

A New Way to Implement Quantum Computation

Gennaro Auletta

University of Cassino, Cassino, Italy
 Email: gennaro.auletta@gmail.com

Received September 10, 2013; revised October 23, 2013; accepted November 8, 2013

Copyright © 2013 Gennaro Auletta. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT

In this paper, I shall sketch a new way to consider a Lindenbaum-Tarski algebra as a 3D logical space in which any one (of the 256 statements) occupies a well-defined position and it is identified by a numerical ID. This allows pure mechanical computation both for generating rules and inferences. It is shown that this abstract formalism can be geometrically represented with logical spaces and subspaces allowing a vectorial representation. Finally, it shows the application to quantum computing through the example of three coupled harmonic oscillators.

Keywords: Lindenbaum-Tarski Algebra; 3D Logical Space; Mechanical Computation; Inference; Quantum Computing; Raising Operators; Lowering Operators

1. Introduction

I shall present an expansion of the Lindenbaum-Tarski algebra into a three-dimensional logical space [1-3]. The Venn diagrams are shown in **Figure 1** while the relative truth-value assignment is in **Table 1**. Let us compute the combinatory of this logical space, considering that we have 8 truth-value assignment and therefore statements defined through a string of 8 numbers.

A combinatorial calculus gives 9 logical levels determined by the number of 1s or 0s (where the first figure denotes the number of 1s while the second open the numbers of 0s):

$$C(8,0) = 1, C(8,1) = 8, C(8,2) = 28,$$

$$C(8,3) = 56, C(8,4) = 70, C(8,5) = 56,$$

$$C(8,6) = 28, C(8,7) = 8, C(8,8) = 1.$$

In this way, any (of the 256) statements occupy a well definite position and it is identified by a numerical ID. This allows pure mechanical computation both for generating rules and inferences. Let us therefore represent all the propositions at the different levels through **Tables 2-10** (the expression f'h' means the product of the complement sets of f and h).

2. Subspaces

A three-dimensional logical space embeds three two-dimensional logical subspaces and three one-dimensional ones. A one-dimensional (1D) subspace is constituted by

the following nodes: $X \vee \neg X$, X , $\neg X$, and $X \wedge \neg X$. In the following, I shall use a number for each of the nodes X and $\neg X$ and do not consider tautology and contradiction as long as we deal with subspaces (in fact, there is every time only one tautology and one contradiction for the whole space). This has the purpose to fully replace the calculus on variables with a

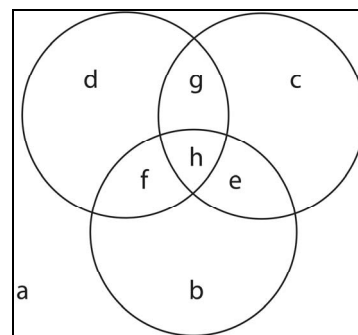


Figure 1. The Venn diagrams when the relations among three statements are considered. Note that we have: $X = d + f + g + h$, $Y = c + e + g + h$, $Z = b + e + f + h$.

Table 1. Inputs and outputs of 3-dimensional logic.

X	0	0	0	1	0	1	1	1
Y	0	0	1	0	1	0	1	1
Z	0	1	0	0	1	1	0	1
output	a	b	c	d	e	f	g	h

Table 2. C(8,8) = C(8,0).

11111111	a + b + c + d + e + f + g + h
----------	-------------------------------

Table 3. C(8,7) = 8 disjunctive eptaplets, C(8, 1) = 8 conjunctive singlets.

1.	11111110	a + b + c + d + e + f + g	h'	2.	11111101	a + b + c + d + e + f + h	g'
3.	11111011	a + b + c + d + e + g + h	f'	4.	11110111	a + b + c + d + f + g + h	e'
5.	11101111	a + b + c + e + f + g + h	d'	6.	11011111	a + b + d + e + f + g + h	c'
7.	10111111	a + c + d + e + f + g + h	b'	8.	01111111	b + c + d + e + f + g + h	a'

Table 4. C(8,6) = 28 disjunctive esaplets, C(8,2) = 28 conjunctive duplets.

1.	11111100	a + b + c + d + e + f	g'h'	2.	11111010	a + b + c + d + e + g	fh'
3.	11111001	a + b + c + d + e + h	f'g'	4.	11110110	a + b + c + d + f + g	e'h'
5.	11110101	a + b + c + d + f + h	e'g'	6.	11110011	a + b + c + d + g + h	e'f'
7.	11101110	a + b + c + e + f + g	d'h'	8.	11101101	a + b + c + e + f + h	d'g'
9.	11101011	a + b + c + e + g + h	d'f'	10.	11100111	a + b + c + f + g + h	d'e'
11.	11011110	a + b + d + e + f + g	c'h'	12.	11011101	a + b + d + e + f + h	c'g'
13.	11011011	a + b + d + e + g + h	c'f'	14.	11010111	a + b + d + f + g + h	c'e'
15.	11001111	a + b + e + f + g + h	c'd'	16.	10111110	a + c + d + e + f + g	b'h'
17.	10111101	a + c + d + e + f + h	b'g'	18.	10111011	a + c + d + e + g + h	b'f'
19.	10110111	a + c + d + f + g + h	b'e'	20.	10101111	a + c + e + f + g + h	b'd'
21.	10011111	a + d + e + f + g + h	b'c'	22.	01111110	b + c + d + e + f + g	a'h'
23.	01111101	b + c + d + e + f + h	a'g'	24.	01111011	b + c + d + e + g + h	a'f'
25.	01110111	b + c + d + f + g + h	a'e'	26.	01101111	b + c + e + f + g + h	a'd'
27.	01011111	b + d + e + f + g + h	a'c'	28.	00111111	c + d + e + f + g + h	a'b'

calculus on the subspaces. Therefore, I shall assign the numbers 1 and 2 to nodes $\neg X$ and X , respectively, and call each one the *complement* zero-dimensional (0D) space of the other. For the sake of simplicity, here and in the following I shall call also 1 or 2 simply *subspaces* although they are in fact nodes of a single one-dimensional (1D) subspace or 0D spaces. Keeping this in mind, that widening of the term makes no problem but helps the simplification of the language. Therefore, the one-dimensional subspace can be represented as a thread connecting the nodes 1 and 2 as shown in **Figure 2**. Therefore, let us assign a (0D) subspace to each of the values displayed in **Table 1** for each variable:

The six one-dimensional subspaces are therefore 1 - 2, 7 - 8 for X , 3 - 4, 9 - 10 for Y and 5 - 6, 11 - 12 for Z . This generates three two-dimensional (2D) subspaces

that are therefore embedded in the three-dimensional logical space:

As we can see in **Figure 3**, a single 2D space is build by creating a surface connecting the two 1D subspaces and therefore all their nodes. In order to generate all the possible three-dimensional combinations we need first to understand what means to be in a 3D space. Let us first understand what is the correct relation between subspaces and variables. For one-dimensional subspaces they obviously coincide. In this case, in fact the variable X ($\neg X$) is spanned by the subspace 2 (1). For the two-dimensional subspaces the same remains true, although here there is a multiplication of subspaces relative to the variables. However, as long as these 2D subspaces remain separated, the situation is not substantially different. However, in the three-dimensional space

Table 5. $C(8,5) = 56$ disjunctive pentaplets, $C(8,3) = 56$ conjunctive triplets.

1.	11111000	$a + b + c + d + e$	$f'g'h'$	2.	11110100	$a + b + c + d + f$	$e'g'h'$
3.	11110010	$a + b + c + d + g$	$e'fh'$	4.	11110001	$a + b + c + d + h$	$e'fg'$
5.	11101100	$a + b + c + e + f$	$d'g'h'$	6.	11101010	$a + b + c + e + g$	$d'fh'$
7.	11101001	$a + b + c + e + h$	$d'fg'$	8.	11100110	$a + b + c + f + g$	$d'e'h'$
9.	11100101	$a + b + c + f + h$	$d'e'g'$	10.	11100011	$a + b + c + g + h$	$d'e'f'$
11.	11011100	$a + b + d + e + f$	$c'g'h'$	12.	11011010	$a + b + d + e + g$	$c'fh'$
13.	11011001	$a + b + d + e + h$	$c'fg'$	14.	11010110	$a + b + d + f + g$	$c'e'h'$
15.	11010101	$a + b + d + f + h$	$c'e'g'$	16.	11010011	$a + b + d + g + h$	$c'e'f'$
17.	11001110	$a + b + e + f + g$	$c'd'h'$	18.	11001101	$a + b + e + f + h$	$c'd'g'$
19.	11001011	$a + b + e + g + h$	$c'd'f'$	20.	11000111	$a + b + f + g + h$	$c'd'e'$
21.	10111100	$a + c + d + e + f$	$b'g'h'$	22.	10111010	$a + c + d + e + g$	$b'fh'$
23.	10111001	$a + c + d + e + h$	$b'fg'$	24.	10110110	$a + c + d + f + g$	$b'e'h'$
25.	10110101	$a + c + d + f + h$	$b'e'g'$	26.	10110011	$a + c + d + g + h$	$b'e'f'$
27.	10101110	$a + c + e + f + g$	$b'd'h'$	28.	10101101	$a + c + e + f + h$	$b'd'g'$
29.	10101011	$a + c + e + g + h$	$b'd'f'$	30.	10100111	$a + c + f + g + h$	$b'd'e'$
31.	10011110	$a + d + e + f + g$	$b'c'h'$	32.	10011101	$a + d + e + f + h$	$b'c'g'$
33.	10011011	$a + d + e + g + h$	$b'cf'$	34.	10010111	$a + d + f + g + h$	$b'ce'$
35.	10001111	$a + e + f + g + h$	$b'cd'$	36.	01111100	$b + c + d + e + f$	$a'g'h'$
37.	01111010	$b + c + d + e + g$	$a'fh'$	38.	01111001	$b + c + d + e + h$	$a'fg'$
39.	01110110	$b + c + d + f + g$	$a'e'h'$	40.	01110101	$b + c + d + f + h$	$a'e'g'$
41.	01110011	$b + c + d + g + h$	$a'e'f'$	42.	01101110	$b + c + e + f + g$	$a'd'h'$
43.	01101101	$b + c + e + f + h$	$a'd'g'$	44.	01101011	$b + c + e + g + h$	$a'd'f'$
45.	01100111	$b + c + f + g + h$	$a'd'e'$	46.	01011110	$b + d + e + f + g$	$a'c'h'$
47.	01011101	$b + d + e + f + h$	$a'c'g'$	48.	01011011	$b + d + e + g + h$	$a'cf'$
49.	01010111	$b + d + f + g + h$	$a'ce'$	50.	01001111	$b + e + f + g + h$	$a'cd'$
51.	00111110	$c + d + e + f + g$	$a'b'h'$	52.	00111101	$c + d + e + f + h$	$a'b'g'$
53.	00111011	$c + d + e + g + h$	$a'b'f'$	54.	00110111	$c + d + f + g + h$	$a'be'$
55.	00101111	$c + e + f + g + h$	$a'bd'$	56.	00011111	$d + e + f + g + h$	$a'b'c'$

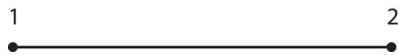


Figure 2. One-dimensional space.

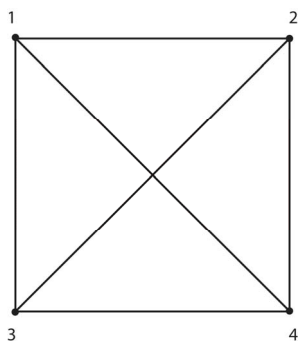


Figure 3. Two-dimensional space.

all variables and subspaces can be connected. In fact, we have e.g. that at the level 4 - 4 displayed in **Table 6** we have the following combination of subspaces from below

$$[1 \otimes (10 \oplus 12)] \oplus [1 \otimes (9 \oplus 12)] \\ \oplus [1 \otimes (10 \oplus 11)] \oplus [1 \otimes (9 \oplus 11)],$$

where, when building subspaces, I shall consider the operation of sum (\oplus) and product (\otimes) of these subspaces; and from above

$$[1 \oplus (10 \otimes 12)] \otimes [1 \oplus (9 \otimes 12)] \\ \otimes [1 \oplus (10 \otimes 11)] \otimes [1 \oplus (9 \otimes 11)]$$

These two expressions, somehow single out the subspace 1. However, since the subspace 1 is outside the 2D subspace represented by the intersection 9 - 11, 10 -

Table 6. $C(8,4) = 70$ disjunctive quadruplets, $C(8,4) = 70$ conjunctive quadruplets.

1.	11110000	$a + b + c + d$	$e'f'g'h'$	2.	11101000	$a + b + c + e$	$d'f'g'h'$
3.	11100100	$a + b + c + f$	$d'e'g'h'$	4.	11100010	$a + b + c + g$	$d'e'f'h'$
5.	11100001	$a + b + c + h$	$d'e'f'g'$	6.	11011000	$a + b + d + e$	$c'f'g'h'$
7.	11010100	$a + b + d + f$	$c'e'g'h'$	8.	11010010	$a + b + d + g$	$c'e'f'h'$
9.	11010001	$a + b + d + h$	$c'e'f'g'$	10.	11001100	$a + b + e + f$	$c'd'g'h'$
11.	11001010	$a + b + e + g$	$c'd'f'h'$	12.	11001001	$a + b + e + h$	$c'd'f'g'$
13.	11000110	$a + b + f + g$	$c'd'e'h'$	14.	11000101	$a + b + f + h$	$c'd'e'g'$
15.	11000011	$a + b + g + h$	$c'd'e'f'$	16.	10111000	$a + c + d + e$	$b'f'g'h'$
17.	10110100	$a + c + d + f$	$b'e'g'h'$	18.	10110010	$a + c + d + g$	$b'e'f'h'$
19.	10110001	$a + c + d + h$	$b'e'f'g'$	20.	10101100	$a + c + e + f$	$b'd'g'h'$
21.	10101010	$a + c + e + g$	$b'd'f'h'$	22.	10101001	$a + c + e + h$	$b'd'f'g'$
23.	10100110	$a + c + f + g$	$b'd'e'h'$	24.	10100101	$a + c + f + h$	$b'd'e'g'$
25.	10100011	$a + c + g + h$	$b'd'e'f'$	26.	10011100	$a + d + e + f$	$b'c'g'h'$
27.	10011010	$a + d + e + g$	$b'c'f'h'$	28.	10011001	$a + d + e + h$	$b'c'f'g'$
29.	10010110	$a + d + f + g$	$b'c'e'h'$	30.	10010101	$a + d + f + h$	$b'c'e'g'$
31.	10010011	$a + d + g + h$	$b'c'e'f'$	32.	10001110	$a + e + f + g$	$b'c'd'h'$
33.	10001101	$a + e + f + h$	$b'c'd'g'$	34.	10001011	$a + e + g + h$	$b'c'd'f'$
35.	10000111	$a + f + g + h$	$b'c'd'e'$	36.	01111000	$b + c + d + e$	$a'f'g'h'$
37.	01110100	$b + c + d + f$	$a'e'g'h'$	38.	01110010	$b + c + d + g$	$a'e'f'h'$
39.	01110001	$b + c + d + h$	$a'e'f'g'$	40.	01101100	$b + c + e + f$	$a'd'g'h'$
41.	01101010	$b + c + e + g$	$a'd'f'h'$	42.	01101001	$b + c + e + h$	$a'd'f'g'$
43.	01100110	$b + c + f + g$	$a'd'e'h'$	44.	01100101	$b + c + f + h$	$a'd'e'g'$
45.	01100011	$b + c + g + h$	$a'd'e'f'$	46.	01011100	$b + d + e + f$	$a'c'g'h'$
47.	01011010	$b + d + e + g$	$a'c'f'h'$	48.	01011001	$b + d + e + h$	$a'c'f'g'$
49.	01010110	$b + d + f + g$	$a'c'e'h'$	50.	01010101	$b + d + f + h$	$a'c'e'g'$
51.	01010011	$b + d + g + h$	$a'c'e'f'$	52.	01001110	$b + e + f + g$	$a'c'd'h'$
53.	01001101	$b + e + f + h$	$a'c'd'g'$	54.	01001011	$b + e + g + h$	$a'c'd'f'$
55.	01000111	$b + f + g + h$	$a'c'd'e'$	56.	00111100	$c + d + e + f$	$a'b'g'h'$
57.	00111010	$c + d + e + g$	$a'b'f'h'$	58.	00111001	$c + d + e + h$	$a'b'f'g'$
59.	00110110	$c + d + f + g$	$a'b'e'h'$	60.	00110101	$c + d + f + h$	$a'b'e'g'$
61.	00110011	$c + d + g + h$	$a'b'e'f'$	62.	00101110	$c + e + f + g$	$a'b'd'h'$
63.	00101101	$c + e + f + h$	$a'b'd'g'$	64.	00101011	$c + e + g + h$	$a'b'd'f'$
65.	00100111	$c + f + g + h$	$a'b'd'e'$	66.	00011110	$d + e + f + g$	$a'b'c'h'$
67.	00011101	$d + e + f + h$	$a'b'c'g'$	68.	00011011	$d + e + g + h$	$a'b'c'f'$
69.	00010111	$d + f + g + h$	$a'b'c'e'$	70.	00001111	$e + f + g + h$	$a'b'c'd'$

Table 7. $C(8,3) = 56$ disjunctive triplets, $C(8,5) = 56$ conjunctive pentaplets.

1.	11100000	a + b + c	d'e'f'g'h'	2.	11010000	a + b + d	c'e'f'g'h'
3.	11001000	a + b + e	c'd'f'g'h'	4.	11000100	a + b + f	c'd'e'g'h'
5.	11000010	a + b + g	c'd'e'f'h'	6.	11000001	a + b + h	c'd'e'f'g'
7.	10110000	a + c + d	b'e'f'g'h'	8.	10101000	a + c + e	b'd'f'g'h'
9.	10100100	a + c + f	b'd'e'g'h'	10.	10100010	a + c + g	b'c'e'f'h'
11.	10100001	a + c + h	b'd'e'f'g'	12.	10011000	a + d + e	b'c'f'g'h'
13.	10010100	a + d + f	b'c'e'g'h'	14.	10010010	a + d + g	b'c'e'f'h'
15.	10010001	a + d + h	b'c'e'f'g'	16.	10001100	a + e + f	b'c'd'g'h'
17.	10001010	a + e + g	b'c'd'f'h'	18.	10001001	a + e + h	b'c'd'f'g'
19.	10000110	a + f + g	b'c'd'e'h'	20.	10000101	a + f + h	b'c'd'e'g'
21.	10000011	a + g + h	b'c'd'e'f'	22.	01110000	b + c + d	a'e'f'g'h'
23.	01101000	b + c + e	a'd'f'g'h'	24.	01100100	b + c + f	a'd'e'g'h'
25.	01100010	b + c + g	a'd'e'f'h'	26.	01100001	b + c + h	a'd'e'f'g'
27.	01011000	b + d + e	a'c'f'g'h'	28.	01010100	b + d + f	a'c'e'g'h'
29.	01010010	b + d + g	a'c'e'f'h'	30.	01010001	b + d + h	a'c'e'f'g'
31.	01001100	b + e + f	a'c'd'g'h'	32.	01001010	b + e + g	a'c'd'f'h'
33.	01001001	b + e + h	a'c'd'f'g'	34.	01000110	b + f + g	a'c'd'e'h'
35.	01000101	b + f + h	a'c'd'e'g'	36.	01000011	b + g + h	a'c'd'e'f'
37.	00111000	c + d + e	a'b'f'g'h'	38.	00110100	c + d + f	a'b'e'g'h'
39.	00110010	c + d + g	a'b'e'f'h'	40.	00110001	c + d + h	a'b'e'f'g'
41.	00101100	c + e + f	a'b'd'g'h'	42.	00101010	c + e + g	a'b'd'f'h'
43.	00101001	c + e + h	a'b'd'f'g'	44.	00100110	c + f + g	a'b'd'e'h'
45.	00100101	c + f + h	a'b'd'e'g'	46.	00100011	c + g + h	a'b'd'e'f'
47.	00011100	d + e + f	a'b'c'g'h'	48.	00011010	d + e + g	a'b'c'f'h'
49.	00011001	d + e + h	a'b'c'f'g'	50.	00010110	d + f + g	a'b'c'e'h'
51.	00010101	d + f + h	a'b'c'e'g'	52.	00010011	d + g + h	a'b'c'e'f'
53.	00001110	e + f + g	a'b'c'd'h'	54.	00001101	e + f + h	a'b'c'd'g'
55.	00001011	e + g + h	a'b'c'd'f'	56.	00000111	f + g + h	a'b'c'd'e'

Table 8. $C(8,2) = 28$ disjunctive duplets, $C(8,6) = 28$ conjunctive esaplets.

1.	11000000	a + b	c'd'e'f'g'h'	2.	10100000	a + c	b'd'e'f'g'h'
3.	10010000	a + d	b'c'e'f'g'h'	4.	10001000	a + e	b'c'd'f'g'h'
5.	10000100	a + f	b'c'd'e'g'h'	6.	10000010	a + g	b'c'd'e'f'h'
7.	10000001	a + h	b'c'd'e'f'g'	8.	01100000	b + c	a'd'e'f'g'h'
9.	01010000	b + d	a'c'e'f'g'h'	10.	01001000	b + e	a'c'd'f'g'h'
11.	01000100	b + f	a'c'd'e'g'h'	12.	01000010	b + g	a'c'd'e'f'h'
13.	01000001	b + h	a'c'd'e'f'g'	14.	00110000	c + d	a'b'e'f'g'h'
15.	00101000	c + e	a'b'd'f'g'h'	16.	00100100	c + f	a'b'd'e'g'h'
17.	00100010	c + g	a'b'd'e'f'h'	18.	00100001	c + h	a'b'd'e'f'g'
19.	00011000	d + e	a'b'c'f'g'h'	20.	00010100	d + f	a'b'c'e'g'h'
21.	00010010	d + g	a'b'c'e'f'h'	22.	00010001	d + h	a'b'c'e'f'g'
23.	00001100	e + f	a'b'c'd'g'h'	24.	00001010	e + g	a'b'c'd'f'h'
25.	00001001	e + h	a'b'c'd'f'g'	26.	00000110	f + g	a'b'c'e'f'h'
27.	00000101	f + h	a'b'c'd'e'g'	28.	00000011	g + h	a'b'c'd'e'f'

Table 9. C(8,1) = 8 disjunctive singlets, C(8,7) = 8 conjunctive eptaplets.

1.	10000000	a	b'c'd'e'f'g'h'	2.	01000000	b	a'c'd'e'f'g'h'
3.	00100000	c	a'b'd'e'f'g'h'	4.	00010000	d	a'b'c'e'f'g'h'
5.	00001000	e	a'b'c'd'f'g'h'	6.	00000100	f	a'b'c'd'e'g'h'
7.	00000010	g	a'b'c'd'e'f'h'	8.	00000001	h	a'b'c'd'e'f'g'

Table 10. C(8,0) = C(8,8).

00000000	a'b'c'd'e'f'g'h'
----------	------------------

Table 11. One-dimensional subspaces in the tridimensional space.

	a	b	c	d	e	f	g	h
X	1	1	1	2	1	2	2	2
	7	7	7	8	7	8	8	8
Y	3	3	4	3	4	3	4	4
	9	9	10	9	10	9	10	10
Z	5	6	5	5	6	6	5	6
	11	12	11	11	12	12	11	12

Table 12. Two-dimensional subspaces in the tridimensional space.

	a	b	c	d	e	f	g	h
X	1	1	1	2	1	2	2	2
Y	3	3	4	3	4	3	4	4
X	7	7	7	8	7	8	8	8
Z	5	6	5	5	6	6	5	6
Y	9	9	10	9	10	9	10	10
Z	11	12	11	11	12	12	11	12

12, then we could also substitute 1 by 7 and write from below

$$[7 \otimes (10 \oplus 12)] \oplus [7 \otimes (9 \oplus 12)] \\ \oplus [7 \otimes (10 \oplus 11)] \oplus [7 \otimes (9 \oplus 11)],$$

and from above

$$[7 \oplus (10 \otimes 12)] \otimes [7 \oplus (9 \otimes 12)] \\ \otimes [7 \oplus (10 \otimes 11)] \otimes [7 \oplus (9 \otimes 11)].$$

This shows that both 1 and 7 point to the same variable $\neg X$ and therefore we can consider the variable X ($\neg X$) to be the coincidence of these two subspaces. Let us symbolize it as $2 = 8, 1 = 7$, and similarly for the other variables. Therefore, when we deal with connections through the 3D space we always have these cross spaces. This is crucial, since in this way we are able to explicitly symbolize the *range* of each variable when it is taken in connection with other ones. This is what makes the use of subspaces more rigorous than those of

variables and allows us both to drop any use of quantification and deal with generalized relations. Therefore, the 3D space takes the form of a pentahedron with three surfaces representing the three 2D subspaces that generate three triangles in which the three vertexes are nodes of all the three variables, as displayed in **Figure 4**.

From a geometrical point of view, each $n-1D$ subspace can be seen as a projection of the nD overordined space. This allows us to make use of partial derivatives with the additional difficulty that the planes we deal with are not orthogonal.

This formalism allows in principle to treat subspaces as the main tool of logic and therefore as boxes in which we can insert some content, and the variables as “dummy” variables that can be put in those boxes according to needs of computation. Nevertheless I shall still keep the use of variables as far as it can help the reader to assimilate this new way to treat logic. I also remark that in order to ascertain how many subspaces we have for a n -dimensional logic, it suffices to multiply the

number of subspaces of the $n-1$ -dimensional logic with the number of variables of the n -dimensional logic. This makes the following series:

- 1 1D subspace (2 0D ones) for the one-dimensional logical space,
- $1 \times 2 = 2$ 1D (4 0D) subspaces for the two-dimensional logical space,
- $2 \times 3 = 6$ 1D (12 0D) subspaces for the three-dimensional logical space,
- $6 \times 4 = 24$ 1D (48 0D) subspaces for the four-dimensional logical space,
- $24 \times 5 = 120$ 1D (240 0D) subspaces for the five-dimensional logical space, and so on.

This is one of the main advantages of dealing with subspaces instead of logical expressions. In fact, whilst the number of monadic subspaces increases by a multiplication factor as the number of dimensions (variables) increases according to the formula $n = (n-1) \times N$, where n parametrizes the subspaces and N parametrizes the variables, the number m of logical expressions grows exponentially as the number of variables increases:

$$m = 2^{(2^N)} = 2^{(2^{N-1})} 2^{(2^{N-1})}, \quad (1)$$

where again N is the number of variables.

In analogy with traditional logical expressions, the following formulae can be helpful:

- $[(1 = 7 \otimes 3 = 9 \otimes 5 = 11) \oplus (1 = 7 \otimes 3 = 9 \otimes 6 = 12)]$
 $= 1 = 7 \otimes 3 = 9 = 1 \otimes 3$
- $[(1 = 7 \otimes 3 = 9 \otimes 5 = 11) \oplus (1 = 7 \otimes 3 = 9 \otimes 6 = 12)]$
 $= 1 = 7 \otimes [(3 = 9 \otimes 5 = 11) \oplus (4 = 10 \otimes 6 = 12)]$
 $= 1 = 7 \otimes [(9 \otimes 11) \oplus (10 \otimes 12)]$
- $[(1 = 7 \otimes 3 = 9) \oplus (1 = 7 \otimes 5 = 11)] = 1 = 7 \otimes (9 \oplus 11).$
- $[(1 = 7 \oplus 3 = 9) \otimes (1 = 7 \oplus 5 = 11)] = 1 = 7 \oplus (9 \otimes 11).$
- $[(1 = 7 \oplus 3 = 9 \oplus 5 = 11) \otimes (1 = 7 \oplus 3 = 9 \oplus 6 = 12)]$
 $= 1 = 7 \oplus [(9 \oplus 11) \otimes (10 \oplus 12)]$
- $[(1 = 7 \oplus 3 = 9 \oplus 5 = 11) \otimes (1 = 7 \oplus 3 = 9 \oplus 6 = 12)]$
 $= 1 \oplus 3$

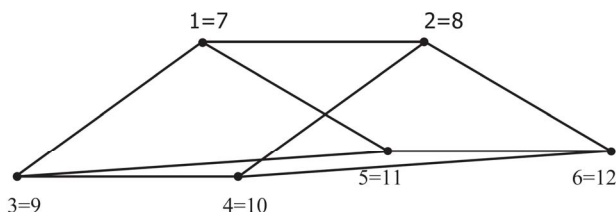


Figure 4. Three-dimensional space. Only the edges of the pentahedron are shown for the sake of representation.

These operations display the fact that the concept of subspace has certain analogies with that of logical expression. However, it is also slightly different. The last passage on the right shows only simplified connections between Y and Z since it involves only their 2D subspace.

Let us consider that the whole 3D space is constituted from a low level and an upper level instantiating formulae representing the combination of all three variables through product and sum, respectively (Levels 1 - 7 and 7 - 1 as displayed in **Tables 3** and **9**, respectively: I do not consider here contradiction and tautology). The pairwise combination of both the first and the second kind of statements will generate binary relations in terms of product and sum (Levels 2 - 6 and 6 - 2 as displayed in **Tables 4** and **8**, respectively), that is a piece of two-dimensional logic. The pairwise combination of the first and second kind of binary relations will generate all cross triplets (Levels 3 - 5 and 5 - 3 as displayed in **Tables 5** and **7**, respectively). Finally, the combination of the latter two kinds of formulae will generate from above and below the same formulae (Level 4 - 4). Most of these expressions involve all three variables but some of them are also monadic expression, another piece of two-dimensional logic and also the essence of one-dimensional one (Level 4-4). We may consider the expressions above for X , as an example.

3. A Vectorial Representation

An interesting possibility is to conceive all logical statements and also subspaces in vectorial terms [4]. We can indeed represent the 8 value assignments in Tab. 1 in terms of the following orthogonal basis:

$$\begin{aligned}
 |a\rangle &= \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, & |b\rangle &= \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, & |c\rangle &= \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, & |d\rangle &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \\
 |e\rangle &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, & |f\rangle &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, & |g\rangle &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, & |h\rangle &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \quad (2)
 \end{aligned}$$

We are now in position to write any statement in the 3D space as a combination of these basis vectors. However, also the 0D components of any subspace can be written in this terms, since we have:

$$\begin{aligned}
 |1\rangle = |7\rangle &= \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, & |3\rangle = |9\rangle &= \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, & |5\rangle = |11\rangle &= \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \\
 |2\rangle = |8\rangle &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}, & |4\rangle = |10\rangle &= \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}, & |6\rangle = |12\rangle &= \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \quad (3)
 \end{aligned}$$

where the columnar sequence of numbers corresponds to the ID of each denoted variable or statement. Note that $|1\rangle$ ($|7\rangle$) and $|2\rangle$ ($|8\rangle$) constitute an orthogonal basis for the 1D logical space and similarly for the other variables. Instead, all vectors on the first row are parallel (as well as those in the second row), what can be seen by the fact that they pairwise share 4 values out of 8 (2 out of the first 4 numbers and the other 2 out of the last 4 numbers of each column vector). This means that the 2D reference frame whose axes are $|3\rangle$ and $|4\rangle$ is displaced of some length relative to reference frame constituted by axes $|1\rangle$ and $|2\rangle$. This means that the line connecting the points individuated by $|1\rangle$ and $|2\rangle$ and the line connecting the points individuated by $|3\rangle$ and $|4\rangle$ are parallel, what allows to recover the plane shown in **Figure 3**, as displayed in **Figure 5**. Similarly, also the reference frame whose axes are $|11\rangle$ and $|12\rangle$ is displaced of same length relative to the reference frame $|9\rangle$ - $|10\rangle$ as well as the reference frame whose axes are $|5\rangle$ and $|6\rangle$ is displaced of same length relative to the reference frame $|7\rangle$ - $|8\rangle$. This allows to fully recover the 3D logical space of **Figure 4**, as displayed in **Figure 6**. Note that at least one of the reference frames (here $|1\rangle$ - $|2\rangle$) needs to be displaced along two directions, one for each of the other two reference frames.

An alternative representation is to make use of 2D subspaces as vectors. For instance an orthogonal basis is constituted by subspaces

$\{2 = 8 \odot 3 = 9, 1 = 7 \odot 6 = 12, 4 = 10 \odot 5 = 11\}$, where \odot stands for either sum or product between subspaces. If we take all these alternative choices together in the 3D space, we get an intersection of two planes as shown in **Figures 7** and **8**. Note that the vectors of the second representation can be considered as superpositions of the first one and vice versa.

This representation allows us to write a classical derivation like the syllogism *Barbara* in this terms:

$$\begin{aligned}
 \text{Tr}_X (|1 = 7\rangle \oplus |4 = 10\rangle \otimes |2 = 8\rangle \oplus |5 = 11\rangle) \\
 = |11\rangle \oplus |10\rangle, \quad (4)
 \end{aligned}$$

where Tr_X symbolizes the mathematical operation of tracing the “system” X out. However, considered that

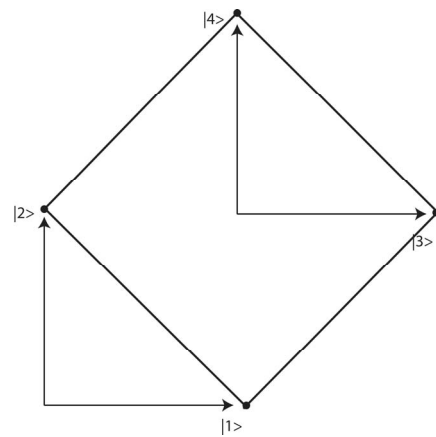


Figure 5. Relations between 2D vectorial spaces and 2D logical subspaces.

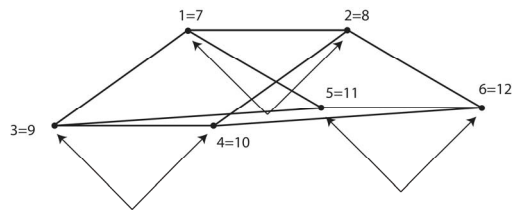


Figure 6. Relations between 3D vectorial spaces and 3D logical space.

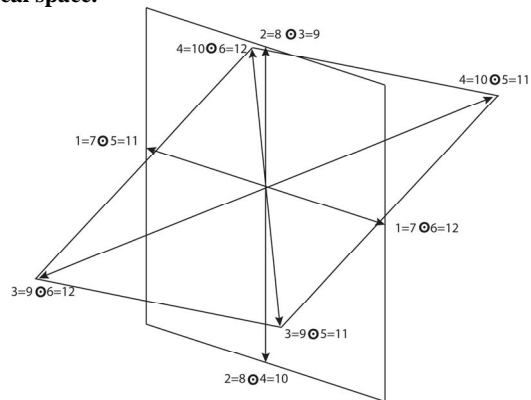


Figure 7. 2D subspaces as vectors: first two planes.

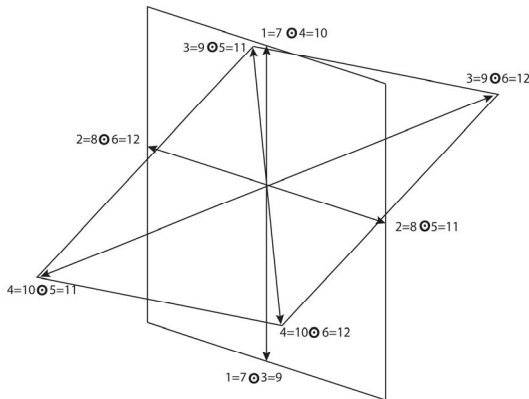


Figure 8. 2D subspaces as vectors: second two planes (the horizontal plane is only rotated relatively to the previous figure).

the nature of the connections (that have the value of operations on logical subspaces) the sign connecting (either \oplus or \otimes) the two states in the final state needs to correspond to logical rules. Such a tracing out corresponds to a kind of information election, what establishes an interesting connection between inferences and information [5]. These rules allow for following definitions:

$$|1 \oplus 4\rangle = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}, |8 \oplus 5\rangle = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}, |11 \oplus 10\rangle = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}. \quad (5)$$

Analogously, we can define similar vectors in the case of subspaces product, for instance:

$$|1 \otimes 4\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, |8 \otimes 5\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, |11 \otimes 10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}. \quad (6)$$

Clearly, all of the above vectors represent the IDs of the relative statements.

4. Quantum Computing: Raising and Lowering Operators

The previous formalism can be easily used for imple-

menting quantum computation [6-8; 9: Ch.17]. For instance, we can represent the three sets X, Y and Z as three harmonic oscillators that can be in ground (0) or excited (1) state each (it suffices to choose an arbitrary state $|n\rangle$ as the threshold between ground and excited states; otherwise systems implementing binary choices can be chosen, for instance spin-particles). What we need then is a family of raising and lowering operators allowing us to climb or descend the ladder of the possible states. We focus on the tridimensional logic (the results can be easily extracted for the one- and bi-dimensional case). Obviously, the rules for product and sum established previously need always to be taken into account. We can build a family of raising operators whose combination can give rise to any of the passages from one level to the next higher one (see **Figure 9**). Any of these operators can be assumed to act on columnar vectors represented by the ID sequence of each statement of the starting level (as shown in the previous section) and produces other columnar vectors of the upper level as output (repetitions are not considered as well as results that are identical to the input).

Mathematically speaking, we cannot act with a raising operator on a vector composed only of zeros (Level 0 - 8). However, this can be easily done by adding a qubit representing the environment and keeping it constant (=1) so that it is irrelevant for the operations inside the logical space. Having said this, in the following I shall no longer deal with this problem.

A family of raising operators from Level 1 - 7 to Level 2 - 6 is shown in **Figure 10**. They are the result of the combination of the previously shown operators. Starting from the top line from the left to right (see **Tables 8 and 9**):

- The first operator allows the generation of Level 2 - 6 Statements 1, 8, 14, 19, 23, 26, and 28. Consider that the number of the propositions perfectly correspond to the number of the operator in the series displayed in **Figure 9** and the same is true for each of the subsequent transformations. This is due to the fact that each of these statements is an eigenvector of this operator and the same is true for the following transformations.
 - The second one the generation of Statements 2, 9, 15, 20, 24, and 27.
 - The third the generation of Statements 3, 10, 16, 21, 25.
 - The fourth (the first on left in the bottom line) the generation of Statements 4, 11, 17, and 22.
 - The fifth the generation of Statements 5, 12, 18.
 - The sixth the generation of Statements 6 and 13.
 - Finally, the last operator generates Statement 7.
- The raising operators allowing the ascension from

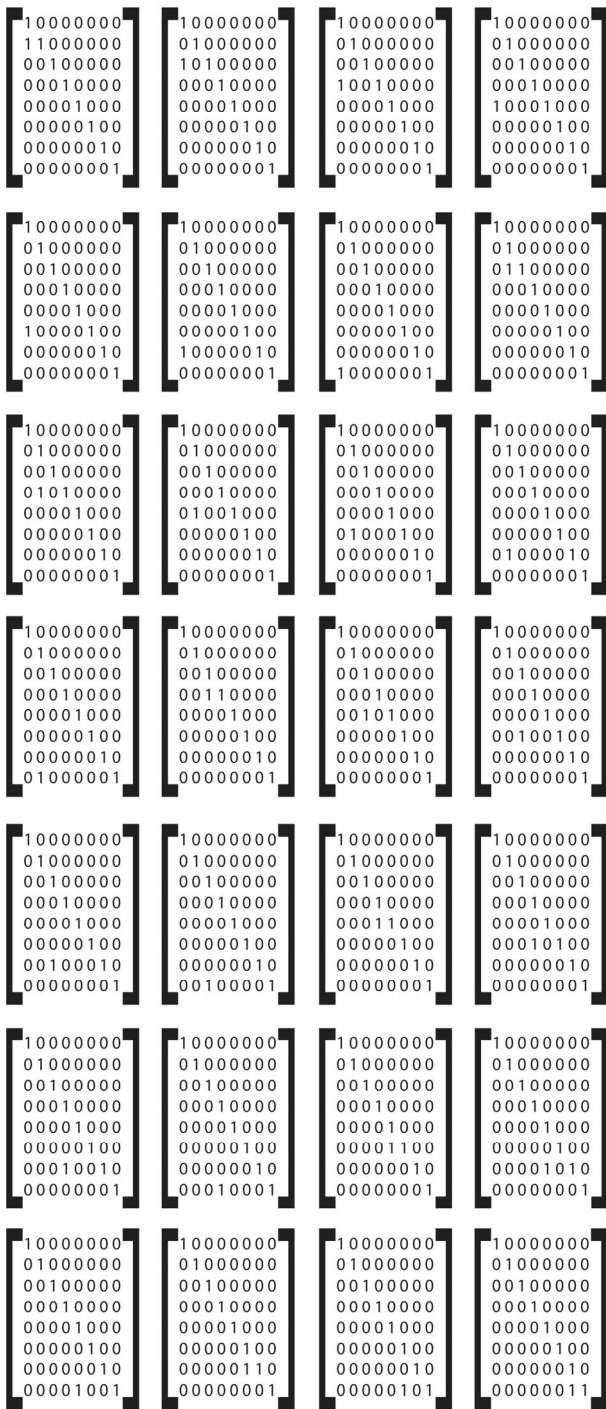


Figure 9. The 28 raising operators allowing the passage from any level to the next higher one.

Level 2 - 6 to Level 3-5 (see Tables 7 and 8) are represented in Figure 9 (always starting from the top line from the left to right):

- Operator 1 generates Level 5-3 Statements 1, 2, 3, 4, 5, 6.
- Operator 2 generates Statements 7, 8, 9, 10, 11.
- Operator 3 generates Statements 12, 13, 14, 15.

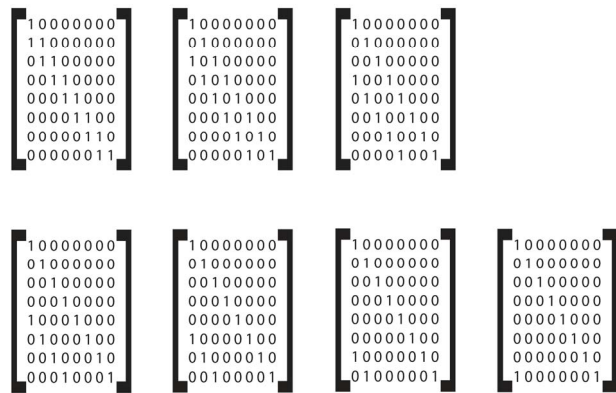


Figure 10. The seven raising operators allowing the passage from Level 1 - 7 to Level 2 - 6.

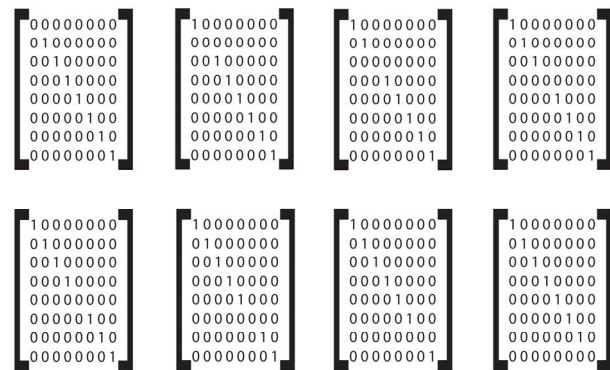


Figure 11. The eight lowering operators allowing the passage from any level to the next lower level.

- Operator 4 generates statements 16, 17, 18.
- Operator 5 generates Statements 19, 20.
- Operator 6 generates Statement 21. Operator 7 is redundant.
- Operator 8 generates Statements 22, 23, 24, 25, 26.
- Operator 9 generates Statements 27, 28, 29, 30.
- Operator 10 generates Statements 31, 32, 33.
- Operator 11 generates Statements 34 and 34.
- Operator 12 generates Statement 36. Operator 13 is redundant.
- Operator 14 generates Statements 37, 38, 39, 40.
- Operator 15 generates Statements 41, 42, 43.
- Operator 16 generates Statements 44 and 45.
- Operator 17 generates Statement 46. Operator 18 is redundant.
- Operator 19 generates Statements 47, 48, 49.
- Operator 20 generates Statements 50 and 51.
- Operator 21 generates Statement 52. Operator 22 is redundant.
- Operator 23 generates Statements 53 and 54.
- Operator 24 generates Statement 55. Operator 25 is redundant.
- Operator 26 generates Statement 56. Operators 27 and 28 are redundant.

Again I have not considered repetition, so that "later" operators are more diminished in their generating capacity than is actually the case. We can reiterate this procedure and generate any subsequent level.

The lowering operators that bring back statements from any given level to the next lower one can be built as in **Figure 11**. It is clear that the first one (always starting from the top line from the left to right) annihilates the term a in any statement, the second the term b , and so on. They are sort of negative projectors: instead of projecting e.g. on the component a they project on $\text{not-}a$.

5. Conclusion

What is interesting with the previous approach is that we can build a quantum computer that generates only logical statements (in fact any of the previous 256 statements and their connections in the 3D space are logical). This means that we can build in this way any kind of logical rule. Moreover, by implementing the procedures of tracing-out, we are able to generate any kind of inference on a quantum computer. In other words, a quantum "processor" would spontaneously generate both logical rules and inferences, which would represent a considerable progress.

REFERENCES

- [1] G. Boole, "An Investigation of the Laws of Thought, on Which Are Founded the Mathematical Theories of Logic and Probabilities," New York, Dover, 1958.
- [2] A. Tarski, "On the Foundations of Boolean Algebra," Vol. 10, 1935, pp. 320-341.
- [3] G. Auletta, "Mechanical Logic in three-Dimensional Space," PanStanford Pub, Peking, 2014.
- [4] G. Auletta and S.-Y. Wang, "Quantum Mechanics for Thinkers," Pan Stanford Pub, Peking, 2014.
- [5] G. Auletta, "Inferences with Information," *Universal Journal of Applied computer Science and Technology*, Vol. 2, No. 2, 2012, pp. 216-221.
- [6] H.-K. Lo, S. Popescu and T. Spiller, "Introduction to Quantum Computation and Information," World Scientific, Singapore, 1998.
- [7] M. A. Nielsen and I. L. Chuang, "Quantum Computation and Quantum Information," University Press, Cambridge, 2011.
- [8] D. Bouwmeester, A. K. Ekert and A. Zeilinger, "The Physics of Quantum Information: Quantum Cryptography, Quantum Teleportation, Quantum Computation," Springer, Berlin, 2000.
- [9] G. Auletta, M. Fortunato and G. Parisi, "Quantum Mechanics," University Press, Cambridge, 2009.
- [10] A. Tarski, Logic, "Semantics, Meta-Mathematics," University Press, Oxford, 1956.