Log. Univers. 14 (2020), 175–207

1661-8297/20/020175-33, *published online* May 15, 2020
https://doi.org/10.1007/s11787-020-00252-3

Logica Universalis

Eigenlogic in the Spirit of George Boole

Zeno Toffano

Abstract. This work presents an operational and geometric approach to logic. It starts from the multilinear elective decomposition of binary logical functions in the original form introduced by George Boole. A justification on historical grounds is presented bridging Boole's theory and the use of his arithmetical logical functions with the axioms of Boolean algebra using sets and quantum logic. It is shown that this algebraic polynomial formulation can be naturally extended to operators in finite vector spaces. Logical operators will appear as commuting projection operators and the truth values, which take the binary values $\{0, 1\}$, are the respective eigenvalues. In this view the solution of a logical proposition resulting from the operation on a combination of arguments will appear as a selection where the outcome can only be one of the eigenvalues. In this way propositional logic can be formalized in linear algebra by using elective developments which correspond here to combinations of tensored elementary projection operators. The original and principal motivation of this work is for applications in the new field of quantum information, differences are outlined with more traditional quantum logic approaches.

Mathematics Subject Classification. Primary 81P10; Secondary 03B05, 05C50, 15A69, 81P68, 94C10.

Keywords. Boole's propositional logic, Truth tables, Switching theory, Orthogonal projection matrix, Quantum logic, Quantum gates.

1. Introduction

The year 2015 celebrated discretely the 200th anniversary of the birth of George Boole (1815–1864). His visionary approach to logic has led to the formalization in simple mathematical language what was prior to him a language and philosophy oriented discipline.

His initial motivation as it appears clearly in his first work on logic in 1847: "*Mathematical Analysis of Logic*" [1] was to propose an algebraic formulation which could generate all the possible logical propositions, to express any logical proposition by an equation, and find the most general consequences of

Birkhäuser

any finite collection of logical propositions by algebraic reasoning applied to the corresponding equations. He then wrote the synthesis of all his investigations in logic in 1854 with the "*The Laws of Thought*" [2].

In 1847 George Boole was already an outstanding mathematician, he was awarded the Gold Medal of the Royal Society in 1844 for his memoir "*On a General Method in Analysis*". He was an expert in the resolution of nonlinear differential equations and introduced many new methods using symbolic algebra as is well outlined by Marie-Jose Durand-Richard in [3] and Maria Panteki in [4].

Evidently George Boole became fond of operators because of his successes in applying the algebra of differential operators in the years 1841—1845. His approach can be viewed as operational, this characteristic is rarely considered nowadays as pointed out by Theodeore Hailperin [5,6].

George Boole (see [1] p. 16) uses X, Y, Z... to represent the individual members of classes. He then introduces the symbol x, which he named *elective symbol*, operating upon any object comprehending individuals or classes by selecting all the X's which it contains. It follows that the product of the elective symbols "xy will represent, in succession, the selection of the class Y, and the selection from the class Y of such objects of the class X that are contained in it, the result being the class common to both X's and Y's". In logical language this is the operation of conjunction, AND.

An expression in which the elective symbols, x, y, z..., are involved becomes an elective function only if it can be considered "interpretable" in logic. George Boole did not give a precise definition of what he meant by an elective function, it seems likely that he meant that any algebraic function in elective symbols x, y, z..., would be an elective function. This is the case when the expression sums up to the two possible values 0 and 1. In logic the numbers 0 and 1 correspond to false and true respectively. So, according to George Boole, all the quantities become interpretable when they take the values 0 and 1.

George Boole's logic using symbolic algebra was different and new because he was convinced that logic had not only to do with "quantity" but should possess a "deeper system of relations" that had to do with the activity of "deductive reasoning". Now with these premises he was able to use all the common operations of ordinary algebra but introducing a special condition on the symbols: the idempotence law $x^2 = x$. This law can only be satisfied by the numbers 0 and 1 and was by him considered as the peculiar law for logic. In his second book on logic [2] he gives to this law the status of "the fundamental law of thought". An interesting outlook of the relation in modern logic of this idempotence law with the law of contradiction has been recently given by Jean-Yves Béziau in [7].

For George Boole all arguments and functions in logic can be considered as elective symbols. For example he stated (p. 63 in [1]) : "It is evident that if the number of elective symbols is n, the number of the moduli will be 2^n, and that their separate values will be obtained by interchanging in every possible way the values 1 and 0 in the places of the elective symbols of the given function." n stands for the number of elective symbols which correspond to the number of arguments of the logical system or in modern language its *arity* (the letter m in the original text is here replaced by the letter n). The *moduli*, for George Boole, are the co-factors in the development (p. 62 in [1]). From what stated above an obvious conclusion is that there are 2^{2^n} possible expansions of elective functions, but curiously George Boole does not draw this conclusion explicitly.

With the introduction of truth tables by Charles Sanders Peirce in the early 1880's [8], attracting little attention at the time, as stated by Karl Menger in [9] and successively, around 1920, rediscovered simultaneously and independently by Emil Post [10] and by Ludwig Wittgenstein (prop. 5.101 in the *Tractatus* [11]) the counting of the number of possible elementary logical propositions (connectives) became an evidence.

By the way Emil Post in [10] extended the truth table method to alphabets greater than binary ($m > 2$) leading to the combinatorial number m^{m^n} of elementary multivalued logical connectives with m values and n arguments.

The aspect of Boole's method which has been much discussed was his interpretation given to the two special numbers: 1 and 0. The number 1 represented for him the class of all conceivable objects *i.e.* the entire universe, and naturally the number 0 should have represented the empty class. But it is not clear in [1,2] if George Boole does ever refer to 0 as being a class, or was it just part of his algebraic machinery? As for the objection to the use of 1, it has been to the requirement that it refer to the entire universe as opposed to a *universe of discourse* (extent of the field within which all the objects of our discourse are found) [12].

George Boole introduces considerable vagueness in [1] as to when one is working in a logic of classes, and when in a logic of propositions. In his propositional calculus he restricted his attention to statements that were always true or always false, this reduces hypothetical propositions to categorical propositions. In 1854 [2] George Boole more explicitly replaces the algebra of selection operators by the algebra of classes.

As was outlined by Theodore Hailperin in [5] the elective symbols and functions denote operators and it will be emphasized in this work that the algebra of elective symbols can also be interpreted as an algebra of commuting projection operators and used for developing propositional logic in a linear algebra framework by the isomorphism of Boole's elective symbols and functions with commuting projection operators. This new method is named *Eigenlogic* [13–15].

In this paper hypothetical propositions will not be considered, the analysis will be restricted to what is currently named *propositional logic* (also named *sentential logic*) and will not deal with *predicate logic* (also named *first-order logic*) which uses *quantifiers* (the *existence quantifier* $\exists$ and the *universal quantifier* $\forall$) on propositions. Possibilities of extending Eigenlogic to first order logic have been presented at UNILOG 2018 [14] and discussed in [15].

2. Elective Symbols and Functions

2.1. Idempotence and Boole's Development Theorem

Elective symbols obey the following laws, these are sufficient to build an algebra.

Law (2.1) says that elective symbols are distributive. This means, according to Boole, that "the result of an act of election is independent of the grouping or classification of the subject".

$$x(u+v) = xu + xv \tag{2.1}$$

Law (2.2) says that elective symbols commute, this because: "it is indifferent in what order two successive acts of election are performed".

$$xy = yx \tag{2.2}$$

Law (2.3) called *index law* by George Boole represents the idempotence of an elective symbol, he states: "that the result of a given act of election performed twice or any number of times in succession is the result of the same act performed once".

$$x^n = x \tag{2.3}$$

As a consequence of this law George Boole formulated the two following equivalent equations.

$$\begin{aligned} x^2 &= x \\ x(1-x) &= 0 \end{aligned} \tag{2.4}$$

Equation (2.4) explicitly shows that the numbers 0 and 1 are the only possible ones. It also states the orthogonality between the elective symbol x and $(1-x)$, which represents the complement or negation of x. Also:

$$x + (1-x) = 1 \tag{2.5}$$

this equation shows that the symbol x and its complement $(1-x)$ form the universe class.

Now with these laws and symbols elective functions can be calculated. It is interesting to illustrate how George Boole came to a general expression of an elective function using the Mac Laurin development of the function $f(x)$ around the number 0 (see [1] p. 60). Because of the index law (2.3) or the idempotence law (2.4) the symbol x becomes a factor of the series starting from the second term in the Mac Laurin development, this gives:

$$f(x) = f(0) + x\left[f'(0) + \frac{1}{2!}f''(0) + \frac{1}{3!}f'''(0) + \cdots\right] \tag{2.6}$$

Then by calculating the function at the value 1, $f(x=1)$, using Eq. (2.6), one finds a substitute expression of the series. By substituting this expression back in Eq. (2.6) one finally gets:

$$\begin{aligned} f(x) &= f(0) + x(f(1) - f(0)) \\ &= f(0)(1-x) + f(1)x \end{aligned} \tag{2.7}$$

In a simpler way these expressions can be obtained directly by classical interpolation methods using for example Lagrange interpolation polynomials for a finite number m of distinct points x_i. The Lagrange polynomials being then of degree $m-1$ and are given by:

$$\pi_{x_i}(x) = \prod_{j\ (j \neq i)}^{m} \frac{(x - x_j)}{(x_i - x_j)} \tag{2.8}$$

The interpolation function $f(x)$ of a given function $g(x)$ is then expressed using the finite polynomial development over the chosen m distinct points x_i:

$$f(x) = \sum_{i=1}^{m} g(x_i)\, \pi_{x_i}(x) \tag{2.9}$$

For a binary system ($m = 2$) with alphabet values $\{0, 1\}$ the two interpolation polynomials are easily calculated from (2.8), giving respectively: $\pi_{x_0=0}(x) = (1 - x)$ and $\pi_{x_1=1}(x) = x$, which are the same as in (2.7), and for this alphabet, Eq. (2.9) is equivalent to Eq. (2.7) using the interpolation points $g(0) = f(0)$ and $g(1) = f(1)$. In this way the demonstration of the elective development theorem does not necessitate infinite polynomial power series, *e.g.* the Maclaurin expansion, as was done with the power series proof by George Boole in [1].

It must be underlined that Lagrange polynomials (2.8) are by construction idempotent functions at the interpolation points, more precisely: $\pi_{x_i}(x = x_i) = 1$ and $\pi_{x_i}(x = x_j \neq x_i) = 0$.

The same method can be extended to other binary alphabets, *e.g.* the often employed polar alphabet $u \in \{+1, -1\}$ (which can also be obtained directly using the *Householder transform* $u = 1 - 2x$). Generally the interpolation method applies for all multi-valued alphabets (for developments see [13,15]).

Equation (2.7) shows that an elective function can be uniquely developed using the two orthogonal elective symbols x and $(1-x)$. Now if the function is to be "interpretable" in logic (George Boole's formulation) it should only take the values 0 and 1, and this means that both co-factors $f(0)$ and $f(1)$ (*moduli* for George Boole) take also the values 0 or 1. These coefficients represent the *truth values* for the logical function.

How many possibilities, or stated in logical language, how many different logical functions can one build using n arguments? As already discussed before the possible combinations are 2^{2^n}. So considering a unique symbol (*arity* $n = 1$), one obtains 4 distinct elective functions. These are shown on Table 1.

A similar procedure can be used (see p. 62 in [1]) for elective functions of two arguments $f(x, y)$, this gives the following bilinear development using 4 orthogonal and idempotent polynomials:

$$\begin{aligned} f(x, y) = {} & f(0, 0)(1 - x)(1 - y) + f(0, 1)(1 - x)y \\ & + f(1, 0)x(1 - y) + f(1, 1)xy \end{aligned} \tag{2.10}$$

Table 1. The four single argument logical elective functions

Function $f_i^{[1]}$	Operator	Truth table	Canonical form	Arithmetic form
$f_0^{[1]}$	F	0 0	0	0
$f_1^{[1]}$	$\bar{A}$	1 0	$(1-x)$	$1-x$
$f_2^{[1]}$	A	0 1	x	x
$f_3^{[1]}$	T	1 1	$(1-x)+x$	1

And so on for increasing n. For $n=2$ one has $2^{2^{n=2}}=16$ different elective functions (given in Table 2) and for $n=3$, $2^{2^{n=3}}=256$. All elective functions are idempotent: $f^2=f$. Here also finite interpolation methods could be used this time using multivariate functions.

Equation (2.10) represents the canonical elective development of a two argument elective function and has the same structure as the *minterm* disjunction canonical form in Boolean algebra [5] which represents the disjunction of mutually exclusive conjunctions (see hereafter).

So this discussion shows that all logical functions can be expressed as a combination of degree 1 multilinear polynomials and it can be shown that this decomposition is unique.

George Boole has also developed a method of resolution of what he called *elective equations* where for example the question is: for what values an elective function is true? (see [1] p. 70).

A very simple method used for resolving elective equations uses the orthogonality of the different elective polynomials which are multiplied by the respective co-factors (*moduli*) $f^{[n]}(a,b,...)$ in the development, these polynomials are named $\pi^{[n]}_{(a,b,...)}$ for a given combination of fixed values $(a,b,...)$. This gives the following equation for selecting the individual co-factors for an n symbol elective function:

$$f^{[n]}(x,y,...)\cdot\pi^{[n]}_{(a,b,...)}=f^{[n]}(a,b,...)\,\pi^{[n]}_{(a,b,...)} \tag{2.11}$$

Equation (2.11) can be used whatever the number of symbols and also when the functions are not explicitly put in the canonical form. For example if one wants to select the coefficient $f(0,1)$ out of $f(x,y)$ in Eq. (2.10), one simply multiplies the function by the corresponding orthogonal polynomial $(1-x)y$. Without doubt it is most of the times easier to evaluate directly $f(0,1)$.

3. Elective Symbolic Logic

3.1. Truth Tables and Elective Functions

In this section the link of elective functions with ordinary propositional logic is presented. Functions and symbols will take exclusively the two binary values 0

TABLE 2. The sixteen two argument logical elective functions

Funct. $f_i^{[2]}$	Connective for A and B	Truth table	Canonical form	Arithmetic form
$f_0^{[2]}$	F	0 0 0 0	0	0
$f_1^{[2]}$	NOR, $\bar{A} \wedge \bar{B}$	1 0 0 0	$(1-x)(1-y)$	$1-x-y+xy$
$f_2^{[2]}$	$\overline{A \Leftarrow B}$	0 1 0 0	$(1-x)y$	$y-xy$
$f_3^{[2]}$	$\overline{A}$	1 1 0 0	$(1-x)(1-y)+(1-x)y$	$1-x$
$f_4^{[2]}$	$\overline{A \Rightarrow B}$	0 0 1 0	$x(1-y)$	$x-xy$
$f_5^{[2]}$	$\overline{B}$	1 0 1 0	$(1-x)(1-y)+xy$	$1-y$
$f_6^{[2]}$	XOR, $A \oplus B$	0 1 1 0	$(1-x)y+x(1-y)$	$x+y-2xy$
$f_7^{[2]}$	$NAND$, $\bar{A} \vee \bar{B}$	1 1 1 0	$(1-x)(1-y)+(1-x)y+x(1-y)$	$1-xy$
$f_8^{[2]}$	AND, $A \wedge B$	0 0 0 1	xy	xy
$f_9^{[2]}$	$A \Leftrightarrow B$	1 0 0 1	$(1-x)(1-y)+xy$	$1-x-y-2xy$
$f_{10}^{[2]}$	B	0 1 0 1	$(1-x)y+xy$	y
$f_{11}^{[2]}$	$A \Rightarrow B$	1 1 0 1	$(1-x)(1-y)+(1-x)y$	$1-x+xy$
$f_{12}^{[2]}$	A	0 0 1 1	$x(1-y)+xy+xy$	x
$f_{13}^{[2]}$	$A \Leftarrow B$	1 0 1 1	$(1-x)(1-y)+x(1-y)+xy$	$1-y+xy$
$f_{14}^{[2]}$	OR, $A \vee B$	0 1 1 1	$(1-x)y+x(1-y)+xy$	$x+y+xy$
$f_{15}^{[2]}$	T	1 1 1 1	$(1-x)(1-y)+(1-x)y+x(1-y)+xy$	1

and 1 representing respectively the false (F) and true (T) character of a given proposition. Logical functions are classified according to their truth tables.

Starting from the very simple propositions derived from the single elective symbol x, according to the function development in Eq. (2.7), one sees that there are 4 possible functions depending on the values taken by $f(0)$ and $f(1)$ respectively. This is shown on Table 1.

In this case the two non trivial propositions are the *logical projection* A and its negation $\bar{A}$. The other two give constant outcomes: false F and true T whatever the value of the argument.

On Table 2 are shown the 16 elective functions, $f_i^{[2]}$, for $n = 2$ arguments. The corresponding elective polynomials can be straightforwardly obtained by substituting the respective truth values in front of the four polynomial terms in Eq. (2.10). According to the standard classification, given for example by Donald Knuth in [16], logical functions are ordered with increasing binary number in the truth table (counting order goes from left to right: the lower digit is on the left). The representation used here corresponds to what is often called the *truth vector* of the function: $(f(0,0), f(0,1), f(1,0), f(1,1))$.

$f_0^{[2]}$ has the truth values $(0,0,0,0)$ and represents contradiction, $f_1^{[2]}$ is NOR with truth values $(1,0,0,0)$ and so on... For example conjunction (AND, $\wedge$) is $f_8^{[2]}$ with $(0,0,0,1)$, disjunction (OR, $\vee$) is $f_{14}^{[2]}$ with $(0,1,1,1)$ and exclusive disjunction (XOR, $\oplus$) is $f_6^{[2]}$ with $(0,1,1,0)$.

In Table 2 are also shown the canonical polynomial forms issued directly from Eq. (2.10) and the respective simplified arithmetic expressions.

Some precisions on other logical connectives: the expression $A \Rightarrow B$ signifies "A implies B", and the converse $A \Leftarrow B$ signifies "B implies A". The expression for $NAND$ which is "not AND" is given according to the De Morgan's law [16] by $\bar{A} \vee \bar{B}$. The same for NOR, "not OR", given by $\bar{A} \wedge \bar{B}$.

Negation complements the function, this operation is obtained by subtracting from the number 1.

The conjunction (AND, $\wedge$) corresponds to the following elective function:

$$f_8^{[2]}(x,y) = f_{AND}^{[2]}(x,y) = xy \tag{3.1}$$

and its negation $NAND$ is simply:

$$f_7^{[2]}(x,y) = 1 - xy = 1 - f_{AND}^{[2]}(x,y) = f_{NAND}^{[2]}(x,y) \tag{3.2}$$

By complementing the input symbols *i.e.* by replacing the symbols x and y by $1-x$ and $1-y$ respectively one gets other logical functions. For example considering:

$$\begin{aligned} f_1^{[2]}(x,y) &= (1-x)(1-y) = 1 - x - y - xy = 1 - (x + y - xy) \\ &= 1 - f_{14}^{[2]}(x,y) = 1 - f_{OR}^{[2]}(x,y) = f_{NOR}^{[2]}(x,y) \end{aligned} \tag{3.3}$$

this is the complement of the disjunction OR named NOR. This result corresponds to De Morgan's law [16] that states that the conjunction AND of the complements is the complement of the disjunction OR.

$$f_{14}^{[2]}(x,y) = f_{OR}^{[2]}(x,y) = x + y - xy \tag{3.4}$$

remark that the expression for the disjunction OR is given by a polynomial expression containing a minus sign, this is specific to elective functions, and it must be this way in order that the functions be "interpretable".

The expression for the exclusive disjunction XOR is given by:

$$f_6^{[2]}(x,y) = f_{XOR}^{[2]}(x,y) = x + y - 2xy \tag{3.5}$$

this form differs from what is usually used in logic where the last term is omitted due to the fact that the addition operation is considered a modulo 1 sum in Boolean algebra. This function represents the parity function giving 1 when the total number of 1's in the arguments is odd.

The function for implication (named in propositional logic *material implication*) can also be obtained by the same method, the function corresponding to $A \Rightarrow B$ will be $f_{\Rightarrow}^{[2]}$ and the converse $f_{\Leftarrow}^{[2]}$. According to Table 2:

$$\begin{aligned} f_{\Rightarrow}^{[2]}(x,y) &= f_{11}^{[2]}(x,y) = 1 - x + xy \\ f_{\Leftarrow}^{[2]}(x,y) &= f_{13}^{[2]}(x,y) = 1 - y + xy \end{aligned} \tag{3.6}$$

Using De Morgan's theorem, by complementing the arguments, it is easy to verify that $f_{\Rightarrow}^{[2]}$ transforms into $f_{\Leftarrow}^{[2]}$.

The non-implication cases are given by:

$$\begin{aligned} \bar{f}_{\Rightarrow}^{[2]}(x,y) &= f_4^{[2]}(x,y) = x - xy = 1 - f_{\Rightarrow}^{[2]} \\ \bar{f}_{\Leftarrow}^{[2]}(x,y) &= f_2^{[2]}(x,y) = y - xy = 1 - f_{\Leftarrow}^{[2]} \end{aligned} \tag{3.7}$$

One can of course go on by increasing the number of arguments n in a straightforward way. Let's consider $n = 3$, the conjunction becomes simply:

$$f_{AND}^{[3]}(x,y,z) = xyz \tag{3.8}$$

The expression for disjunction is obtained in the same way as in Eq. (2.10) but with three elective symbols x,y and z. Doing straightforward calculation and using the 8 truth values $(0,1,1,1,1,1,1,1)$ gives:

$$f_{OR}^{[3]}(x,y,z) = x + y + z - xy - xz - yz + xyz \tag{3.9}$$

which represents the well-known *inclusion-exclusion rule*, and can be scaled-up to any n.

For the XOR function with $n = 3$ one gets, using the truth values $(0,1,1,0,1,0,0,1)$:

$$f_{XOR}^{[3]}(x,y,z) = x + y + z - 2xy - 2xz - 2yz + 4xyz \tag{3.10}$$

this last expression represents a specific inclusion-exclusion rule which can be also scaled-up straightforwardly to any n.

Another very popular function for $n = 3$ arguments is the majority MAJ which gives the value 1 when there is a majority of 1's for the arguments. The function is obtained using the truth values $(0,0,0,1,0,1,1,1)$:

$$f_{MAJ}^{[3]}(x,y,z) = xy + xz + yz - 2xyz \tag{3.11}$$

These last two logical connectives are currently used together in digital electronics to build a binary full-adder using logical gates, the three input XOR gives the binary sum and the three input MAJ gives the carry out.

So in conclusion this method is completely general and can be straightforwardly applied to all logical connectives whatever the number of arguments.

3.2. Logical Developments

An idempotent elective function $f(x, y, \ldots)$ can be evaluated at the values 0 and 1 by using ordinary numerical algebra, and all the usual propositional functions have truth tables that can be expressed in either the canonical form or in the arithmetic form as is shown in the two last columns of Tables 1 and 2. For example exclusive disjunction (XOR, $\oplus$), expressed in the canonical form $x(1-y) + (1-x)y$ as well as in the arithmetic form $x + y - 2xy$.

The canonical form corresponds in modern digital logic to the canonical *minterm* decomposition. Minterms correspond here to products of elective polynomials. For example for $n = 2$ arguments the minterms are the 4 orthogonal polynomials given in Eq. (2.10), in logical language each minterm is one of the possible 4 conjunctions obtained by complementing none, one or two arguments.

So one can always put whatever logical function in the SOP (Sum Of Products) canonical form, also named the *full conjunctive normal form* [16] which is a sum of minterms. Specifically a minterm is formed by all input arguments, in a given combination complemented or not, connected by "Product" corresponding to conjunction (AND, $\wedge$) and "Sum" corresponding to disjunction (OR, $\vee$) or equivalently exclusive disjunction (XOR, $\oplus$) (see discussion hereafter). Another canonical decomposition is POS (Product Of Sums) of *maxterms*. A maxterm being formed of all input arguments connected by disjunction (OR, $\vee$), "Sum", in a given combination complemented or not, and 'conjunction (AND, $\wedge$), "Product". This form is also named the *disjunctive normal form*.

A SOP with four input arguments can be considered for the following working example:

$$\begin{aligned} F^{[4]}_{\Sigma m(5,7,10,15)}(A, B, C, D) &= (\overline{A} \wedge B \wedge \overline{C} \wedge D) \vee (\overline{A} \wedge B \wedge C \wedge D) \\ &\vee (A \wedge B \wedge \overline{C} \wedge D) \vee (A \wedge B \wedge C \wedge D) \end{aligned} \tag{3.12}$$

The expression $\Sigma m(5, 7, 10, 15)$ is the standard minterm notation, where the numbers correspond to the specific minterms used in the development. In this form one can easily verify that only one among all minterms can be true at a time, this means that each disjunction $\vee$ is actually an exclusive disjunction $\oplus$. In the minterm SOP decomposition, because all the terms are orthogonal, disjunction and exclusive disjunction play the same role.

One can write the expression given in Eq. (3.12) using the formalism presented above by writing directly the elective decomposition:

$$\begin{aligned} &f^{[4]}_{\Sigma m(5,7,10,15)}(x, y, z, r) \\ &\quad = (1-x)y(1-z)r + (1-x)yzr + xy(1-z)r + xyzr = yr \end{aligned} \tag{3.13}$$

so one can transform an expression into other polynomial forms in order to get a simpler expression.

Significant simplifications are obtained when one can factor an argument and its complement for the same expression, for example x and $(1 - x)$. The simplest case being the logical projectors such as A in Table 2 where the canonical form $x(1 - y) + xy$ reduces to x. This last argument is essentially what is used to operate reduction of logical functions by using Karnaugh maps [16].

3.3. Discussion on Elective Arithmetic Logic

Characteristic of George Boole's method is that while some terms appearing in logical expressions may be uninterpretable, equations always are when suitably interpreted, by the rules $(+, -, \times, 0, 1)$, leading *in fine* always to the two values 0 and 1. He also recognizes terms that cannot always be interpreted, such as the term $2xy$, which arise in equation manipulations as for the elective function corresponding to XOR in (3.5). The coherence of the whole enterprise is justified in what Stanley Burris has later called the rule of 0's and 1's [17], which justifies the claim that uninterpretable terms cannot be the ultimate result of equation manipulations from meaningful starting formulas. George Boole provided no proof of this rule, but the consistency of his system was later proved by Theodore Hailperin [5,17], who provided an interpretation based on a fairly simple construction of rings from the integers to provide an interpretation of Boole's theory (see hereafter).

Even though this procedure is simple and straightforward it is not in the habits of logic to use these arithmetic expressions, and the reason why is not so clear. One explanation could be because of technology driven habits: the development of computers using logical gates as building blocks, and binary-digits (bits) as information units has generalized what is called "Boolean algebra" formulated in its actual form by Edward Huntington in 1904 [18], which is not Boole's elective algebra [6]. For example addition is considered in Boolean algebra as a modulo 1 sum giving: $x + x = x$. For a Boolean ring one has even a different rule: $x + x = 0$ (modulo 2 sum). Whereas the elective calculation employs normal arithmetic addition and subtraction as described previously.

Arithmetic expressions are closely related to polynomial expressions over the Galois field $GF_2 = \mathbb{Z}/\mathbb{Z}_2$, but with variables and function values interpreted as the integers 0 and 1 instead of logic values. In this way, arithmetic expressions can be considered as integer counterparts of polynomial expressions over GF_2. For two Boolean variables x_1 and x_2 (using here more standard notation corresponding to two bits) the necessary relations are:

$$\begin{aligned} \bar{x} &= 1 - x & x_1 \wedge x_2 &= x_1 x_2 \\ x_1 \vee x_2 &= x_1 + x_2 - x_1 x_2 & x_1 \oplus x_2 &= x_1 + x_2 - 2x_1 x_2 \end{aligned} \tag{3.14}$$

this resumes all the discussion of the preceding section, the right part of the equations is called the *arithmetic expression*.

It seems that, historically, only John Venn explicitly used the original reasoning of George Boole in order to build his logical graphic diagrams [19].

He used surfaces on a 2 dimensional space which represented the different logical propositions and more precisely intersection and union corresponding to conjunction and disjunction. Doing this he had, in some cases, to subtract portions of surfaces in order to get the correct surface measure. For example considering two overlapping surfaces, the surface representing disjunction (OR, $\vee$) is obtained by the sum of the two surfaces minus their intersecting surface (without this subtraction one would count twice the intersecting surface), also for exclusive disjunction (XOR, $\oplus$) one has to subtract twice the intersecting surface, this leads to formulae of the *inclusion-exclusion* type as illustrated in equations (3.9) and (3.10). The canonical forms of idempotent elective functions in Boole's algebra are the same as for functions in Boolean algebra, and the number of these were well-known in the second half of the 1800s, and fully written out for three variables in 1881 by John Venn [19] (according to Ernst Schröder in [20]).

In 1933 Hassler Whitney [21], showed how to convert the modern algebra of classes (using union, intersection and complement) into numerical algebra, giving three different normal forms (polynomials in x's, polynomials in $(1-x)$'s, and Boole's form) for functions. He failed to recognize that he was converting the modern algebra of classes into Boole's algebra of classes. Theodore Hailperin would realize this decades later.

Nowadays these arithmetic developments are still used for describing switching functions and decision logic design. A good review is given in the book "*Introduction to Logic Design*" by Svetlana Yanushkevich [22]. Arithmetic representations of Boolean functions (*i.e.* here elective functions) are known as *word-level forms*, and are a way to describe the parallel calculation of several Boolean functions at once. Another useful property of these arithmetic representations is used for linearization techniques.

The observation that one can express propositional functions, viewed as switching functions, using polynomials in ordinary numerical algebra, as George Boole did, was used by Howard Aiken in 1951 in [23], where one finds tables for minimal ordinary numerical algebraic expressions for switching functions $f : \{0,1\}^n \rightarrow \{0,1\}$ up to $n = 4$. It is interesting to note that Howard Aiken, who founded the "Harvard Computing Laboratory", the first laboratory devoted to Computer Science at Havard starting in 1937, developed the first computer, the ASCC (Automatic Sequence Controlled Calculator), also called Harvard MARK 1 in 1944 with IBM. He first found that arithmetic expressions can be useful in designing logic circuits and used them in the successive computers Harvard MARK 3 and MARK 4.

This kind of logic did not breakthrough principally because the family of Harvard MARK computers where replaced by the ENIAC computer generation which used semiconductor transistors instead of electromechanical switches and vacuum tubes and relied on the *bit* and *logical gate* paradigm introduced originally by Claude E. Shannon in his 1938 paper "*A Symbolic Analysis of Relay and Switching Circuits*" [24] where he "tailored" Boolean logic to switching circuits. This fact about Claude E. Shannon is not well known in the scientific community, he deserves mostly his popularity for the mathematical definition of information in his famous 1948 paper "*A mathematical theory of communication*" [25]. But someway 10 years earlier he laid the foundations of digital electronics with the definition of the "ordinary" logical gates which are the basic digital electronic building blocks of all computers. An interesting description of these historical facts is in the book "*The Logician and the Engineer: How George Boole and Claude Shannon Created the Information Age*" by Paul J. Nahin [26].

4. Elective Projector Logic

The following section presents the real new part of this work based on linear algebra. It will be shown that the results given above can be applied within the framework of the following matrix operator formalism. It must be emphasized that at the time of George Boole methods in matrix linear algebra were in their nascent form. Most methods have been introduced around 1850, major contributions are due to Arthur Cayley and James Joseph Sylvester, the latter having introduced the term *matrix*. The modern definition of a vector space was subsequently introduced by Giuseppe Peano in 1888.

4.1. Parallels to Boole's Elective Expansion with Idempotent Projection Operators in Linear Algebra

One question arises: why one would want to find parallels to Boole's expansion theorem for idempotent functions of idempotent symbols in linear algebra? One of the principal motivations of this work is seeking the links with operational algebra as is used in quantum mechanics in Hilbert space with applications in the emerging field of *quantum information* and *quantum computation* [13, 15, 27].

Concerning the possible applications of the idempotent linear operator algebra version of Boole's operator algebra to quantum mechanics, some important things can be recalled. Quantum mechanics was a hot topic at Harvard starting in the late 1920s. Marshall H. Stone, a student of Garret D. Birkhoff, wrote a book in the early 1930's on linear operators on infinite dimensional spaces [28] he then subsequently, starting in 1934, undertook a great research effort in logic culminating in two papers on Boolean algebras, Boolean rings, and Boolean spaces [29, 30].

Marshall H. Stone showed that any Boolean algebra is isomorphic to a field of sets, and he motivated his algebraic approach to logic by the fact that it allows to connect many different areas of mathematics. As underlined by Stanley Burris [17] it is interesting to note that his motivation for studying Boolean algebra came from the mathematics of areas like quantum mechanics: (quote from his 1936 paper [29]) "The writer's interest in the subject, for example, arose in connection with the spectral theory of symmetric transformations in Hilbert space and certain related properties of abstract integrals." This could have meant that he was looking at Boolean algebras of idempotent linear transformations, and realized that there were a lot of examples of Boolean algebras that had not been considered before. He goes on to prove that

Huntington's axioms of Boolean algebras [18] are equivalent with the axioms of commutative rings with unit element, in which every element is idempotent, called Boolean rings ([29] p. 38).

According to Dirk Schlimm in [31] Marshall H. Stone was able to connect the theory of Boolean rings also to topology by proving that "the theory of Boolean rings is mathematically equivalent to the theory of locally-bicompact totally-disconnected topological spaces". This identification, also referred to as the *fundamental representation theorem* allows for the transfer of topological methods to the study of Boolean algebras, and vice-versa, is known as the *Stone duality*.

An important work on developing a specific logic for quantum mechanics has been undertaken by Garrett Birkhoff and John von Neumann in their 1936 seminal paper on the subject [32], they proposed the replacement of Boolean algebras with the lattice of closed subspaces of a (finite) Hilbert space. Quantum logic has become an independent discipline with many promoters and different versions, even though it has not still reached the status of an "operational tool" in the emerging quantum information and quantum computing fields.

But already in 1932 John von Neumann made parallels between projections in Hilbert space and logical propositions (Ch. III: *The Quantum Statistics*, Sect. 5: *Projections as Propositions*, p. 247 in [33]). As is clearly stated by François David in [34] John von Neumann noticed that the *observables* (name given to Hermitian operators in quantum mechanics) given by projection operators $\mathbf{P}$, such that $\mathbf{P}^2 = \mathbf{P} = \mathbf{P}^\dagger$, correspond to propositions with a *Yes* or *No* (*i.e. True* or *False*) outcome in a logical system.

An orthogonal projection operator $\mathbf{P}$ onto a linear subspace P, in Hilbert space is indeed an observable that can take only the eigenvalues 1 (if the corresponding quantum state belongs the subspace P) or 0 (if the corresponding quantum state belongs to the orthogonal subspace to P). Thus the two values 1 and 0 are the only possible eigenvalues of the projection operator $\mathbf{P}$, and this statement, that a measurement can only give one of the eigenvalues, is part of the fundamental measurement postulate in Quantum Mechanics [27,33,34]. Thus measuring the observable $\mathbf{P}$ is equivalent to perform a test on the system, or to check the validity of a logical proposition on the system, which can only be true or false, and not some combination of these values. This states in other terms the Aristotelian *law of the excluded middle* for a proposition.

In his 1932 book [33] John von Neumann cites the book of Marshall H. Stone (p. 70: *Projections* in [28]) about the operations conserving the properties of projection operators and gives the following rules:

$$\mathbf{P}_1 \cdot \mathbf{P}_2 \quad \text{is a projection } \textit{iff} \quad \mathbf{P}_1 \cdot \mathbf{P}_2 \equiv \mathbf{P}_2 \cdot \mathbf{P}_1 \quad \text{(they commute)}$$
$$\mathbf{P}_1 + \mathbf{P}_2 \quad \text{is a projection } \textit{iff} \quad \mathbf{P}_1 \cdot \mathbf{P}_2 \equiv 0 \quad \text{or} \quad \mathbf{P}_2 \cdot \mathbf{P}_1 \equiv 0$$
$$\mathbf{P}_1 - \mathbf{P}_2 \quad \text{is a projection } \textit{iff} \quad \mathbf{P}_1 \cdot \mathbf{P}_2 \equiv \mathbf{P}_2 \quad \text{or} \quad \mathbf{P}_2 \cdot \mathbf{P}_1 \equiv \mathbf{P}_2$$

This shows that the property of projection operators, *i.e.* idempotence, is conserved under the operations of (matrix) product $\mathbf{P}_1 \cdot \mathbf{P}_2$, sum $\mathbf{P}_1 + \mathbf{P}_2$ and difference $\mathbf{P}_1 - \mathbf{P}_2$ only for commuting projection operators, this condition is usually expressed in quantum mechanics by the commutation relation $\mathbf{P}_1 \cdot$

$\mathbf{P}_2 - \mathbf{P}_2 \cdot \mathbf{P}_1 = [\mathbf{P}_1, \mathbf{P}_2] = 0$. The sum is only defined for disjoint subspaces, $P_1 \cap P_2 \equiv 0$, and the difference with the inclusion of subspaces $P_2 \subseteq P_1$. These properties will be at the basis of the development given hereafter for *Eigenlogic*, establishing the connection between eigenvalues and logic because of the fact that idempotent diagonal matrices have only 1's and 0's on the diagonal, and hence these are the only possible outcomes (eigenvalues).

Also it is interesting to note that the very definition of a pure quantum state when expressed by a density matrix, also introduced by John von Neumann, is a *ray*: a rank-1 idempotent projection operator spanning a one-dimensional subspace. All these concepts lay at the foundations of quantum theory.

The work presented here can be understood in this framework, even though one does not need here (at least at this stage) the non-commutative algebra which is at the basis of the peculiar aspects of quantum theory, having as consequence, for example, the non-distributivity of quantum logic. The approach here can be viewed as *classical* in the sense that the discussion is restricted to families of commuting observables which are here projection operators. But because this approach uses observables it can also be considered as being part of the global "quantum machinery". Most problems in traditional quantum physics deal with finding eigenfunctions and eigenvalues of some physical observable, the most investigated being the Hamiltonian observables whose eigenvalues represent the energies of a physical system and whose eigenstates are the stationary states representing the stable equilibrium solutions, in the form of wavefunctions, of the Schrödinger equation. The non-traditional aspects of quantum mechanics, principally superposition, entanglement and non-commutativity, are largely employed in the field of quantum information and are considered as a resource for quantum computing [27]. Nothing in the formulation presented here forbids to explore outside of the family of commuting logical projection operators, or to consider vectors that are not eigenvectors of the same logical family. This is the object of ongoing research (see [13–15,35]).

4.2. Link of George Boole's Formulation with Linear Algebra

If one goes back to the motivation of George Boole's elective symbols, one sees that he applies them as selecting operators on classes of objects. As outlined in [4] expressions which do not represent classes are called by George Boole "uninterpretable", and are formally recognizable as those which do not satisfy the idempotence law $x^2 = x$. Characteristic of the method is that while expressions may be uninterpretable, equations always are when suitably interpreted by rules.

But in his first book [1] he was limited by the interpretation of the number 1 which he considered as the unique class U representing the whole universe. Because of this, without going into all the details, see for example [5,6], he changed the method in his second book in 1854 [2] and applied the formalism to subclasses of the universal class U.

Modern terminology will be used to describe what George Boole was doing: the word *class* should be used as a synonym for the modern word *set*. In [1] he starts with the universe class U and looks successively in [2] at the collection $P(U)$ of subclasses. The definition of the selection (*i.e. elective*) operator S_A defined for $P(U) \to P(U)$ for $A \in P(U)$ acting for $X \in P(U)$ is given, by the intersection:

$$S_A(X) = A \cap X \tag{4.1}$$

Using composition of operators for multiplication, his operators were associative, commutative and idempotent. Letting 0 be the empty class, 1 the universe U, one has $S_0(X) = 0$, $S_1(X) = X$. Addition was partially defined, namely $S_A + S_B$, was defined for $A \cap B = 0$. Likewise subtraction was also partially defined.

When considering all the laws that George Boole actually uses with the operations $(+, -, \times, 0, 1)$ can be viewed as a set of axioms for a mathematical theory, Theodore Hailperin finds [5] that the correct interpretations or models are obtained if one considers, not classes, but multisets as the entities over which the variables range. The operators defined here-above carry over to *signed multisets*, which are conveniently expressed as a map $f : U \to \mathbb{Z}$. Then George Boole's classes correspond to characteristic functions by the means of the map $\alpha : \Lambda \to \hat{\Lambda}$, where $\hat{\Lambda}(u)$ is 1 if $u \in \Lambda$ and 0 otherwise. The collection of maps from U to $\mathbb{Z}$ is usually written as $\mathbb{Z}^U$, a ring of functions with scalar multiplication (by elements of $\mathbb{Z}$), where the operations are given pointwise, that is, for $u \in U$. Boole's election operators S_A on $P(U)$ can thus be translated to corresponding operators which are the set of idempotent elements of the ring $\mathbb{Z}^U$.

If one wants to use linear operations on a vector space, one needs to extend the ring $\mathbb{Z}^U$ to a field F, since vector spaces are defined over fields, thus the set of idempotents $\{0, 1\}^U$, the ring of signed multisets $\mathbb{Z}^U$ and the algebra of functions F^U over F verify:

$$\{0, 1\}^U \subseteq \mathbb{Z}^U \subseteq F^U \tag{4.2}$$

The isomorphism between the ring $\mathbb{Z}^U$ restricted to its idempotent elements $\{0, 1\}^U$ and Boole's algebra of classes on $P(U)$ is due to Theodore Hailperin in [5]. His breakthrough was to point out this equivalence: the set of elements x of an algebra of signed multisets which satisfy $x^2 = x$ constitute a Boolean algebra. But most importantly all the axioms that were needed by Boole's (partial) algebra of logic hold in the complete algebra $\mathbb{Z}^U$. This means that Boole's equational reasoning was correct in $\mathbb{Z}^U$, and thus in his partial algebra $P(U)$. So finally, as is pointed out by Stanley Burris [17], much of Boole's work in logic had a solid foundation.

There is also an isomorphism between the ring of linear operators on F^U, restricted to those linear operators defined by left multiplication (*i.e.* ordered matrix product) by an idempotent element of F^U and Boole's algebra of selection operators S_A on $P(U)$. A linear operator on F^U that is defined by left multiplication by an idempotent is the same as the one given by left multiplication by a diagonal matrix with the idempotent characteristic function $\hat{\Lambda}$ along the diagonal.

From Theodore Hailperin's book [5] it is clear that given any commutative ring R with unity and without nilpotent elements one has parallels to all of George Boole's theorems, not just the development theorem, holding in the ring. One can think of such a ring as a ring of operators acting by left multiplication on R. Indeed R can be viewed as a unitary left R-module. Thus one also has parallels to Boole's results in [1].

If one takes the ring R to be the ring $\mathbb{Z}^N$ of N-tuples of integers, then the idempotent elements are the N-tuples with $\{0, 1\}$ entries. By identifying the N-tuple operators with $N \times N$ diagonal matrices (vector space of dimension $d = N$), and the elements of the ring with column vectors, one gets the linear algebra situation treated hereafter. It must be outlined that because of binary cardinality one has here $d = N = 2^n$.

4.3. The Seed Projection Operator and One Argument Operators

As stated above the elective symbols represent operators acting on a given class of objects (a subclass $P(U)$ of the universe class U). In this way the elective operator represented by the number 1 will simply become the identity operator for the considered subclass. Using the framework of linear algebra, operators are defined on a vector space whose dimension depends on the number of arguments (the arity) in the propositional system.

So what operators can represent the selection of elements out of a class? The straightforward answer in linear algebra are the projection operators which have the property of idempotence.

Considering the case of objects belonging to one single class, the corresponding projection operator $\mathbf{\Pi}$ of this class will act on vectors. Now what are the expected outcomes when applying this projection operator? If a vector $|a>$ corresponds exactly to elements of the class, the following matrix equations will be verified:

$$\mathbf{\Pi} \cdot |a >= 1 \cdot | a > \qquad\qquad \mathbf{\Pi} \cdot |\overline{a} >= 0 \cdot | \overline{a} > \tag{4.3}$$

The Dirac *bra-ket* notation, where $|\psi>$ is the *ket*, is the usual notation in quantum mechanics, representing quantum state vectors, which are orthonormal vectors. Here we use this notation to represent our logical vectors. The values 0 and 1 are the two eigenvalues of the two projection operators associated with the two eigenvectors. As before, if interpretable results are to be considered in logic, the only possible numbers for these eigenvalues are 0 and 1.

1 is obtained for objects belonging to the considered class and 0 for objects not belonging to it. In the last case one defines the complement vector $|\overline{a}>$.

The *True* eigenvalue 1 will correspond to the eigenvector $|a>$, named $|1>$, and the *False* eigenvalue 0 will correspond to the complementary eigenvector $|\overline{a}>$ named $|1>$.

When these properties are expressed in matrix form the projection operators $\mathbf{\Pi}_{(1)}$ and $\mathbf{\Pi}_{(0)}$ are 2×2 square matrices and the vectors $|a>$ and $|\overline{a}>$ are 2 dimensional orthonormal column vectors:

$$\mathbf{\Pi}_{(1)} = \mathbf{\Pi} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \qquad \mathbf{\Pi}_{(0)} = \mathbf{I}_2 - \mathbf{\Pi} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \tag{4.4}$$

$$|a> = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad |\overline{a}> = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{4.5}$$

The two projection operators given in Eq. (4.4) are complementary and idempotent, this last condition is written:

$$\mathbf{\Pi} \cdot \mathbf{\Pi} = \mathbf{\Pi}^2 = \mathbf{\Pi} \tag{4.6}$$

One can then construct the 4 logical operators corresponding to the 4 elective functions given in Table 1 corresponding to the single argument case $n = 1$. Capital bold letters are used here to represent operators.

$$\boldsymbol{A} = \mathbf{\Pi} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \qquad \bar{\boldsymbol{A}} = \mathbf{I}_2 - \mathbf{\Pi} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$
$$\boldsymbol{T} = \mathbf{I}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \boldsymbol{F} = \mathbf{0}_2 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{4.7}$$

$\boldsymbol{A}$ is the *logical projector* and $\bar{\boldsymbol{A}}$ its complement. The $\boldsymbol{T}$ (True, Tautology) operator corresponds here to the identity operator in 2 dimensions $\boldsymbol{I}_2$. $\boldsymbol{F}$ (False, Contradiction) corresponds here to the nil operator $\mathbf{0}_2$.

Remark that $\boldsymbol{I}_2$ and $\mathbf{0}_2$ are also projection operators (idempotent). So in general for one argument the matrix form of the projection operator corresponding to the logical function $f_i^{[1]}(x)$ given on Table 1 is:

$$\boldsymbol{F}_i^{[1]} = f_i^{[1]}(0)\, \mathbf{\Pi}_{(0)} + f_i^{[1]}(1)\, \mathbf{\Pi}_{(1)} = \begin{pmatrix} f_i^{[1]}(0) & 0 \\ 0 & f_i^{[1]}(1) \end{pmatrix} \tag{4.8}$$

This equation represents the *spectral decomposition* of the operator and because the eigenvalues are real the logical operator is *Hermitian* and can thus be considered as an *observable*. In this way, in Eigenlogic, the truth values of the logical proposition are the eigenvalues of the logical operator. In the very simple case where 0 and 1 are both not degenerate eigenvalues, the projection operators relative to the eigenvector basis take the form of the logical projection operator $\boldsymbol{A}$ and its complement $\bar{\boldsymbol{A}}$. As is done in quantum mechanics one can find the set of projection operators that completely represent the system, in particular by lifting the eventual degeneracy of the eigenvalues. Here eigenvalues are always equal to 0 or 1 and the question about the multiplicity of eigenvalues is natural. This last point is important in the model, because not only mutually exclusive projection operators are representative of a logical system, the complete family of commuting projection operators (the *logical family*) must be used in order to completely define the logical system. When these properties are expressed in matrix terms this means that the matrix product of the logical commuting projection operators is not necessarily equal to 0.

4.4. Extending to More Arguments

As seen above when representing logic with n arguments (n-arity) using idempotent projection operators various possibilities are intrinsically present in a unique structure with 2^{2^n} different projection operators. Once the eigenbasis is chosen the remaining structure is intrinsic thus basis independent.

The extension to more arguments can be obtained by increasing the dimension, this is done by using the Kronecker product $\otimes$. It is a standard procedure in linear algebra justified because it can be shown (Wedderburn little theorem [34]) that any finite division ring (a divison ring is the analogue of a field without necessitating commutativity) is a direct product of Galois fields $GF_p = \mathbb{Z}/\mathbb{Z}_p$ (p prime), in the binary case considered here $p = 2$. The direct product becomes explicitly the tensor or Kronecker product of linear operators.

In this work the application of this method was originally inspired from the composition rule of quantum states, which has the status nowadays of postulate in quantum mechanics [27] where the quantum state vector corresponding to the composition of two quantum systems represented by two subspaces in Hilbert space, is the Kronecker product of the respective quantum state vectors. The operators acting in the combined space are combinations of the quantum operators in the respective sub-spaces. The interesting fact is that for the combined case new structures appear, named *non-local*, that cannot be put as simple Kronecker products but are linear combinations of these. It will be shown that several projection operators presented hereafter are not simply Kronecker products of elementary projection operators but can be linear combinations of these.

In the following, as before for the elective logical functions, superscripts are used in order to indicate how many arguments are used (arity) in the propositional system.

One can verify that in Eq. (4.7) all the four logical operators are effectively idempotent and commuting. The correspondence of the elective symbol x with the elementary *seed* projection operator $\mathbf{\Pi}$ will be used in the following to build higher arity logical operators.

For 2 arguments (arity $n = 2$) one needs 4 commuting orthogonal rank-1 projection operator operators in order to express the development in the same way as in Eq. (2.10).

Two properties of the Kronecker product with idempotent projection operators have to be outlined.

- The Kronecker product of two projection operators is also a projection operator.
- If the projection operators have a rank equal to 1 (a single eigenvalue is 1 and all the others are 0) then their Kronecker product is also a rank-1 projection operator.

Using these two properties, the 4 commuting orthogonal rank-1 projection operators spanning the 4 dimensional vector space can be calculated in a straightforward way:

$$\mathbf{\Pi}^{[2]}_{(0,0)} = (\mathbf{I}_2 - \mathbf{\Pi}) \otimes (\boldsymbol{I}_2 - \mathbf{\Pi}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{\Pi}^{[2]}_{(0,1)} = (\mathbf{I}_2 - \mathbf{\Pi}) \otimes \mathbf{\Pi} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{\Pi}^{[2]}_{(1,0)} = \mathbf{\Pi} \otimes (\mathbf{I}_2 - \mathbf{\Pi}) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{\Pi}^{[2]}_{(1,1)} = \mathbf{\Pi} \otimes \mathbf{\Pi} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.9}$$

By the same procedure as in Eq. (2.10), one can write the operators for $n = 2$ arguments for a two-argument function (see Table 2) using the projection operators given in Eq. (4.9):

$$\boldsymbol{F}^{[2]}_i = \begin{pmatrix} f^{[2]}_i(0,0) & 0 & 0 & 0 \\ 0 & f^{[2]}_i(0,1) & 0 & 0 \\ 0 & 0 & f^{[2]}_i(1,0) & 0 \\ 0 & 0 & 0 & f^{[2]}_i(1,1) \end{pmatrix} \tag{4.10}$$

The coefficients (co-factors) are the logical function's truth values given on Table 2.

This method can be extended to whatever number of arguments n using the same seed projection operator $\mathbf{\Pi}$ and its complement $(\mathbf{I}_2 - \mathbf{\Pi})$.

4.5. Logical Operators for Two Arguments

For arity $n = 2$ the polynomial expressions have already been calculated in Table 2, so one can write down directly the corresponding operators. One has to express the logical projectors corresponding to the two arguments $x = a$ and $y = b$ and this is given using the formulation (4.10) by considering the truth values of the functions $f^{[2]}_{12}$ and $f^{[2]}_{10}$, these operators are:

$$\boldsymbol{A}^{[2]} = \boldsymbol{F}^{[2]}_{12} = 1 \cdot \mathbf{\Pi}^{[2]}_{(1,0)} + 1 \cdot \mathbf{\Pi}^{[2]}_{(1,1)} = \mathbf{\Pi} \otimes \mathbf{I}_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.11}$$

$$\boldsymbol{B}^{[2]} = \boldsymbol{F}^{[2]}_{10} = 1 \cdot \mathbf{\Pi}^{[2]}_{(0,1)} + 1 \cdot \mathbf{\Pi}^{[2]}_{(1,1)} = \mathbf{I}_2 \otimes \mathbf{\Pi} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.12}$$

Here are some examples: the conjunction operator for $n = 2$ will simply be the product of the two logical projectors:

$$\boldsymbol{F}^{[2]}_{AND} = \boldsymbol{A}^{[2]} \cdot \boldsymbol{B}^{[2]} = (\boldsymbol{\Pi} \otimes \mathbf{I}_2) \cdot (\mathbf{I}_2 \otimes \boldsymbol{\Pi}) = \boldsymbol{\Pi} \otimes \boldsymbol{\Pi} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.13}$$

where the following particular distributive property of the Kronecker product has been used: if $\boldsymbol{P}$, $\boldsymbol{Q}$, $\boldsymbol{R}$ and $\boldsymbol{S}$ are operators having the same dimension:

$$(\boldsymbol{P} \otimes \boldsymbol{Q}) \cdot (\boldsymbol{R} \otimes \boldsymbol{S}) = (\boldsymbol{P} \cdot \boldsymbol{R}) \otimes (\boldsymbol{Q} \cdot \boldsymbol{S}) \tag{4.14}$$

The disjunction operator can be directly written, using Eq. (3.4):

$$\boldsymbol{F}^{[2]}_{OR} = \boldsymbol{A}^{[2]} + \boldsymbol{B}^{[2]} - \boldsymbol{A}^{[2]} \cdot \boldsymbol{B}^{[2]} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.15}$$

The exclusive disjunction can also be directly written, using Eq. (3.5):

$$\boldsymbol{F}^{[2]}_{XOR} = \boldsymbol{A}^{[2]} + \boldsymbol{B}^{[2]} - 2\boldsymbol{A}^{[2]} \cdot \boldsymbol{B}^{[2]} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{4.16}$$

Negation is obtained by subtracting from the identity operator (complementation) giving in general for n arguments:

$$\bar{\boldsymbol{A}}^{[n]} = \mathbf{I}_{2^n} - \boldsymbol{A}^{[n]} \tag{4.17}$$

this equation can be used to obtain the $NAND$ operator:

$$\boldsymbol{F}^{[2]}_{NAND} = \mathbf{I}_4 - \boldsymbol{F}^{[2]}_{AND} = \mathbf{I}_4 - \boldsymbol{A}^{[2]} \cdot \boldsymbol{B}^{[2]} \tag{4.18}$$

Using De Morgan's law:

$$\boldsymbol{F}^{[2]}_{NOR} = \mathbf{I}_4 - \boldsymbol{A}^{[2]} - \boldsymbol{B}^{[2]} + \boldsymbol{A}^{[2]} \cdot \boldsymbol{B}^{[2]} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = (\mathbf{I}_2 - \boldsymbol{\Pi}) \otimes (\mathbf{I}_2 - \boldsymbol{\Pi}) \tag{4.19}$$

Material implication is also straightforwardly obtained using the expression given in Table 2:

$$\boldsymbol{F}^{[2]}_{\Rightarrow} = \mathbf{I}_4 - \boldsymbol{A}^{[2]} + \boldsymbol{A}^{[2]} \cdot \boldsymbol{B}^{[2]} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \mathbf{I}_4 - (\boldsymbol{\Pi}) \otimes (\mathbf{I}_2 - \boldsymbol{\Pi}) \tag{4.20}$$

On Table 3 are given the logical operator forms for the 16 two-argument logical connectives.

4.6. Logical Operators for Three Arguments

For arity $n = 3$ one can generate 8 orthogonal 8-dimensional rank-1 projection operators, for example two of these are given, by:

$$\mathbf{\Pi}^{[3]}_{(1,1,1)} = \mathbf{\Pi} \otimes \mathbf{\Pi} \otimes \mathbf{\Pi} \qquad \mathbf{\Pi}^{[3]}_{(0,1,0)} = (\mathbf{I}_2 - \mathbf{\Pi}) \otimes \mathbf{\Pi} \otimes (\mathbf{I}_2 - \mathbf{\Pi}) \tag{4.21}$$

The logical projectors are:

$$\boldsymbol{A}^{[3]} = \mathbf{\Pi} \otimes \mathbf{I}_2 \otimes \mathbf{I}_2 \qquad \boldsymbol{B}^{[3]} = \mathbf{I}_2 \otimes \mathbf{\Pi} \otimes \mathbf{I}_2 \qquad \boldsymbol{C}^{[3]} = \mathbf{I}_2 \otimes \mathbf{I}_2 \otimes \mathbf{\Pi} \tag{4.22}$$

For arity $n = 3$ the conjunction AND becomes then straightforwardly:

$$\boldsymbol{F}^{[3]}_{AND} = \boldsymbol{A}^{[3]} \cdot \boldsymbol{B}^{[3]} \cdot \boldsymbol{C}^{[3]} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.23}$$

For arity $n = 3$ the majority MAJ operator will be a 8×8 matrix, its expression can written directly using Eqs. (3.11) and (4.22):

$$\boldsymbol{F}^{[3]}_{MAJ} = \boldsymbol{A}^{[3]} \cdot \boldsymbol{B}^{[3]} + \boldsymbol{A}^{[3]} \cdot \boldsymbol{C}^{[3]} + \boldsymbol{B}^{[3]} \cdot \boldsymbol{C}^{[3]} - 2\boldsymbol{A}^{[3]} \cdot \boldsymbol{B}^{[3]} \cdot \boldsymbol{C}^{[3]}$$
$$= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.24}$$

4.7. Selection Operators

The method for selecting eigenvalues is similar to the one for elective functions given in Eq. (2.11). Because the projection operators of the type $\mathbf{\Pi}^{[n]}_{(a,b,\dots)}$ are rank-1 projecton operators, the product (matrix product) with whatever other commuting projection operator (for example the logical operator $\boldsymbol{F}^{[n]}_i$) will also give a rank-1 projection operator and more precisely this will be the same projection operator multiplied by the eigenvalue. So for whatever logical operator $\boldsymbol{F}^{[n]}_i$ of the considered family one has:

$$\boldsymbol{F}^{[n]}_i \cdot \mathbf{\Pi}^{[n]}_{(a,b,\dots)} = f^{[n]}_i(a, b, \dots)\, \mathbf{\Pi}^{[n]}_{(a,b,\dots)} \tag{4.25}$$

On the right of Eq. (4.25) the truth value is multiplied by the corresponding rank 1 projection operator.

TABLE 3. The sixteen two-argument connectives and the respective Eigenlogic operators

Connective for Boolean $\boldsymbol{A}\,,\boldsymbol{B}$	Operator diagonal form $diag$(truth vector)	Logical operator $\boldsymbol{F}_i^{[2]}$ in the $\boldsymbol{A}\,,\boldsymbol{B}$ argument form	Logical operator $\boldsymbol{F}_i^{[2]}$ in the seed operator $\boldsymbol{\Pi}$ form
False F	$diag(0,0,0,0)$	$\mathbf{0}$	$\mathbf{0}$
NOR ; $\overline{A \vee B}$	$diag(1,0,0,0)$	$\mathbf{I} - \boldsymbol{A} - \boldsymbol{B} + \boldsymbol{A}\cdot\boldsymbol{B}$	$(\mathbf{I} - \boldsymbol{\Pi}) \otimes (\mathbf{I} - \boldsymbol{\Pi})$
$\overline{A \Leftarrow B}$	$diag(0,1,0,0)$	$\boldsymbol{B} - \boldsymbol{A}\cdot\boldsymbol{B}$	$\boldsymbol{\Pi} \otimes (\mathbf{I} - \boldsymbol{\Pi})$
$\overline{A}$	$diag(1,1,0,0)$	$\mathbf{I} - \boldsymbol{A}$	$\mathbf{I} - (\boldsymbol{\Pi} \otimes \mathbf{I})$
$\overline{A \Rightarrow B}$	$diag(0,0,1,0)$	$\boldsymbol{A} - \boldsymbol{A}\cdot\boldsymbol{B}$	$(\mathbf{I} - \boldsymbol{\Pi}) \otimes \boldsymbol{\Pi}$
$\overline{B}$	$diag(1,0,1,0)$	$\mathbf{I} - \boldsymbol{B}$	$\mathbf{I} - (\mathbf{I} \otimes \boldsymbol{\Pi})$
$A \oplus B$	$diag(0,1,1,0)$	$\boldsymbol{A} + \boldsymbol{B} - 2\boldsymbol{A}\cdot\boldsymbol{B}$	$\boldsymbol{\Pi} \otimes (\mathbf{I} - \boldsymbol{\Pi}) + (\mathbf{I} - \boldsymbol{\Pi}) \otimes \boldsymbol{\Pi}$
NAND ; $\overline{A \wedge B}$	$diag(1,1,1,0)$	$\mathbf{I} - \boldsymbol{A}\cdot\boldsymbol{B}$	$\mathbf{I} - (\boldsymbol{\Pi} \otimes \boldsymbol{\Pi})$
AND ; $A \wedge B$	$diag(0,0,0,1)$	$\boldsymbol{A}\cdot\boldsymbol{B}$	$\boldsymbol{\Pi} \otimes \boldsymbol{\Pi}$
$A \equiv B$	$diag(1,0,0,1)$	$\mathbf{I} - \boldsymbol{A} - \boldsymbol{B} + 2\boldsymbol{A}\cdot\boldsymbol{B}$	$\boldsymbol{\Pi} \otimes \boldsymbol{\Pi} + (\mathbf{I} - \boldsymbol{\Pi}) \otimes (\mathbf{I} - \boldsymbol{\Pi})$
B	$diag(0,1,0,1)$	$\boldsymbol{B}$	$\mathbf{I} \otimes \boldsymbol{\Pi}$
$A \Rightarrow B$	$diag(1,1,0,1)$	$\mathbf{I} - \boldsymbol{A} + \boldsymbol{A}\cdot\boldsymbol{B}$	$\mathbf{I} - [(\mathbf{I} - \boldsymbol{\Pi}) \otimes \boldsymbol{\Pi}]$
A	$diag(0,0,1,1)$	$\boldsymbol{A}$	$\boldsymbol{\Pi} \otimes \mathbf{I}$
$A \Leftarrow B$	$diag(1,0,1,1)$	$\mathbf{I} - \boldsymbol{B} + \boldsymbol{A}\cdot\boldsymbol{B}$	$\mathbf{I} - [\boldsymbol{\Pi} \otimes (\mathbf{I} - \boldsymbol{\Pi})]$
OR ; $A \vee B$	$diag(0,1,1,1)$	$\boldsymbol{A} + \boldsymbol{B} - \boldsymbol{A}\cdot\boldsymbol{B}$	$\mathbf{I} - [(\mathbf{I} - \boldsymbol{\Pi}) \otimes (\mathbf{I} - \boldsymbol{\Pi})]$
True T	$diag(1,1,1,1)$	$\mathbf{I}$	$\mathbf{I}$

To get explicitly the eigenvalue one can take the trace of the product of the two operators on the left of Eq. (4.25) as shown hereafter in Eq. (5.3). In this way one obtains the truth value $f_i^{[n]}(a, b, ...)$ corresponding to a case of a fixed combination of the values $(a, b, ...)^{[n]}$ of the logical arguments (an *interpretation*).

5. Eigenvectors, Eigenvalues and Truth Values

Starting with the two-dimensional rank-1 projection operators $\mathbf{\Pi}$ for the one-argument case, the eigenvectors $|0>$ and $|1>$ are 2-dimensional orthonormal column vectors as shown in Eq. (4.5).

The choice of the position of value 1 in the column vector follows the quantum information convention for a "qubit-1" [27]. The the column vectors for the "0" and "1" are:

$$|0>= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1>= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{5.1}$$

For the two-argument case $n = 2$ the vectors have the dimension $2^{n=2} = 4$ and the complete family of 16 commuting projection operators represents all possible logical propositions and will be interpretable when applied on the four possible orthonormal eigenvectors of this family that form the complete canonical basis. These vectors will be represented by the symbol $|ab>$, where the arguments a, b take the values $\{0, 1\}$ and represent one of the four possible cases:

$$|00>= \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad |01>= \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad |10>= \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad |11>= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{5.2}$$

When applying the logical projection operators on these vectors the resulting eigenvalue is the truth value of the corresponding logical proposition meaning that operations on the eigenspace of a logical operator family are interpretable. For example for $n = 2$ arguments the complete family of 16 commuting logical operators represents all possible logical connectives. Operations are interpretable when applied to one of the four possible canonical eigenvectors of the family, $|00>$, $|01>$, $|10>$ and $|11>$, representing all the possible interpretations. These vectors form a complete orthonormal basis.

The operation of selection of a truth value can be undertaken using this formalism

$$f_i^{[n]}(a, b, ...) = \langle a, b, ...| \, \boldsymbol{F}_i^{[n]} \cdot \mathbf{\Pi}_{(a,b,...)}^{[n]} \, |a, b, ...\rangle = \mathrm{Tr}\left[\boldsymbol{F}_i^{[n]} \cdot \mathbf{\Pi}_{(a,b,...)}^{[n]}\right] \tag{5.3}$$

this is actually the definition of the mean value in quantum mechanics and is generalized for non-eignevectors in the Born rule giving a probability measure.

What happens when the state-vector is not one of the eigenvectors of the logical system? One can always express a normalized vector as a decomposition on a complete orthonormal basis. In particular one can express it over the canonical eigenbasis of the logical operator family. For two-arguments this vector can be written as:

$$|\phi >= c_{00}\,|00> \,+ c_{01}\,|00> \,+ c_{10}\,|00> \,+ c_{11}\,|00> \tag{5.4}$$

When only one of the coefficients is non-zero (in this case its absolute value must take the value 1) then one is back in the preceding situation of a determinate interpretation (determinate input atomic propositional case). But when more than one coefficient is non-zero one is in a "mixed" or "fuzzy" case. Such a state can be considered as a coherent superposition of interpretations. This can lead to a fuzzy-logic treatment as is proposed in [13,15]. Fuzzy Logic deals with truth values that may be any number between 0 and 1, here the truth of a proposition may range between completely true and completely false and this value can be considered a probability, which is a number between 0 and 1, given by the mean value calculated using the expression (5.3).

An important remark is that the choice of the eigenbasis is not fixed, meaning that for every choice there is a complete family of logical projection operators, so as stated above one could imagine working with two (or more) logical systems characterized each by their family of projective operators. The operators of one family do not (generally) commute with the operators of another family.

This non-commuting property has its analogue in the general quantum mechanical formalism. Without extending this argument further one sees the potentiality of considering this kind of approach keeping in mind that in linear algebra basis change is obtained by means of unitary operators and this is somewhat at the heart of quantum computation [27] where all logical operations are done by means of unitary transformations and by measurements using projection operators. A discussion for a proposal, using non-commutative operators, to extend to first-order logic has been given in [14] and in [15] where an *Eigenlogic program* is developed showing links with quantum computing and quantum information.

6. Properties of Eigenlogic

To summarize, all the logical projection operators have the following properties in Eigenlogic.

1. All logical operators are idempotent projection operators. This means that in the logical eigenbasis the matrices are diagonal with eigenvalues either 0 or 1.
2. All the logical projection operators of a given logical family are commutative pairwise. This means that all the respective matrices are diagonal on the same logical eigenbasis. The logical projection operators are not necessarily orthogonal. This means that the matrix product of two logical operators is not necessarily the nil operator.
3. The logical family represents a complete system of logical propositions. The number of different logical projection operators of a given family is 2^{2^n}. This number corresponds to the number of different commuting

diagonal matrices obtained for all the combinations of 0's and 1's on the diagonal of the matrices.

4. The dimension of the vector space spanned by the logical operators is $d_n = 2^n$. Where n is the arity of the logical system. All logical projection operators of the same family are $d_n \times d_n$ square matrices.
5. For each family there are 2^n orthogonal rank-1 projection operators spanning the entire vector space. The corresponding matrices will have a single eigenvalue of value 1, the other eigenvalues being 0.
6. Every rank-1 projection operator of the family can be obtained by the means of the Kronecker product, the unique seed projection operator $\mathbf{\Pi}$ and its complement $(\mathbf{I}_2 - \mathbf{\Pi})$ (see Eqs. (4.4), (4.9) and (4.21)).
7. Every logical projection operator can be expressed as an elective decomposition using the 2^n orthogonal rank-1 projection operators, where the coefficients of the decomposition can only take the values 0 or 1 (see formulation (4.10) for $n = 2$).
8. The negation of a logical operator, which is its complement, is obtained by subtracting the operator from the identity operator (see Eq. (4.17)).
9. The eigenvectors of the family of the n-arity commuting logical projection operators form an orthonormal complete basis of dimension $d_n = 2^n$. This basis corresponds to the canonical basis and each eigenvector corresponds to a given unique combination of logical arguments, named an *interpretation* of the logical propositional system.
10. The eigenvalues of the logical operators are the truth values of the respective logical proposition and each eigenvalue is associated to a given eigenvector corresponding to an interpretation of the input atomic propositions.
11. The truth value of a given logical operator for a given interpretation of n arguments can be obtained using Eq. (4.25).

7. Related Research

Attempts to link geometry to logic are very numerous and date back to the first efforts to formalize logic. A famous example is Aristotle's *Square of Opposition* using 4 categorical propositions in the form of *subject-copula-predicate*, which has recently aroused renewed interest in logic (for a thorough discussion see for example [36] and for research activities see [37]). Other examples are Leonhard Euler's (1707-1783) diagrams illustrating propositions and quantifiers (all, no, some,...), C. L. Dodgson's (alias Lewis Carroll 1832-1898) diagrams seeking symmetry for true and false having a striking resemblance with modern Karnaugh maps and of course the methods developed by John Venn [19] which were mentioned above.

In modern logic design methods, truth tables, Karnaugh maps, hypercubes, logic and threshold networks, decision trees and diagram graphs, are extensively used for representing Boolean data structures [22]. Logical reduction based on symmetry is a very important topic which uses Hesse diagrams, Shannon and Davio expansions and the Post theorems on symmetries of Boolean functions. Vectorization is also a standard procedure in logic for example using *truth vectors* and *carrier vectors* (reduced truth vectors of symmetric Boolean functions).

In the following are briefly quoted recent researches which came up during this investigation and which support the approach based on linear algebra presented in this paper.

Starting with *Matrix Logic* developed by August Stern [38] which gives directly a matrix formulation for logical operators, by putting the truth values as matrix coefficients, in the way of Karnaugh diagrams. So for example a two argument logical function becomes a 2×2 matrix, this is a fundamental difference when compared with the method given here above where 4×4 matrices are used. Using scalar products on vectors and mean values on operators, this formalism gives a method to resolve logical equations and allows to enlarge the alphabet of the truth-values with negative logic antivalues.

A breakthrough has been undoubtedly made by *Vector Logic* developed by Eduardo Mizraji [39]. This approach vectorizes logic where the truth values map on orthonormal vectors. Technically this approach is different from the one presented in this paper because the resulting operators for 2 arguments are represented by 2×4 matrices and do not represent projection operators. Vector logic can also handle three-valued logic and applications have been proposed for neural networks.

A very pertinent development, which is close to the approach in this paper, was done by Vannet Aggarwal and Robert Caldebrabnk [40] in the framework of quantum error-coding theory, their work was also justified by the *Projection Logic* formulation of David Cohen [41]. In their method they connect Boolean logic to projection operators derived initially from the Heisenberg-Weyl group. They associate the dimension of the considered projection operator with the Hamming weight (number of 1's in the truth table) of the corresponding Boolean function. The logical operators they obtain are commuting projection operators, as in the work presented here.

Recently the concept of *quantum predicate* introduced by E. D'Hondt and P. Panangaden [42] proposes an interpretation similar to the one presented here. As stated by Mingsheng Ying in [43]: "In classical logic, predicates are used to describe properties of individuals or systems... then what is a quantum predicate?" ; "... a quantum predicate is defined to be a physical observable represented by a Hermitian operator with eigenvalues within the unit interval".

8. Discussion

In the formulation given here a general method in logic is proposed, enabling the construction of general logical projection operators from a single seed projection operator using the Kronecker product. It gives also a simpler formulation because George Boole's elective interpretation of logic shows that the idempotence property (2.3) and (2.4) in association with distributivity (2.1) and commutativity (2.2) permit to identify directly commuting projection operators with logical functions.

The formulation of logic presented here uses operators in linear algebra as propositions and is linked to the formulation of elective symbolic algebra of George Boole in [1]. This similarity is striking and is more than just an analogy, as justified here-above, at the heart of this is the idempotence property of projectors. The logical operators belong to families of commuting projection operators. The interesting feature is that the eigenvalues of these operators are the truth values of the corresponding logical connective, the associated eigenvectors corresponding to one of the fixed combination of the logical inputs corresponding to the interpretations of the logical system. The outcome of a "measurement" or "observation" on a logical operator will give the truth value of the associated logical proposition, and becomes "interpretable" when applied to its eigenspace leading to a natural analogy with the measurement postulate in Quantum Mechanics.

The following diagram summarizes this point of view:

$$\begin{aligned} \text{logical operators} &\longrightarrow \text{logical connectives} \\ \text{operator eigenvalues} &\longrightarrow \text{logical truth values} \\ \text{operator eigenvectors} &\longrightarrow \text{logical interpretations} \end{aligned}$$

Some precisions must be given concerning the last line of the diagram, the word *intepretation* is meant in the way used in logic: an *interpretation* is an assignment of truth values for each atomic proposition that occurs in a *well-formed formula*. A *well-formed formula* being a complex formula containing exclusively logical connectives. This means that the set of atomic propositions can have different interpretations, the ones leading to the *satisfaction* of a logical proposition (a proposition is satisfied when it is true) are called the *models* (*n.b.* sometimes the word model is used more generally as a synonymous of the word interpretation).

Because of the central role played by eigenvalues (truth values) and eigenvectors (interpretations) in this approach, this formulation has been named *Eigenlogic*.

A theoretical justification and a link to quantum mechanics can also be found in Pierre Cartier [44], relating the link between the algebra of logical propositions and the set of all valuations on it, he writes: "...in the *theory of models* in logic a model of a set of propositions has the effect of validating certain propositions. With each logical proposition one can associate by duality the set of all its true valuations represented by the number 1. This correspondence makes it possible to interpret the algebra of propositions as a class of subsets, conjunction and disjunction becoming respectively the intersection and union of sets. This corresponds to the *Stone duality* proved by the Stone representation theorem and is one of the spectacular successes of twentieth century mathematics....The development of quantum theory led to the concept of a quantum state, which can be understood as a new embodiment of the concept of a valuation". The idea is not new, as was discussed before and also developed in [34], and stems from John Von Neumann's proposal of "projections as propositions" in [33] which was subsequently formalized in quantum logic with Garret Birkhoff in [32].

Concerning the second line of the diagram one can generalize to eigenvalues different from the Boolean couple $\{0, 1\}$ associated to projection operators. For example one can consider the alphabet $\{+1, -1\}$ associated to self-inverse unitary operators (involutions) obtained from the logical projection operators by the *quantum Householder transform* [45], this has been proposed in [13] showing that this logical alphabet has a direct correspondence with quantum spin.

In general one can associate a binary logical operator with whatever couple of distinct eigenvalues $\{\lambda_1, \lambda_2\}$, the corresponding family of logical operators can then be found straightforwardly by matrix interpolation methods as thoroughly discussed in [13,15,35].

The extension to multivalued alphabets using specific logical operators is also natural, this was proposed in [13,15,35].

In propositional logic the elementary arguments of a compound logical proposition with more than one argument ($n \geq 2$), are the *atomic propositions*. In Eigenlogic these are the *logical projectors* (also sometimes named *dictators* in logic [13,46]). Examples are the two two-argument logical projectors $\boldsymbol{A}^{[2]}$ and $\boldsymbol{B}^{[2]}$ in equations (4.11, 4.12); the three three-argument logical projectors $\boldsymbol{A}^{[3]}$, $\boldsymbol{B}^{[3]}$ and $\boldsymbol{C}^{[3]}$ in Eq. (4.22) and so on for higher arity.

This is a fundamental difference with what is commonly considered in quantum logic where atomic propositions are associated with rays (rank-1 projection operators) and correspond to pure quantum state density matrices (for a definition of atomic propositions in quantum logic see *e.g.* [34] p. 98). Whereas, on the other hand, in Eigenlogic a ray corresponds, for example, to logical conjunction (AND, $\wedge$), which is a non-atomic connective, (see equations (4.13, 4.23)). The other $n - 1$ rays are simply obtained by complementing selectively the arguments of the conjunction (De Morgan's theorem). In Eigenlogic rays correspond to Kronecker products of generating projection operators, the *seed operators*, as shown in equations (4.9, 4.21) and are non-atomic.

In propositional logic atomic propositions must be independent propositions. Independence can only be achieved with the formulation for the atomic projectors obtained by the extension with the identity operator using the Kronecker product given in (4.11) and (4.12). On the other hand the rays, which are mutually exclusive rank-1 projection operators, do not correspond to independent propositions. Thus for Eigenlogic atomic propositions do not correspond to rays.

In this work complete logical families of commuting projection operators correspond to compatible propositions this is also a difference with quantum logic. As mentioned by David W. Cohen (p. 37 [41]) "A quantum logic is a logic with at least two propositions that are not compatible". In future research the interplay of logical operators which do not belong to the same compatible logical family of commuting operators will be considered, this could bring insights for quantum logic and quantum computation and addresses the important topic of quantum non-contextuality this is part of the Eigenlogic program presented in [15]. For example an extension to first order logic is proposed using two maximally incompatible logical families such as those generated by the Pauli $\boldsymbol{X}$ and $\boldsymbol{Z}$ spin operators.

9. Conclusion and Perspectives

An algorithmic approach for logical connectives with a large number of arguments could be interesting to develop using the Eigenlogic operators in high-dimensional vector spaces. But because the space grows in dimension very quickly, it may not be particularly useful for practical implementation without logical reduction. It would be interesting to develop specific algebraic reduction methods for logical operators inspired from actual research in the field. For a good synthesis of the state of the art, *e.g.* [22].

The idea of linking logic and linear algebra is becoming natural because of the research effort due to the promise that quantum theory can bring to fields outside of physics, principally in computer science. Of course one must consider the quantum computer quest [27,35] but also more recent developments in other research areas such as semantic web information retrieval [47,48] and machine learning [49]. All these methods lie on linear algebra methods using vectors and operators in Hilbert space.

The *Quantum Interaction* community through annual conferences promotes links between quantum mechanics and fields outside physics with many applications in social sciences [50]. The methods are based upon the exploitation of the mathematical formalism, basely linear algebra in Hilbert space, of quantum mechanics [47] combined with the peculiar aspects of the quantum postulates. Applications are found in modern semantic theories such as distributional semantics or in connectionist models of cognition [51].

More generally this view of logic could bring some insight on more fundamental issues. Boolean functions are nowadays considered as a "toolbox" for resolving many problems in theoretical computer science, information theory and even fundamental mathematics. In the same way Eigenlogic could be considered as a new "toolbox".

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

[1] Boole, G.: The Mathematical Analysis of Logic. Being an Essay To a Calculus of Deductive Reasoning, 1847, (reissued Ed. Forgotten Books ISBN 978-1444006642-9) (1847)

[2] Boole, G.: An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities, Macmillan, 1854, (reissued by Cambridge University Press, 2009; ISBN 978-1-108-00153-3) (1854)

[3] Durand-Richard, M-J: Logic versus algebra: English debates and boole's mediation create. In: Gasser J (ed) A Boole anthology. Synthese library (studies in epistemology, logic, methodology, and philosophy of science), vol 291. Springer, Dordrecht (2000)

[4] Panteki, M.: The mathematical background of George Boole's mathematical analysis of logic (1847). In: Gasser J (eds) A Boole Anthology. Synthese Library (Studies in Epistemology, Logic, Methodology, and Philosophy of Science), vol 291. Springer, Dordrecht (2000)

[5] Hailperin, T.: Boole's Logic and Probability, a Critical Exposition from the Standpoint of Contemporary Logic and Probability Theory, North Holland (1976) II ed (1986)

[6] Hailperin, T.: Boole's Algebra isn't Boolean algebra. A description using modern algebra, of what boole really did create, Math. Mag. 54(4): 172–184 (1981). Reprinted in: Gasser J. (eds) A Boole Anthology. Synthese Library (Studies in Epistemology, Logic, Methodology, and Philosophy of Science), vol 291. Springer, Dordrecht (2000)

[7] Béziau, J.-Y.: Is the principle of contradiction a consequence of $x^2 = x$? Log. Univers. **12**, 55–81 (2018)

[8] Peirce, C.S.: On the algebra of logic: a contribution to the philosophy of notation. Am. J. Math. **7**, (1885)

[9] Menger, K.: Reminiscences of the Vienna Circle and the Mathemathical Colloquium, (1942), Editors: L. Golland B.F. McGuinness, Abe Sklar, Series: Vienna Circle Collection-Springer (1994)

[10] Post, E.L.: Introduction to a general theory of elementary propositions. Am. J. Math. **43**, 163–185 (1921)

[11] Wittgenstein, L.: Logisch-Philosophische Abhandlung", Annalen der Naturphilosophie, Ed. Wilhelm Ostwald, Wien: "Tractatus Logico-Philosophicus", translated and published in, bilingual edn, p. 1921. Routledge & Kegan Paul, London (1921)

[12] Corcoran, J.: Aristotle's prior analytics and Boole's laws of thought. Hist. Philos. Log. **24**, 261–288 (2003)

[13] Dubois, F., Toffano, Z.: Eigenlogic: A Quantum View for Multiple-Valued and Fuzzy Systems, Quantum Interaction. QI 2016. Lecture Notes in Computer Science, vol. 10106. Springer, Berlin, pp. 239–251 (2017)

[14] Toffano, Z.: "Eigenlogic", UNILOG 2018 (6th World Congress and School on Universal Logic), Workshop Logic and Physics, Vichy (F), June 16–26 (2018)

[15] Toffano, Z., Dubois, F.: Adapting logic to physics: the quantum-like eigenlogic program. Entropy **22**, 139 (2020)

[16] Knuth, D.E.: "The Art of Computer Programming", Vol. 4, Fascicle 0: Introduction to Combinatorial Algorithms and Boolean Functions, Ed. Addison-Wesley Professional (2009)

[17] Burris, S.: "The Laws of Boole's Thought". Manuscript (http://www.math.uwaterloo.ca/~snburris/htdocs/myworks/preprints/aboole.pdf), 2000, and private correspondence (2016)

[18] Huntington, E.V.: Sets of independent postulates for the algebra of logic. Trans. AMS **5**, 288–309 (1904)

[19] Venn, J.: Symbolic Logic. Macmillan and Company, London (1881)

[20] Schröder, E.: Vorlesungen über die Algebra der Logik, Vol. I, Anh. 6, B.G. Teubner, Leipzig (1890)

[21] Whitney, H.: Characteristic functions and the algebra of logic. Ann. Math. **34**, 40–414 (1933)

[22] Yanushkevich, S.N., Shmerko, V.P.: Introduction to Logic Design. CRC Press, London (2008)

[23] Aiken, H.: Synthesis of electronic computing and control circuits. Computation Laboratory of Harvard University, XXVII, Harvard University, Cambridge, MA (1951)

[24] Shannon, C.E.: A symbolic analysis of relay and switching circuits. Trans. AIEE **57**(12), 713–723 (1938)

[25] Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**(4) (1948)

[26] Nahin, P.J.: The Logician and the Engineer: How George Boole and Claude Shannon Created the Information Age. Princeton University Press, Princeton (2012)

[27] Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2000)

[28] Stone, M.H.: Linear Transformations in Hilbert Space and Their Applications to Analysis, p. 70: "Projections" (1932)

[29] Stone, M.H.: The theory of representation for Boolean algebras. Trans. Am. Math. Sdictatorociety **40**(1), 37–111 (1936)

[30] Stone, M.H.: Applications of the theory of Boolean rings to general topology. Trans. Am. Math. Soc. **41**(3), 375–481 (1937)

[31] Schlimm, D.: Bridging Theories with Axioms: Boole, Stone, and Tarski, New Perspectives on Mathematical Practices, World Scientific, Singapore, pp. 222–235 (2009)

[32] Birkhoff, G., Von Neumann, J.: The logic of quantum mechanics. Ann. Math. 2nd Ser. **37**(4), 823–843 (1936)

[33] Von Neumann, J.: "Mathematische Grundlagen der Quantenmechanik. Grundlehren der mathematischen Wissenschaften", volume Bd. 38. (Springer, Berlin, 1932) 106. "Mathematical Foundations of Quantum Mechanics". Investigations in Physics, vol. 2. (Princeton University Press, Princeton, 1955) (1932)

[34] David, F.: The Formalisms of Quantum Mechanics, An Introduction. Springer Lecture Notes in Physics, **ISBN 978-3-319-10538-3** (2015)

[35] Toffano, Z., Dubois, F.: Interpolating binary and multivalued logical quantum gates. MDPI Proc. **2**(4), 152 (2018)

[36] Moretti, A.: The Geometry of Logical Opposition. University of Neuchtel, Switzerland (2009). PhD Thesis

[37] https://www.square-of-opposition.org/

[38] Stern, A.: Matrix Logic. North-Holland, Amsterdam (1988)

[39] Mizraji, E.: Vector logics: the matrix-vector representation of logical calculus. Fuzzy Sets Syst. **50**, 179–185 (1992)

[40] Aggarwal, V., Calderbank, R.: Boolean functions, projection operators, and quantum error correcting codes. In: Proceedings International Symposium Information Theory, Nice, France, pp. 2091–2095 (2007)

[41] Cohen, D.W.: An Introduction to Hilbert Space and Quantum Logic. Springer, Berlin (1989)

[42] D'Hondt, E., Panangaden, P.: Quantum weakest preconditions. Math. Struct. Comput. Sci. **16**, 429–451 (2006)

[43] Ying, M.S.: Foundations of Quantum Programming. Morgan Kaufmann, Burlington (2016)

[44] Cartier, P.: A mad day's work: from Grothendieck to Connes and Kontsevich. The evolution of concepts of space and symmetry. J.: Bull. Am. Math. Soc. **38**, 389–408 (2001)

[45] Ivanov, P.A., Kyoseva, E.S., Vitanov, N.V.: Engineering of arbitrary U(N) transformations by quantum Householder reflections. Phys. Rev. A **74**, 022323 (2006)

[46] O'Donnell, R.: Analysis of Boolean Functions. Cambridge University Press, Cambridge (2014)

[47] Van Rijsbergen, K.: The Geometry of Information Retrieval. Cambridge University Press, Cambridge (2004)

[48] Barros, J., Toffano, Z., Meguebli, Y., Doan, B.-L.: Contextual query using bell tests. In: Atmanspacher, H., Haven, E., Kitto, K., Raine, D. (eds.) QI 2013. LNCS, vol. 8369, pp. 110–121. Springer, Berlin (2014)

[49] Wittek, P.: Quantum Machine Learning. What Quantum Computing Means to Data Mining. Academic Press, Elsevier, Amsterdam (2014)

[50] Haven, E., Khrennikov, A.: Quantum Soc. Sci. Cambridge University Press, Cambridge (2013)

[51] Busemeyer, J.R., Bruza, P.D.: Quantum Models of Cognition and Decision. Cambridge University Press, Cambridge (2012)

Zeno Toffano
CentraleSupélec, Laboratoire des Signaux et Systèmes-CNRS-(L2S-UMR8506)
Université Paris-Saclay
3 Rue Joliot Curie
91190 Gif-sur-Yvette
France
e-mail: zeno.toffano@centralesupelec.fr

Received: February 1, 2019.
Accepted: April 13, 2020.