	1110	1001	0110	1000	234	0001	0110	1001	0111
24	1110	1000	1110	1000	233	0001	0111	0001	0111
25	1110	0111	0001	1000	232	0001	1000	1110	0111
26	1110	0110	1001	1000	231	0001	1001	0110	0111
27	1110	0101	0101	1000	230	0001	1010	1010	0111
28	1110	0100	1101	1000	229	0001	1011	0010	0111
29	1110	0011	0011	1000	228	0001	1100	1100	0111
30	1110	0010	1011	1000	227	0001	1101	0100	0111
31	1110	0001	0111	1000	226	0001	1110	1000	0111
32	1110	0000	1111	1000	225	0001	1111	0000	0111
33	1101	1111	0000	0100	224	0010	0000	1111	1011
34	1101	1110	1000	0100	223	0010	0001	0111	1011
35	1101	1101	0100	0100	222	0010	0010	1011	1011
36	1101	1100	1100	0100	221	0010	0011	0011	1011
37	1101	1011	0010	0100	220	0010	0100	1101	1011
38	1101	1010	1010	0100	219	0010	0101	0101	1011
39	1101	1001	0110	0100	218	0010	0110	1001	1011
40	1101	1000	1110	0100	217	0010	0111	0001	1011
41	1101	0111	0001	0100	216	0010	1000	1110	1011
42	1101	0110	1001	0100	215	0010	1001	0110	1011
43	1101	0101	0101	0100	214	0010	1010	1010	1011
44	1101	0100	1101	0100	213	0010	1011	0010	1011
45	1101	0011	0011	0100	212	0010	1100	1100	1011
46	1101	0010	1011	0100	211	0010	1101	0100	1011
47	1101	0001	0111	0100	210	0010	1110	1000	1011
48	1101	0000	1111	0100	209	0010	1111	0000	1011
49	1100	1111	0000	1100	208	0011	0000	1111	0011
50	1100	1110	1000	1100	207	0011	0001	0111	0011
51	1100	1101	0100	1100	206	0011	0010	1011	0011

52	1100	1100	1100	1100	205	0011	0011	0011	0011
53	1100	1011	0010	1100	204	0011	0100	1101	0011
54	1100	1010	1010	1100	203	0011	0101	0101	0011
55	1100	1001	0110	1100	202	0011	0110	1001	0011
56	1100	1000	1110	1100	201	0011	0111	0001	0011
57	1100	0111	0001	1100	200	0011	1000	1110	0011
58	1100	0110	1001	1100	199	0011	1001	0110	0011
59	1100	0101	0101	1100	198	0011	1010	1010	0011
60	1100	0100	1101	1100	197	0011	1011	0010	0011
61	1100	0011	0011	1100	196	0011	1100	1100	0011
62	1100	0010	1011	1100	195	0011	1101	0100	0011
63	1100	0001	0111	1100	194	0011	1110	1000	0011
64	1100	0000	1111	1100	193	0011	1111	0000	0011
65	1011	1111	0000	0010	192	0100	0000	1111	1101
66	1011	1110	1000	0010	191	0100	0001	0111	1101
67	1011	1101	0100	0010	190	0100	0010	1011	1101
68	1011	1100	1100	0010	189	0100	0011	0011	1101
69	1011	1011	0010	0010	188	0100	0100	1101	1101
70	1011	1010	1010	0010	187	0100	0101	0101	1101
71	1011	1001	0110	0010	186	0100	0110	1001	1101
72	1011	1000	1110	0010	185	0100	0111	0001	1101
73	1011	0111	0001	0010	184	0100	1000	1110	1101
74	1011	0110	1001	0010	183	0100	1001	0110	1101
75	1011	0101	0101	0010	182	0100	1010	1010	1101
76	1011	0100	1101	0010	181	0100	1011	0010	1101
77	1011	0011	0011	0010	180	0100	1100	1100	1101
78	1011	0010	1011	0010	179	0100	1101	0100	1101
79	1011	0001	0111	0010	178	0100	1110	1000	1101
80	1011	0000	1111	0010	177	0100	1111	0000	1101
81	1010	1111	0000	1010	176	0101	0000	1111	0101
82	1010	1110	1000	1010	175	0101	0001	0111	0101
83	1010	1101	0100	1010	174	0101	0010	1011	0101
84	1010	1100	1100	1010	173	0101	0011	0011	0101
85	1010	1011	0010	1010	172	0101	0100	1101	0101
86	1010	1010	1010	1010	171	0101	0101	0101	0101
87	1010	1001	0110	1010	170	0101	0110	1001	0101
88	1010	1000	1110	1010	169	0101	0111	0001	0101
89	1010	0111	0001	1010	168	0101	1000	1110	0101
90	1010	0110	1001	1010	167	0101	1001	0110	0101
91	1010	0101	0101	1010	166	0101	1010	1010	0101
92	1010	0100	1101	1010	165	0101	1011	0010	0101
93	1010	0011	0011	1010	164	0101	1100	1100	0101
94	1010	0010	1011	1010	163	0101	1101	0100	0101
95	1010	0001	0111	1010	162	0101	1110	1000	0101
96	1010	0000	1111	1010	161	0101	1111	0000	0101
97	1001	1111	0000	0110	160	0110	0000	1111	1001
98	1001	1110	1000	0110	159	0110	0001	0111	1001
99	1001	1101	0100	0110	158	0110	0010	1011	1001
100	1001	1100	1100	0110	157	0110	0011	0011	1001
101	1001	1011	0010	0110	156	0110	0100	1101	1001
102	1001	1010	1010	0110	155	0110	0101	0101	1001
103	1001	1001	0110	0110	154	0110	0110	1001	1001
104	1001	1000	1110	0110	153	0110	0111	0001	1001
105	1001	0111	0001	0110	152	0110	1000	1110	1001
106	1001	0110	1001	0110	151	0110	1001	0110	1001

107	1001	0101	0101	0110	150	0110	1010	1010	1001
108	1001	0100	1101	0110	149	0110	1011	0010	1001
109	1001	0011	0011	0110	148	0110	1100	1100	1001
110	1001	0010	1011	0110	147	0110	1101	0100	1001
111	1001	0001	0111	0110	146	0110	1110	1000	1001
112	1001	0000	1111	0110	145	0110	1111	0000	1001
113	1000	1111	0000	1110	144	0111	0000	1111	0001
114	1000	1110	1000	1110	143	0111	0001	0111	0001
115	1000	1101	0100	1110	142	0111	0010	1011	0001
116	1000	1100	1100	1110	141	0111	0011	0011	0001
117	1000	1011	0010	1110	140	0111	0100	1101	0001
118	1000	1010	1010	1110	139	0111	0101	0101	0001
119	1000	1001	0110	1110	138	0111	0110	1001	0001
120	1000	1000	1110	1110	137	0111	0111	0001	0001
121	1000	0111	0001	1110	136	0111	1000	1110	0001
122	1000	0110	1001	1110	135	0111	1001	0110	0001
123	1000	0101	0101	1110	134	0111	1010	1010	0001
124	1000	0100	1101	1110	133	0111	1011	0010	0001
125	1000	0011	0011	1110	132	0111	1100	1100	0001
126	1000	0010	1011	1110	131	0111	1101	0100	0001
127	1000	0001	0111	1110	130	0111	1110	1000	0001
128	1000	0000	1111	1110	129	0111	1111	0000	0001

Table 9. Atomic variables in the 4D logical space.

		#	abcd	efgh	ijkl	mnop			#	abcd	efgh	ijkl	mnop
Set 1	$X_{1.1.1}$	256	0000	0000	1111	1111	Set 2	$X_{1.2.1}$	240	0001	0000	1111	0111
	$X_{16.1.1}$	241	0000	1111	0000	1111		$X_{16.2.1}$	225	0001	1111	0000	0111
	$X_{4.4.1}$	205	0011	0011	0011	0011		$X_{4.3.1}$	221	0010	0011	0011	1011
	$X_{13.4.1}$	196	0011	1100	1100	0011		$X_{13.3.1}$	212	0010	1100	1100	1011
	$X_{6.2.2}$	171	0101	0101	0101	0101		$X_{6.1.2}$	187	0100	0101	0101	1101
	$X_{11.2.2}$	166	0101	1010	1010	0101		$X_{11.1.2}$	182	0100	1010	1010	1101
	$X_{7.3.2}$	154	0110	0110	1001	1001		$X_{7.4.2}$	138	0111	0110	1001	0001
	$X_{10.3.2}$	151	0110	1001	0110	1001		$X_{10.4.2}$	135	0111	1001	0110	0001
Set 3	$X_{2.1.1}$	255	0000	0001	0111	1111	Set 4	$X_{2.2.1}$	239	0001	0001	0111	0111
	$X_{15.1.1}$	242	0000	1110	1000	1111		$X_{15.2.1}$	226	0001	1110	1000	0111
	$X_{3.4.1}$	206	0011	0010	1011	0011		$X_{3.3.1}$	222	0010	0010	1011	1011
	$X_{14.4.1}$	195	0011	1101	0100	0011		$X_{14.3.1}$	211	0010	1101	0100	1011
	$X_{5.2.2}$	172	0101	0100	1101	0101		$X_{5.1.2}$	188	0100	0100	1101	1101
	$X_{12.2.2}$	165	0101	1011	0010	0101		$X_{12.1.2}$	181	0100	1011	0010	1101
	$X_{8.3.2}$	153	0110	0111	0001	1001		$X_{8.4.2}$	137	0111	0111	0001	0001
	$X_{9.3.2}$	152	0110	1000	1110	1001		$X_{9.4.2}$	136	0111	1000	1110	0001
Set 5	$X_{3.1.1}$	254	0000	0010	1011	1111	Set 6	$X_{3.2.1}$	238	0001	0010	1011	0111
	$X_{14.1.1}$	243	0000	1101	0100	1111		$X_{14.2.1}$	227	0001	1101	0100	0111
	$X_{2.4.1}$	207	0011	0001	0111	0011		$X_{2.3.1}$	223	0010	0001	0111	1011
	$X_{15.4.1}$	194	0011	1110	1000	0011		$X_{15.3.1}$	210	0010	1110	1000	1011
	$X_{8.2.2}$	169	0101	0111	0001	0101		$X_{8.1.2}$	185	0100	0111	0001	1101
	$X_{9.2.2}$	168	0101	1000	1110	0101		$X_{9.1.2}$	184	0100	1000	1110	1101
	$X_{5.3.2}$	156	0110	0100	1101	1001		$X_{5.4.2}$	140	0111	0100	1101	0001
	$X_{12.3.2}$	149	0110	1011	0010	1001		$X_{12.4.2}$	133	0111	1011	0010	0001
Set 7	$X_{4.1.1}$	253	0000	0011	0011	1111	Set 8	$X_{4.2.1}$	237	0001	0011	0011	0111
	$X_{13.1.1}$	244	0000	1100	1100	1111		$X_{13.2.1}$	228	0001	1100	1100	0111
	$X_{1.4.1}$	208	0011	0000	1111	0011		$X_{1.3.1}$	224	0010	0000	1111	1011
	$Y_{16.4.1}$	193	0011	1111	0000	0011		$X_{16.3.1}$	209	0010	1111	0000	1011
	$X_{7.2.2}$	170	0101	0110	1001	0101		$X_{7.1.2}$	186	0100	0110	1001	1101
	$X_{10.2.2}$	167	0101	1001	0110	0101		$X_{10.1.2}$	183	0100	1001	0110	1101
	$X_{6.3.2}$	155	0110	0101	0101	1001		$X_{6.4.2}$	139	0111	0101	0101	0001
	$X_{11.3.2}$	150	0110	1010	1010	1001		$X_{11.4.2}$	134	0111	1010	1010	0001
Set 9	$X_{5.1.1}$	252	0000	0100	1101	1111	Set 10	$X_{5.2.1}$	236	0001	0100	1101	0111
	$X_{12.1.1}$	245	0000	1011	0010	1111		$X_{12.2.1}$	229	0001	1011	0010	0111
	$X_{8.4.1}$	201	0011	0111	0001	0011		$X_{8.3.1}$	217	0010	0111	0001	1011
	$X_{9.4.1}$	200	0011	1000	1110	0011		$X_{9.3.1}$	216	0010	1000	1110	1011
	$X_{2.2.2}$	175	0101	0001	0111	0101		$X_{2.1.2}$	191	0100	0001	0111	1101
	$X_{15.2.2}$	162	0101	1110	1000	0101		$X_{15.1.2}$	178	0100	1110	1000	1101
	$X_{3.3.2}$	158	0110	0010	1011	1001		$X_{3.4.2}$	142	0111	0010	1011	0001
	$X_{14.3.2}$	147	0110	1101	0100	1001		$X_{14.4.2}$	131	0111	1101	0100	0001
Set 11	$X_{6.1.1}$	251	0000	0101	0101	1111	Set 12	$X_{6.2.1}$	235	0001	0101	0101	0111
	$X_{11.1.1}$	246	0000	1010	1010	1111		$X_{11.2.1}$	230	0001	1010	1010	0111
	$X_{7.4.1}$	202	0011	0110	1001	0011		$X_{7.3.1}$	218	0010	0110	1001	1011
	$X_{10.4.1}$	199	0011	1001	0110	0011		$X_{10.3.1}$	215	0010	1001	0110	1011
	$X_{4.2.2}$	173	0101	0011	0011	0101		$X_{4.1.2}$	189	0100	0011	0011	1101
	$X_{13.2.2}$	164	0101	1100	1100	0101		$X_{13.1.2}$	180	0100	1100	1100	1101
	$X_{1.3.2}$	160	0110	0000	1111	1001		$X_{1.4.2}$	144	0111	0000	1111	0001
	$X_{16.3.2}$	145	0110	1111	0000	1001		$X_{16.4.2}$	129	0111	1111	0000	0001
Set 13	$X_{7.1.1}$	250	0000	0110	1001	1111	Set 14	$X_{7.2.1}$	234	0001	0110	1001	0111
	$X_{10.1.1}$	247	0000	1001	0110	1111		$X_{10.2.1}$	231	0001	1001	0110	0111
	$X_{6.4.1}$	203	0011	0101	0101	0011		$X_{6.3.1}$	219	0010	0101	0101	1011
	$X_{11.4.1}$	198	0011	1010	1010	0011		$X_{11.3.1}$	214	0010	1010	1010	1011
	$X_{1.2.2}$	176	0101	0000	1111	0101		$X_{1.1.2}$	192	0100	0000	1111	1101
	$X_{16.2.2}$	161	0101	1111	0000	0101		$X_{16.1.2}$	177	0100	1111	0000	1101

	$X_{4.3.2}$	157	0110	0011	0011	1001		$X_{4.4.2}$	141	0111	0011	0011	0001
	$X_{13.3.2}$	148	0110	1100	1100	1001		$X_{13.4.2}$	132	0111	1100	1100	0001
Set 15	$X_{8.1.1}$	249	0000	0111	0001	1111	Set 16	$X_{8.2.1}$	233	0001	0111	0001	0111
	$X_{9.1.1}$	248	0000	1000	1110	1111		$X_{9.2.1}$	232	0001	1000	1110	0111
	$X_{5.4.1}$	204	0011	0100	1101	0011		$X_{5.3.1}$	220	0010	0100	1101	1011
	$X_{12.4.1}$	197	0011	1011	0010	0011		$X_{12.3.1}$	213	0010	1011	0010	1011
	$X_{3.2.2}$	174	0101	0010	1011	0101		$X_{3.1.2}$	190	0100	0010	1011	1101
	$X_{14.2.2}$	163	0101	1101	0100	0101		$X_{14.1.2}$	179	0100	1101	0100	1101
	$X_{2.3.2}$	159	0110	0001	0111	1001		$X_{2.4.2}$	143	0111	0001	0111	0001
	$X_{15.3.2}$	146	0110	1110	1000	1001		$X_{15.4.2}$	130	0111	1110	1000	0001

Table 10. The 16 spanning sets for the 4D space. Note that all columns in any set have four 1s and four 0s apart from the first (eight 0s) and the last (eight 1s). The 16 spanning sets can be easily generated by focussing on the last eight values and first considering the last four values of each row. All the possible combinations for four truth–values are 16: 1 for four 1s, 1 for four 0s, 4 for three 1s and one 1 and vice versa, 6 for two 1s and two 0s.

$$\begin{aligned}
X_{1.1} &= (X_{2.1}+X_{3.1})(X_{1.2}+X'_{4.2}) = (X_{2.1}+X_{1.2})(X_{3.1}+X'_{4.2}) = (X_{2.1}+X'_{4.2})(X_{3.1}+X_{1.2}),\\
X_{4.1} &= (X_{2.1}+X_{3.1})(X'_{1.2}+X_{4.2}) = (X_{2.1}+X'_{1.2})(X_{3.1}+X_{4.2}) = (X_{2.1}+X_{4.2})(X_{3.1}+X'_{1.2}),\\
X_{2.2} &= (X_{2.1}+X_{1.2})(X'_{3.1}+X_{4.2}) = (X_{2.1}+X_{4.2})(X'_{3.1}+X_{1.2}) = (X_{1.2}+X_{4.2})(X_{2.1}+X'_{3.1}),\\
X_{3.2} &= (X'_{2.1}+X_{1.2})(X_{3.1}+X_{4.2}) = (X'_{2.1}+X_{4.2})(X_{3.1}+X_{1.2}) = (X_{1.2}+X_{4.2})(X'_{2.1}+X_{3.1}),\\
X_{2.1} &= (X_{1.1}+X_{4.1})(X_{2.2}+X'_{3.2}) = (X_{1.1}+X_{2.2})(X_{4.1}+X'_{3.2}) = (X_{4.1}+X_{2.2})(X_{1.1}+X'_{4.2}),\\
X_{3.1} &= (X_{1.1}+X_{4.1})(X'_{2.2}+X_{3.2}) = (X_{1.1}+X'_{2.2})(X_{4.1}+X_{3.2}) = (X_{4.1}+X'_{2.2})(X_{1.1}+X_{3.2}),\\
X_{1.2} &= (X_{1.1}+X_{2.2})(X'_{4.1}+X_{3.2}) = (X_{1.1}+X_{3.2})(X'_{4.1}+X_{2.2}) = (X_{1.1}+X'_{4.1})(X_{2.2}+X_{3.2}),\\
X_{4.2} &= (X'_{1.1}+X_{2.2})(X_{4.1}+X_{3.2}) = (X'_{1.1}+X_{3.2})(X_{4.1}+X_{2.2}) = (X'_{1.1}+X_{4.1})(X_{2.2}+X_{3.2}).
\end{aligned} \tag{21}$$

References

[1] G. Auletta, *Mechanical Logic in three-Dimensional Space*, Pan Stanford Pub., Singapore (2013).

[2] G. Auletta, *A New Way To implement Quantum Computation*, Journal of Quantum Information Science **3.4**, 127–37 (2013).

[3] G. Auletta, M. Fortunato, G. Parisi, *Quantum Mechanics*, Cambridge University Press (2009); rev. paperback ed. (2013).

[4] S. Awodey, *Category Theory*, Oxford University Press (2006).

[5] G. Boole, *An Investigation of the Laws of Thought, on which are Founded the Mathematical Theories of Logic and Probabilities*, Walton and Maberly, London (1854); Dover, New York (1958).

[6] K. Gödel, *Über formal unentscheidbar Sätze der Principia Mathematica und verwandter Systeme. I'*, Monatshefte für Mathematik und Physik **38**, 173–98 (1931).

[7] G. Peano, *Arithmetices principia, novo methodo exposita*, Torino (1889).

[8] J. Piaget, *Traité de Logique Opératoire* (1949); 2nd ed. (1972).

[9] Russell, Bertrand, *Principles of Mathematics* (1903), 2nd ed. (1937); rep. Routledge, London (1992).

[10] D. I. Spivak, *Category Theory for Scientists*, MIT Press, Cambridge, MA (2013).

1	11111110	$X'_{1.1}+X'_{4.1}+X'_{2.2}=X'_{1.1}+X'_{4.1}+X'_{3.2}=X'_{1.1}+X'_{2.2}+X'_{3.2}=X'_{4.1}+X'_{2.2}+X'_{3.2}$ $X'_{2.1}+X'_{3.1}+X'_{1.2}=X'_{2.1}+X'_{3.1}+X'_{4.2}=X'_{2.1}+X'_{1.2}+X'_{4.2}=X'_{3.1}+X'_{1.2}+X'_{4.2}$
2	11111101	$X'_{1.1}+X'_{4.1}+X_{2.2}=X'_{1.1}+X'_{4.1}+X_{3.2}=X_{2.2}+X_{3.2}+X'_{1.1}=X_{2.2}+X_{3.2}+X'_{4.1}$ $X'_{2.1}+X'_{3.1}+X_{1.2}=X'_{2.1}+X'_{3.1}+X_{4.2}=X_{1.2}+X_{4.2}+X'_{2.1}=X_{1.2}+X_{4.2}+X'_{3.1}$
3	11111011	$X'_{1.1}+X'_{2.2}+X_{4.1}=X'_{1.1}+X'_{2.2}+X_{3.2}=X_{4.1}+X_{3.2}+X'_{1.1}=X_{4.1}+X'_{2.2}+X_{3.2}$ $X'_{2.1}+X'_{1.2}+X_{3.1}=X'_{2.1}+X'_{1.2}+X_{4.2}=X_{3.1}+X_{4.2}+X'_{2.1}=X_{3.1}+X'_{1.2}+X_{4.2}$
4	11110111	$X'_{1.1}+X'_{3.2}+X_{4.1}=X'_{1.1}+X_{2.2}+X'_{3.2}=X_{4.1}+X_{2.2}+X'_{1.1}=X_{4.1}+X_{2.2}+X'_{3.2}$ $X'_{2.1}+X'_{4.2}+X_{3.1}=X'_{2.1}+X_{1.2}+X'_{4.2}=X_{3.1}+X_{1.2}+X'_{2.1}=X_{3.1}+X_{1.2}+X'_{4.2}$
5	11101111	$X_{1.1}+X_{3.2}+X'_{4.1}=X_{1.1}+X'_{2.2}+X_{3.2}=X'_{4.1}+X'_{2.2}+X_{1.1}=X'_{4.1}+X'_{2.2}+X_{3.2}$ $X_{2.1}+X_{4.2}+X'_{3.1}=X_{2.1}+X'_{1.2}+X_{4.2}=X'_{3.1}+X'_{1.2}+X_{2.1}=X'_{3.1}+X'_{1.2}+X_{4.2}$
6	11011111	$X_{1.1}+X_{2.2}+X'_{4.1}=X_{1.1}+X_{2.2}+X'_{3.2}=X'_{4.1}+X'_{3.2}+X_{1.1}=X'_{4.1}+X_{2.2}+X'_{3.2}$ $X_{2.1}+X_{1.2}+X'_{3.1}=X_{2.1}+X_{1.2}+X'_{4.2}=X'_{3.1}+X'_{4.2}+X_{2.1}=X'_{3.1}+X_{1.2}+X'_{4.2}$
7	10111111	$X_{1.1}+X_{4.1}+X'_{2.2}=X_{1.1}+X_{4.1}+X'_{3.2}=X'_{2.2}+X'_{3.2}+X_{1.1}=X'_{2.2}+X'_{3.2}+X_{4.1}$ $X_{2.1}+X_{3.1}+X'_{1.2}=X_{2.1}+X_{3.1}+X'_{4.2}=X'_{1.2}+X'_{4.2}+X_{2.1}=X'_{1.2}+X'_{4.2}+X_{3.1}$
8	01111111	$X_{1.1}+X_{4.1}+X_{2.2}=X_{1.1}+X_{4.1}+X_{3.2}=X_{1.1}+X_{2.2}+X_{3.2}=X_{4.1}+X_{2.2}+X_{3.2}$ $X_{2.1}+X_{3.1}+X_{1.2}=X_{2.1}+X_{3.1}+X_{4.2}=X_{2.1}+X_{1.2}+X_{4.2}=X_{3.1}+X_{1.2}+X_{4.2}$

Table 11. The 8 statements of Level 7-1 for the 3D space.

1	$11111100 = 11111110 \times 11111101$	$X'_{1.1}+X'_{4.1}=X'_{2.1}+X'_{3.1}$
2	$11111010 = 11111110 \times 11111011$	$X'_{1.1}+X'_{2.2}=X'_{2.1}+X'_{1.2}$
3	$11111001 = 11111101 \times 11111011$	$X'_{1.1}+X_{3.2}=X'_{2.1}+X_{4.2}$
4	$11110110 = 11111110 \times 11110111$	$X'_{1.1}+X'_{3.2}=X'_{2.1}+X'_{4.2}$
5	$11110101 = 11111101 \times 11110111$	$X'_{1.1}+X_{2.2}=X_{3.1}+X_{4.2}$
6	$11110011 = 11111011 \times 11110111$	$X'_{1.1}+X_{4.1}=X_{1.2}+X'_{4.2}$
7	$11101110 = 11111110 \times 11101111$	$X'_{4.1}+X'_{2.2}=X'_{2.1}+X'_{4.2}$
8	$11101101 = 11111101 \times 11101111$	$X'_{4.1}+X_{3.2}=X'_{2.1}+X_{1.2}$
9	$11101011 = 11111011 \times 11101111$	$X'_{2.2}+X_{3.2}=X'_{2.1}+X_{3.1}$
10	$11100111 = 11110111 \times 11101111$	$(X'_{1.1}+X_{4.1}+X'_{3.2})(X_{1.1}+X'_{4.1}+X_{3.2})=(X'_{2.1}+X_{3.1}+X'_{4.2})(X_{2.1}+X'_{3.1}+X_{4.2})$
11	$11011110 = 11111111 \times 11011111$	$X'_{4.1}+X'_{3.2}=X'_{3.1}+X'_{4.2}$
12	$11011101 = 11111101 \times 11011111$	$X'_{4.1}+X_{2.2}=X'_{3.1}+X_{1.2}$
13	$11011011 = 11111011 \times 11011111$	$(X'_{1.1}+X_{4.1}+X'_{2.2})(X_{1.1}+X'_{4.1}+X_{2.2})=(X'_{2.1}+X_{3.1}+X'_{1.2})(X_{2.1}+X'_{3.1}+X_{1.2})$
14	$11010111 = 11110111 \times 11011111$	$X_{2.1}+X'_{3.1}=X_{2.2}+X'_{3.2}$
15	$11001111 = 11101111 \times 11011111$	$X_{1.1}+X'_{4.1}=X_{1.2}+X'_{4.2}$
16	$10111110 = 11111110 \times 10111111$	$X'_{2.2}+X'_{3.2}=X'_{1.2}+X'_{4.2}$
17	$10111101 = 11111101 \times 10111111$	$(X'_{1.1}+X'_{4.1}+X_{3.2})(X_{1.1}+X_{4.1}+X'_{3.2})=(X'_{2.1}+X'_{3.1}+X_{4.2})(X_{2.1}+X_{3.1}+X'_{4.2})$
18	$10111011 = 11111011 \times 10111111$	$X_{4.1}+X'_{2.2}=X_{3.1}+X'_{1.2}$
19	$10110111 = 11110111 \times 10111111$	$X_{4.1}+X'_{3.2}=X_{2.1}+X'_{1.2}$
20	$10101111 = 11101111 \times 10111111$	$X_{1.1}+X'_{2.2}=X_{3.1}+X'_{4.2}$
21	$10011111 = 11011111 \times 10111111$	$X_{1.1}+X'_{3.2}=X_{2.1}+X'_{4.2}$
22	$01111110 = 11111110 \times 01111111$	$(X'_{1.1}+X'_{4.1}+X'_{3.2})(X_{1.1}+X_{4.1}+X_{3.2})=(X'_{2.1}+X'_{3.1}+X'_{4.2})(X_{2.1}+X_{3.1}+X_{4.2})$
23	$01111101 = 11111101 \times 01111111$	$X_{2.2}+X_{3.2}=X_{1.2}+X_{4.2}$
24	$01111011 = 11111011 \times 01111111$	$X_{4.1}+X_{3.2}=X_{3.1}+X_{4.2}$
25	$01110111 = 11110111 \times 01111111$	$X_{4.1}+X_{2.2}=X_{2.1}+X_{4.2}$
26	$01101111 = 11101111 \times 01111111$	$X_{1.1}+X_{3.2}=X_{3.1}+X_{1.2}$
27	$01011111 = 11011111 \times 01111111$	$X_{1.1}+X_{2.2}=X_{2.1}+X_{1.2}$
28	$00111111 = 10111111 \times 01111111$	$X_{1.1}+X_{4.1}=X_{2.1}+X_{3.1}$

Table 12. The 28 statements of Level 6-2.

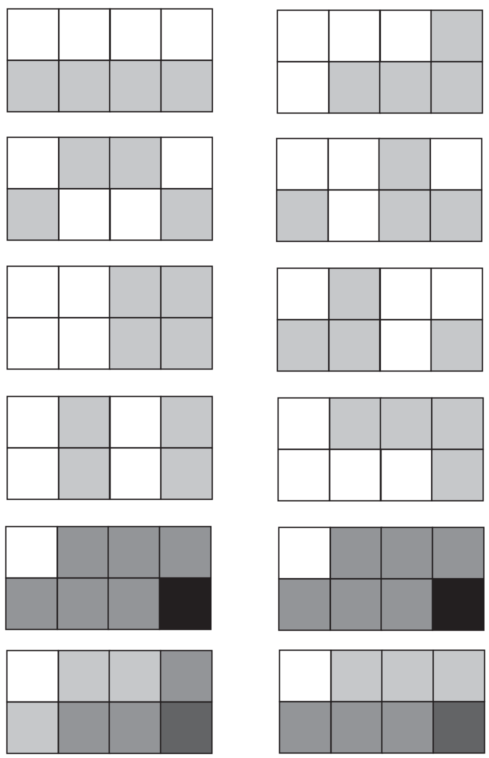

Figure 6. The two columns represent the two spanning sets for the 3D space displayed in Tab. 7. The first 4 rectangles in each column are the 4 generators pertaining to the set. The 4 squares constituting the first row of each rectangle are the areas a,b,c,d, while the 4 squares constituting the second row are areas e,f,g,h. The second last rectangle in each column represents the combination of the 4 variables of the set. Note that all spanning sets must keep this character: 1 square white, 1 black and all others superposition of half the number of variables, i.e. two areas. The last rectangle in each column represents an example of basis: first, third and fourth variable for the column on the left and the first 3 variables for the column on the right (the first case corresponds to the assignment displayed in Fig. 1). The grayscale reflects the intersection areas: in particular, in the last two rectangles we can see no area, single area, superposition of two areas, superposition of three areas, according to the rule (17).