Published in Journal of Logic and Computation, Vol. 18, pp. 97-121 (2008)

Vector Logic: a natural algebraic representation of the fundamental logical gates

Eduardo Mizraji Sección Biofísica, Facultad de Ciencias, Universidad de la República, C.C. 6695, 11000 Montevideo, Uruguay mizraj@fcien.edu.uy

ABSTRACT

Vector logic is a matrix-vector representation of the logical calculus inspired in neural network models. In this algebraic formalism, the truth values map on orthonormal Q-dimensional vectors, the monadic operations are represented by square matrices, and the dyadic operations produce rectangular matrices that act on the Kronecker product of the vector truth values. In this formalism, the theorems and tautologies of classical logic are demonstrated using the rules of matrix algebra. In the present work, we analyze a three-valued vector logic that adds to the "yes" and "no" vectors, a third "uncertain" vector that represents the truth value corresponding to undecidable propositions. Fuzziness is produced both via linear combinations of "yes" and "no" vectors, and by the supplementary dimension of the logical vector logic, the modalities "possibility" and "necessity" are simple square matrices instead of infinite recursive processes. Finally, we explore the application of this formalism to represent the complex valued operator \sqrt{NOT} , and the usefulness of vector logic to understand the powers and limitations of some reversible logical computations.

1. INTRODUCTION

The logical abilities integrate the repertoire of cognitive performances of the human brain, and the understanding of the nature of these abilities is of the first importance. Since the foundational investigations of George Boole [1], the searching for this understanding generates an unexpected field of convergence of biological, philosophical and technological contributions. Engineers are currently developing conceptual and technical instruments, in particular artificial neural networks, which originate in the intersection between philosophical logic and theoretical neuroscience (see [14], [15]).

During the last decades, linear algebra has become a basic formalism to analyze the properties of neural network models [2], [15], [18], [26], [27]. In this framework, matrices and vectors have been used to represent logical operations sustained by associative memories [19]. Some recent publications are exploring the potentialities of vectorial representations to approach different aspects of logic. In Dick [8], the author uses vectors and matrices to extend the logical formalisms to the complex domain; the work by Westphal and Hardy [29] actualize a geometrical approach (suggested by Leibniz) where vectors are used to construct representations for many logical rules, a formalism developed by these authors to explore the computational potentialities of optical logical gates.

Based in biologically inspired neural models, we developed some years ago a formalism named "vector logic". In this vector logic, truth values map on orthonormal Q-dimensional vectors, monadic operations generate square matrices, and dyadic operations produce rectangular matrices that act on the Kronecker product of the two truth values [20], [27]. Within this formalism, the theorems and tautologies of classical logic are demonstrated using the rules of matrix algebra. In the two-valued domain, this vector logic can very naturally generate a fuzzy logic closely related with Lukasiewicz and Kleene three-valued logics. This fuzzy logic appears when the matrix operators act over truth values represented by linear combinations of "yes" and "no" vectors [21], [22], [23]. Vector logic is an algebraic logic that introduces very naturally an operator approach, a fact that could confer to this construction an additional interest.

Until now, vector logic has been restricted to vector spaces of dimension two. Nevertheless, the potentialities and usefulness of larger dimensionalities remain to be explored. Consequently, the objective of the present article is to extend the vector logic representation in new directions. We begin with a brief mention to some historical facts related with the interactions between algebraic logic and neural modeling. After that, we introduce the notion of classificatory matrix, a device that establishes a correspondence between semantic vectors and truth-values. We describe the basic formalism, and we analyze some of the properties of two and three-dimensional vector logic, showing how this algebra can be used to represent the classical three-valued logics of Lukasiewicz and Kleene. We also show that the basic modal operators "possibility" and "necessity" are generated by completely different procedures in two and three-valued vector logics. Finally, we explore the extension of vector logic formalism to the complex domain and its usefulness to understand the powers and limitations of computations performed by reversible logical gates.

2. ALGEBRAIC LOGIC AND NEURAL MODELS

Nowadays, the mathematical analysis of the logic has reached a high level of conceptual and technical sophistication. This level is the consequence of a deep formalization program that has produced, together with many other important results, some of the highlights of modern computational theory. Clearly, the contributions of Gödel and Turing are between the most important works in the history of science. In another flank, several recent pragmatic contributions to parallel computation have begun to use the theory of the neural networks. As we are going to argue in what follows, the initial neural network theories were strongly based in the ideas and mathematics of George Boole and Charles S. Peirce. To exemplify the pragmatic achievements of these theories, we mention the contemporary design of adaptive programs able to implement the 'cognitive computations" of virtuous players in strategy games (eg. the backgammon).

These two types of contributions reflect, with different emphasis, two conceptual approaches. These approaches approximately map on the theoretical orientations called

"mathematical logic" an "algebraic logic". A deep analysis of the subtle relationships between mathematical logic and algebraic logic is in [12]. Using an extremely schematic description, we can say that one approach focus the logic as being the basis of the mathematics; on the contrary, in the other approach, the mathematics is considered as a source of formalisms able to model the logical operations. Here we will elude all philosophical discussions, but to describe the nature of these two different approaches, let us to quote *in extenso* a celebrated text of Jorge Luis Borges: "Coleridge observes that all men are born Aristotelians or Platonists. The latter feel that classes, orders, and genres are realities; the former, that they are generalizations. For the latter, language is nothing but an approximative set of symbols; for the former, it is the map of the universe. The Platonist knows that the universe is somehow a cosmos, an order; that order, for the Aristotelian, can be an error or a fiction of our partial knowledge. Across the latitudes and the epochs, the two immortal antagonists change their name and language: one is Parmenides, Plato, Spinoza, Kant, Francis Bradley; the other, Heraclitus, Aristotle, Locke, Hume, William James." [2].

In the present paper we adopt an "Aristotelian" approach, looking for simple but powerful representations for the logical functions using matrix operators based in neural models of associative memories. The introduction by Lukasiewicz of the parenthesis-free notation (or Polish notation) can be considered as a first step towards the introduction of an operator theory in logic. In this notation, a dyadic operation ω between propositions p and q, ie. p ω q, become expressed as Ωpq , where Ω can be interpreted as an operator acting on two variables p and q. A full operator theory for logical operations requires that the operators by themselves become algebraic objects susceptible of being operated between them. In this sense, let us mention an interesting historical point. George Boole was an expert and an innovator in the application of operator methods to the theory of differential and difference equations [7]. Nevertheless, his logical algebra was mainly an algebra concerning operations and not operators. Only few years after Boole's "Laws of Thought", Arthur Cayley published the foundational article of matrix theory. The application of Cayley's matrices to the logical theory was suggested by Peirce and performed by Inving Copi in the framework of the Peirce's algebraic theory of relations [5].

The algebra of the logic, rooted mainly in the mathematical constructions of Boole and Peirce, has been decisive for the creation, in the following century, of the first mathematical theories of neuronal networks. These theories assumed that formal neurons are connected in networks that exhibit large and versatile computing abilities. The first influential theory of neuronal networks, published by McCulloch and Pitts in 1943, establishes that any formal neuron in the network was a logical device that computed Boolean inputs. The crucial result of this theory was that any well formed formula (wff) of propositional calculus can be implemented by means of a network of formal neurons [17]. In addition, these authors show that networks having formal neurons with recursive connections became able to implement existential and universal quantifiers, hence creating the conditions to install in the networks a first order predicate calculus. In a sense, the neural model designated by McCulloch and Pitts seems to crystallize the objective enunciated by Boole in the first phrase of his book "An Investigation of the Laws of Thought": "The design of the following treatise is to investigate the fundamental laws of those operations of the mind by which reasoning is performed; to give expression to them in the symbolic language of a Calculus, and upon this foundation to establish the science of Logic and construct its method." [1].

In the theory of neural networks, Boolean nets were followed by matrix associative memories. Both approaches are the basis of the modern neural network theory. The matrix memories are distributed memories that store information in the structure of their numerical coefficients [15]. The matrix memories have versatile computing abilities, and can perform parallel processing of data coded in vectors. The matrix memories are naturally governed by the laws of matrix algebra and consequently, they are operators capable of acting over vectorial inputs; in addition, they are also capable of interacting with other matrix memories via standard matrix operations. Among these computing abilities, let us mention the capacity to build up logical gates [20] [22] [23].

Vector logic is the result of using these matrix logical gates to re-describe the basic and classical results of elementary logic. A point of interest is that Boole's algebraic logic can be embedded in the vector logic formalism [21]. In addition, the rules of matrix algebra produce that in the vector logic framework, the order of variables and operators involved in

a wff of the propositional calculus, is exactly the order emerged from Polish notation, a confirmation that Lukasiewicz's notation was a germ of an operator theory [20] [23], 1996).

Assuming that in the neural realm concepts map on vectors [25], we are going to introduce here some definitions.

Definition 2.1. Reduced conceptual space. It is a set of conceptual normal (ie. module = 1) vectors $RCS = \{c_1, \dots, c_n\}$ such that for each $c \in RCS$ only one complementary vectorial concept $c' \in RCS$ exists. If vector c represents the concept "book", c'represents "no-book". Here we have $c \perp c'$.

Definition 2.2. Extended conceptual space. It is a set of conceptual vectors $ECS = \{c_1, \dots, c_n\}$ such that for each $c \in ECS$ a complementary vectorial concept $c' \in ECS$ exists, but there exists also a vector $c'' \in ECS$. If vector c represents the concept "book" and c' represents "no-book", vector c'' represents a concept partially compatible with "book" and "no-book", eg. the oral version of a legend, or an electronic-book. We assume that $c \perp c'$, $c \perp c''$ and $c' \perp c''$.

Definition 2.3. **Classificatory Matrix**. It is a matrix F(concept) that assigns to each vectorial concept, a vectorial truth-value. In the case of a Reduced Conceptual Space, we define

 $F(c) = sc^{T} + nc'^{T}$, with $c, c' \in RCS$.

In the case of an Extended Conceptual Space, we have

 $F(c) = sc^{T} + nc'^{T} + hc''^{T}$, with $c, c', c'' \in ECS$.

The vectors s,n,h are othonormal Q-dimensional real vector that represent truth-values: s corresponds to "true" (or "yes"), n to "false" (or "not") and h to "uncertain".

These classificatory matrices are devices capable to decide if a normalized neural vector f represents a category providing the answers "yes", "not" or "uncertain". For instance, if F(c)f = s, then f is identical to categorical vector c. These matrices establish a natural link between cognitive categorization and logical systems based on vectorial truth-values.

3. DEFINITIONS IN BASIC LOGIC.

In this section we present the basic definitions that prepare us for the particular developments presented in the next sections.

Definition 3.1. Truth-values. We define as "truth-values" a set of m abstract objects t_1, \dots, t_m that allows one to assign an abstract (i.e. qualitative or quantitative) value to a given proposition. The set of truth-values

 $\boldsymbol{\tau}_{m} = \left\{ t_{1}, \cdots, t_{m} \right\}$

allows to define a m-valued logic. Particular cases are $\tau_2 = \{t, f\}$ (t = "true", f = "false") and $\tau_3 = \{t, f, u\}$ (t = "true", f = "false", u = "uncertain") that allow to define two-valued and three-valued logical systems, respectively.

Definition 3.2. Monadic operations. They are the applications of the form

$$Mon(m) : \tau_m \rightarrow \tau_m$$
,

and let $M_m = \{Mon(m)\}$.

Definition 3.3. Dyadic operations. They are functions of the form

 $Dyad(m) : \tau_m \times \tau_m \rightarrow \tau_m$,

being \times the Cartesian product.

Definition 3.4. Elemetary m-valued logic. It is the following triad

$$\mathrm{EL}_{\mathrm{m}} = \langle \boldsymbol{\tau}_{\mathrm{m}}, \mathbf{M}_{\mathrm{m}}, \mathbf{D}_{\mathrm{m}} \rangle$$

with $D_m \equiv \{Dyad(m)\}.$

Definition 3.5. Particular elementary m-valued logic. It is a triad where some particular subset of monadic and dyadic operations have been selected:

$$\text{PEL}_{m} = \left\langle \tau_{m}, \; (\text{Mon}(m)^{1}, \text{Mon}(m)^{2}, \cdots), \; (\text{Dyad}(m)^{1}, \text{Dyad}(m)^{2}, \cdots) \right\rangle \,.$$

For instance, the particular elementary logic

$$\text{PEL}_2 = \left\langle \{0,1\}, (-), (+,\times) \right\rangle$$

is the classical Boolean logic over the symbols 0 and 1, with the monadic complementation -0 = 1 and -1 = 0 together with the dyadic addition + (addition modulo 2 if symbols 0 and 1 are natural numbers) and the dyadic multiplication ×.

A vector logic is an algebraic model of elementary logic where the truth-values map on Q-dimensional vectors and the monadic and dyadic operations are executed by matrix operators. Now we define the basic notions.

Definition 3.6. m-dimensional Vector Logic. This vector logic is a systems of logic consisting in

1. A set of m Q-dimensional normal column vectors

$$\mathbf{V}_{\mathrm{m}} = \left\{ \mathbf{v}_{1}, \cdots, \mathbf{v}_{\mathrm{m}} \right\} , \ \mathbf{v}_{\mathrm{i}} \in \mathbb{R}^{Q \times 1},$$

2. A set of monadic matrices

$$M_{\mu}: V_m \rightarrow V_m$$
 , $M_{\mu} \in \mathbb{R}^{Q \times Q}$

3. A set of dyadic matrices

$$\mathbf{D}_{\delta}: \, \mathbf{V}_{\mathrm{m}} \otimes \mathbf{V}_{\mathrm{m}} \rightarrow \mathbf{V}_{\mathrm{m}} \quad , \quad \mathbf{D}_{\delta} \in \mathbf{R}^{\mathbf{Q} \times \mathbf{Q}^{2}} \, ,$$

where $V_m \otimes V_m$ is the tensor product of V_m . The elements of this tensor product are the Kronecker products of the vectors belonging to V_m : $(v_i \otimes v_j) \in V_m \otimes V_m$, $\forall v_i, v_j \in V_m$.

The Kronecker product is essential in our vector logic, and is defined as follows [11]. Given two matrices $A = [a_{ij}]_{m \times n}$ and $B = [b_{ij}]_{p \times q}$, the Kronecker product $A \otimes B$ is given by

$$A \otimes B = [a_{ij}B]_{(mp)\times(nq)}$$

Two basic properties are the following:

$$(P1) \ (A \otimes B)^{T} = A^{T} \otimes B^{T}$$

(P2) $(A \otimes B)(A' \otimes B') = (AA') \otimes (BB')$.

Property (P2) requires conformable matrices (or vectors). In the case of two r-dimensional column vectors a and c, and two r'-dimensional vectors b and d, we have

$$(\mathbf{a} \otimes \mathbf{b})^{\mathrm{T}} (\mathbf{c} \otimes \mathbf{d}) = (\mathbf{a}^{\mathrm{T}} \mathbf{c})(\mathbf{b}^{\mathrm{T}} \mathbf{d}) = \langle \mathbf{a}, \mathbf{c} \rangle \langle \mathbf{b}, \mathbf{d} \rangle,$$

an important result for the operation of the dyadic vector logic functions.

In the following, we are going to use these basic definitions to construct two- and threedimensional vector logics and to explore many interesting properties.

4. VECTOR LOGIC IN A TWO-DIMENSIONAL SUBSPACE.

Let us define a propositional calculus as a classification system that assigns a truth value to each proposition. In this case, the truth values are: true t (or "yes"), false f (or "no"). Hence, $\tau_2 = \{t, f\}$. This classical dyadic logic possesses 4 monadic operations and 16 dyadic operations. The establishment of a two-dimensional vector logic begins with the existence of a correspondence between the abstract truth-values t and f, and two Q-dimensional normalized column vectors: $t \mapsto s$ and $f \mapsto n$, $s, n \in \mathbb{R}^{Q \times 1}$, with $Q \ge 2$. Hence, $V_2 = \{s, n\}$.

We can consider that this kind of vector logic is the result of a classificatory matrix F(c) acting on a reduced conceptual space RCS (see Definition 2.1). In the evolution of cognition, this RCS is the initial fundamental scenario. A sequel of this fact is a well defined two-valued logic with uncertainty not included in the potential repertoire of truth-values. In a scalar representation of the emerged logic, with a set of truth-values $\tau_2 = \{t, f\}$, this is the end of the story. But inside the vector representation, the set of truth-values $V_2 = \{s, n\}$ and the associated matrix logical gates, allow to compute linear combinations of vectors s and n, and these linear combinations of vectors become a first natural representation for uncertain truth-value assignments.

4.1. BASIC OPERATORS.

The scalar product between Q-dimensional column vectors, $\mathbf{u}^T \mathbf{v} = \langle \mathbf{u}, \mathbf{v} \rangle$, is the basic operation behind the interesting properties displayed by vector logic. The orthonormality between vectors s and n implies that $\langle \mathbf{u}, \mathbf{v} \rangle = 1$ if $\mathbf{u} = \mathbf{v}$, and $\langle \mathbf{u}, \mathbf{v} \rangle = 0$ if $\mathbf{u} \neq \mathbf{v}$, $\mathbf{u}, \mathbf{v} \in V_2$

In this two-dimensional vector logic, the two basic monadic operators

$$M_{\mu}: V_2 \rightarrow V_2$$
, $M_{\mu} \in \mathbb{R}^{Q \times Q}$

are the identity and the negation, defined as follows:

M1) Identity. A logical identity up eq p corresponds with a logical matrix behaving as follows: $I_2u = u$, $u \in V_2$; the structure of this matrix is

 $I_2 = ss^{T} + nn^{T};$

notice that Is = s $\langle s, s \rangle$ + n $\langle n, s \rangle$ = s and In = n, due to the orthogonality of s respect to n.

M2) Negation. The classical negation $\neg p$ corresponds with the matrix operation $N_2 u$, $u \in V_2$, being

$$N_2 = ns^T + sn^T$$
.

Hence $N_2 s = n$ and $N_2 n = s$. Notice that involutory character of the logical negation, $\neg (\neg p) eq p$, corresponds with the fact that $(N_2)^2 = I_2$ (but remark that the vector logic identity matrix is not generally an identity matrix in the sense of matrix algebra, except in very special cases).

Between the 16 two-valued dyadic operators

 $D_{\delta}: V_2 \otimes V_2 \to V_2 \quad , \quad D_{\delta} \in \mathbb{R}^{Q \times Q^2} \,,$

we are going to begin with the conjunction and the disjunction.

D1) Conjunction. The classical conjunction between two propositions $p \land q$ is represented by a matrix acting over two vector truth-values: $C_2(u \otimes v)$, $u, v \in V_2$. The matrix C_2 that reproduces the features of the classical conjunction truth-table is:

$$C_{2} = s(s \otimes s)^{T} + n(s \otimes n)^{T} + n(n \otimes s)^{T} + n(n \otimes n)^{T}.$$

This matrix behaves as follows: $C_2(s \otimes s) = s$; $C_2(s \otimes n) = C_2(n \otimes s) = C_2(n \otimes n) = n$.

D2) Disjunction. The classical disjunction $p \lor q$ corresponds with the matrix

$$D_2 = s(s \otimes s)^T + s(s \otimes n)^T + s(n \otimes s)^T + n(n \otimes n)^T,$$

being $D_2(s \otimes s) = D_2(s \otimes n) = D_2(n \otimes s) = s$ and $D_2(n \otimes n) = n$.

In the classical two-valued logic, the conjunction and the disjunction operations satisfy the De Morgan Law: $p \land q = q \neg (\neg p \lor \neg q)$ (and also is true the converse: $p \lor q = q \neg (\neg p \land \neg q)$). In the two-dimensional vector logic this Law is also verified

$$C_2(\mathbf{u} \otimes \mathbf{v}) = N_2 D_2(N_2 \mathbf{u} \otimes N_2 \mathbf{v}) ,$$

but the Kronecker product allows the following factorization:

$$C_2(\mathbf{u} \otimes \mathbf{v}) = N_2 D_2(N_2 \otimes N_2)(\mathbf{u} \otimes \mathbf{v}).$$

In this case we can prove directly, using the previous matrix definitions that in the two – dimensional vector logic, that the De Morgan Law is not only a law concerning operations, but also a law involving operators:

$$C_2 = N_2 D_2 (N_2 \otimes N_2).$$

A detailed study of this logic as an operator logic has been published in [23].

These matrix expressions for conjunction and disjunction allow us to define the matrices $S_2 = N_2C_2$ and $P_2 = N_2D_2$, corresponding to the Sheffer (or NAND) and the Peirce (or NOR) gates, respectively.

D3) Implication. The "material" implication can be defined in classical logic using the expression $p \Rightarrow q eq \neg p \lor q$. In vector logic, the translation of this equivalence leads to a matrix L₂ that represents vector logic "material" implication:

$$L_2 = D_2(N_2 \otimes I_2) .$$

The explicit expression for this implication is

$$L_2 = s(s \otimes s)^{T} + n(s \otimes n)^{T} + s(n \otimes s)^{T} + n(n \otimes n)^{T},$$

that satisfies the classical properties of classical implication:

 $L_2(s \otimes s) = L_2(n \otimes s) = L_2(n \otimes n) = s$ and $L_2(s \otimes n) = n$.

D4) Symmetric operators: Equivalence and Exclusive-Or. The equivalence $p \equiv q$ corresponds in vector logic with the following matrix:

$$E_{2} = s(s \otimes s)^{T} + n(s \otimes n)^{T} + n(n \otimes s)^{T} + s(n \otimes n)^{T}.$$

Hence, $E_2(s \otimes s) = E_2(n \otimes n) = s$ and $E_2(s \otimes n) = E_2(n \otimes s) = n$.

The Exclusive-Or is the negation of the equivalence, $\neg(p \equiv q)$, and is represented by the matrix $X_2 = N_2E_2$ which explicit expression is

$$X_2 = n(s \otimes s)^T + s(s \otimes n)^T + s(n \otimes s)^T + n(n \otimes n)^T;$$

consequently $X_2(s \otimes s) = X_2(n \otimes n) = n$ and $X_2(s \otimes n) = X_2(n \otimes s) = s$.

4.2 MANY-VALUED TWO-DIMENSIONAL LOGIC.

In the case of two-valued vector logic, uncertainties in the truth values are introduced via vectors $f = \delta s + \gamma n$, with $\delta, \gamma \in [0,1]$, $\delta + \gamma = 1$. These vectors are processed by "Boolean", non many-valued, matrix operators that produce linear combinations of s and n as outputs. The many-valued character of this logic appears in the structure of the coefficients of these linear combinations [20] [23]. In particular, this situation generates a class of probabilistic logic, which maps on the scalar coefficients that weights vector s. For the case $u = \alpha s + \beta n$ and $v = \alpha' s + \beta' n$ the many-valued (or "fuzzy") logic obtained from any two-valued matrix G is given by its projection over vector s :

 $Val(scalars) = s^T G(vectors)$.

Thus, we have the following results:

 $NOT(\alpha) = s^{T}N_{2} u = 1 - \alpha$ $OR(\alpha, \alpha') = s^{T}D_{2} (u \otimes v) = \alpha + \alpha' - \alpha \alpha'$ $AND(\alpha, \alpha') = s^{T}C_{2} (u \otimes v) = \alpha \alpha'$ $IMPL(\alpha, \alpha') = s^{T}L_{2} (u \otimes v) = 1 - \alpha(1 - \alpha')$ $XOR(\alpha, \alpha') = s^{T}X_{2} (u \otimes v) = \alpha + \alpha' - 2\alpha \alpha'$

With these equations we can further define the corresponding negations:

NOR
$$(\alpha, \alpha') = 1 - OR(\alpha, \alpha')$$

NAND $(\alpha, \alpha') = 1 - AND(\alpha, \alpha')$
EQUI $(\alpha, \alpha') = 1 - XOR(\alpha, \alpha')$

It is well known that the searching of a truth-functional representation for the modal operations lead Jan Lukasiewicz to construct, around 1920, a three-valued logic [16] (see Section 5.6). The scalar projections of our two-valued vector logic generate an infinite-valued scalar logic that allows an operational construction of the logical modalities "Possibility" and "Necessity" (the full development of this approach was published in [22]). Let us assume the existence of an infinite set with binary evaluations concerning a proposition P : $\{p_1, p_2, ..., p_n, ...\}$, with $p_i \in \{t, f\}$. Then, the proposition "P is possible", $\exists P$, can be represented by

 $\exists P = p_1 \lor p_2 \lor \ldots \lor p_n \lor \ldots ,$

or by the recursive process

 $\boldsymbol{Q}_{n+1}\!=\!\boldsymbol{p}_{n+1}\,\vee\,\boldsymbol{Q}_n$, $n\!=\!1,\,2,\,...$

with $Q_1 = p_1$. In this process, the possibility $\exists P$ is the limit of Q_n for $n \to \infty$.

Let us mention in passing, that the McCulloch-Pitts neural model [17] implement this recursive process using binary threshold neurons able of computing the AND and the OR. In the matrix recursion that we are going to describe, the place of threshold binary neurons is filled by logical matrix with the structure of associative memories.

The matrix version of this process is as follows:

$$\exists_{n+1}[u] = D_2(u_{n+1} \otimes \exists_n[u])$$

with $\exists_1[u] = u_1$, being in general $u_i = \alpha_i s + (1 - \alpha_i)n$, $\alpha_i \in [0,1]$. The scalar projection of $\exists [u] = \lim_{n \to \infty} \exists_n[u]$ is given by the infinite product

$$s^{T} \exists [u] = 1 - (1 - \alpha_1)(1 - \alpha_2)(1 - \alpha_3) \dots;$$

this product can be condensed in a compact geometrical mean expression:

$$s^{\mathrm{T}} \exists [u] = \lim_{n \to \infty} [1 - (1 - \alpha)^n]$$

Hence

$$s^{\mathrm{T}} \exists [u] = \begin{cases} 0 \text{ iff } \alpha = 0 \\ 1 \text{ iff } \alpha \neq 0 \end{cases}$$
.

The proposition "P is necessary", \forall P, can be represented using a concatenated conjunction

$$\forall \mathbf{P} = \mathbf{p}_1 \land \mathbf{p}_2 \land \dots \land \mathbf{p}_n \land \dots,$$

or by the limit for $n \to \infty$ of the recursive process

$$R_{n+1} = p_{n+1} \wedge R_n$$
, $n = 1, 2, ...$

with $R_1 = p_1$. The matrix version is:

$$\forall_{n+1}[u] = C_2(u_{n+1} \otimes \forall_n[u])$$

where $\forall_1[u] = u_1$ $(u_i = \alpha_i s + (1 - \alpha_i)n$, $\alpha_i \in [0,1])$. The scalar projection of $\forall [u] = \lim_{n \to \infty} \forall_n[u]$ is the infinite product

$$s^T \forall [u] = \alpha_1 \alpha_2 \alpha_3 \dots,$$

and the limit geometrical mean is

$$s^T \forall [u] = \lim_{n \to \infty} \alpha^n$$
.

This limit produces the classical scalar expression for the necessity operator

$$s^{\mathrm{T}} \forall [\mathbf{u}] = \begin{cases} 0 \text{ iff } \alpha \neq 1 \\ 1 \text{ iff } \alpha = 1 \end{cases}$$
.

We recall that these modal operators are linked by the well known "Aristotle's theorem":

 $\forall [u] = \neg \exists [\neg u]$. It is easy to prove that, as a consequence of De Morgan Law, both the matrix equations and the scalar projections satisfy this theorem [22].

5. THREE-VALUED VECTOR LOGIC.

Let us consider that a propositional calculus results from a classification system that assigns a truth value to each proposition. In the case of three-valued logic, the truth values can adopt three values: true t (or "yes"), false f (or "no"), and uncertain u (or "undecidable"). The calculus allow us to define the basic two classes of functions: the monadic functions $M_{\mu}: \tau_3 \rightarrow \tau_3$ and the dyadic functions $D_{\delta}: \tau_3 \times \tau_3 \rightarrow \tau_3$, with $\tau_3 = \{t, f, u\}$.

Neural systems are capable to create new concepts based on previous concepts. In the evolution of cognition, we can consider that the first conceptual constructions emerged directly from sensorial experiences, forcing the search of crisp conceptual partitions (hence to a RCS and a two-valued logic). In a further step, the experience with logical uncertainties (as described in section 4.2) could naturally conduce to enlarge the conceptual space generating a ECS (Definition 2.2) that include non-complementary concepts c'', a fact that allows to introduce a third truth-valued in the logical evaluations.

The corresponding three-valued vector logic results from the mapping between the three abstract truth values $\tau_3 = \{t, f, u\}$ on a Q-dimensional vector space containing orthonormal column vectors. We have $t \mapsto s; f \mapsto n; u \mapsto h$, with $s, n, h \in V_3$ (V_3 a Q-dimensional vector space, $Q \ge 3$). Similarly with the two-valued vector logic, for two vectors $u, v \in \{s, n, h\}$ the variety of inner products between u and v is $\langle u, v \rangle = 1$ if u = v, and $\langle u, v \rangle = 0$ if $u \neq v$.

In general, a m-valued logic possesses $m^{(m^n)}$ n-adic operators. Consequently, for a three-valued logic we have 27 monadic operators and 19,683 dyadic operators (the classical two-valued logic displays 4 monadic and 16 dyadic operators). These figures indicate that inside a three-valued logic the searching of relevant operators must be oriented in some

way. We begin with the construction of the matrix operators corresponding to the logical identity, the negation (NOT), the conjunction (AND) and the disjunction (OR) as defined in the Lukasiewicz's three-valued logic. A variety of many-valued logics is described in references [14] and [28].

As in the case of the two-valued logic, in the classical three-valued logic the basic gates are described using truth tables. In the three-valued vector logic the tables became matrices.

5.1 MONADIC OPERATORS.

The two central monadic operators are the logical identity and the negation, defines in what follows.

M1) Identity

$$I_3 = ss^T + nn^T + hh^T;$$

notice that $I_3 s = s$, $I_3 n = n$ and $I_3 h = h$.

M2) Negation

$$N_3 = ns^T + sn^T + hh^T$$
;
hence $N_3 s = n$, $N_3 n = s$ and $N_3 h = h$, being
 $(N_3)^2 = I_3$.

5.2 BASIC DYADIC OPERATORS.

We adopt for these dyadic operators the general polynomial expression. Let Op represent any dyadic operator defined by

$$\begin{split} & Op = a_1 \left(s \otimes s \right)^T + a_2 \left(s \otimes n \right)^T + a_3 (n \otimes s)^T + a_4 \left(n \otimes n \right)^T + \\ & b_1 \left(s \otimes h \right)^T + b_2 (h \otimes s)^T + b_3 \left(n \otimes h \right)^T + b_4 \left(h \otimes n \right)^T + b_5 \left(h \otimes h \right)^T , \end{split}$$

with $a_i, b_i \in \{s, n, h\}$. Then we put

$$Op = R(a_1 a_2 a_3 a_4) + I(b_1 b_2 b_3 b_4 b_5)$$

where the coefficients in R(...) depend only on "classical" truth values s and n , and coefficients in I(...) involve terms including the uncertain vector h. The terms represented by R describe the matrices corresponding to the two-valued vector logic, and the terms represented by I indicate the effect of the additional dimension h in the logical vector subspace.

We proceed now to describe the conjunction and disjunction three-valued matrices.

D1) Conjunction

$$\begin{split} &C_3 = s \left(s \otimes s \right)^T + n \left(s \otimes n \right)^T + n \left(n \otimes s \right)^T + n \left(n \otimes n \right)^T + \\ &h \left(s \otimes h \right)^T + h \left(h \otimes s \right)^T + n \left(n \otimes h \right)^T + n \left(h \otimes n \right)^T + h \left(h \otimes h \right)^T . \end{split}$$

This matrix satisfies the definition of the three-valued Lukasiewicz's conjunction: $C_3(s \otimes s) = s$; $C_3(s \otimes n) = C_3(n \otimes s) = C_3(n \otimes n) = C_3(n \otimes h) = C_3(h \otimes n) = n$; and $C_3(s \otimes h) = C_3(h \otimes s) = C_3(h \otimes h) = h$. Consequently

 $C_3 = R(snnn) + I(hhnnh)$.

D2) Disjunction

$$\begin{split} &D_3 = s(s \otimes s)^T + s(s \otimes n)^T + s(n \otimes s)^T + n(n \otimes n)^T + \\ &s(s \otimes h)^T + s(h \otimes s)^T + h(n \otimes h)^T + h(h \otimes n)^T + h(h \otimes h)^T , \\ &\text{with } D_3(s \otimes s) = D_3(s \otimes n) = D_3(n \otimes s) = D_3(s \otimes h) = D_3(h \otimes s) = s ; \ D_3(n \otimes n) = n ; \\ &\text{and } D_3(n \otimes h) = D_3(h \otimes n) = D_3(h \otimes h) = h . \ \text{This three-valued disjunction can be} \\ &\text{expressed as} \end{split}$$

 $D_3 = R(sssn) + I(sshhh)$.

Notice that, due to the fact that the term $R(a_1 a_2 a_3 a_4)$ represents the classic vector logic operator, the previous monadic and dyadic operators can be written as follows:

$$\begin{split} I_3 &= I_2 + hh^T ; \\ N_3 &= N_2 + hh^T ; \\ C_3 &= C_2 + h (s \otimes h)^T + h (h \otimes s)^T + n (n \otimes h)^T + n (h \otimes n)^T + h (h \otimes h)^T \\ D_3 &= D_2 + s (s \otimes h)^T + s (h \otimes s)^T + h (n \otimes h)^T + h (h \otimes n)^T + h (h \otimes h)^T . \end{split}$$

In the two valued vector logic, the operators C_2 and D_2 satisfy De Morgan Laws: $C_2 = N_2 D_2 (N_2 \otimes N_2)$ and $D_2 = N_2 C_2 (N_2 \otimes N_2)$. In the present case, is easy to prove that three-valued C_3 and D_3 also satisfy De Morgan Laws:

$$\begin{split} &C_3 = N_3 D_3 (N_3 \otimes N_3) \ , \\ &D_3 = N_3 C_3 (N_3 \otimes N_3) \ . \end{split}$$

We also note that the matrices $S_3 = N_3C_3$ and $P_3 = N_3D_3$, corresponding to the Sheffer (or NAND) and the Peirce (or NOR) gates, respectively, are:

$$S_3 = R(nsss) + I(hhssh)$$
,
 $P_3 = R(nnns) + I(nnhhh)$.

5.3. LOOKING FOR IMPLICATION

Lukasiewicz and Kleene create two different three-valued implications [14], [28]. Lukasiewicz's implication is a function of two variables, IL(x,y) with $x, y \in \{t, f, u\}$, defined as follows: IL(t,t) = IL(f,t) = IL(f,f) = IL(u,t) = IL(t,u) = IL(u,u) = t : IL(t,f) = f; IL(t,u) = IL(u,f) = u. The three-valued implication proposed by Kleene focus the semantic meaning of the truth value u, generating a slight but fundamental difference respect to Lukasiewicz's implication. Kleene's implication can be defined by a function IK(x,y) as follows: IK(t,t) = IK(f,t) = IK(f,f) = IK(u,t) = IK(f,u) = t : IK(t,f) = f ; IK(t,u) = IK(u,f) = IK(u,u) = u.

If we try to construct an implication inside the three-valued vector logic, we discover that using the classical definition $p \rightarrow q$ eq $\neg p \lor q$, we obtain a matrix implication that behaves like Kleene's implication. The three-valued matrix version for the classical definition of the implication is:

 $L_3(u\otimes v) = D_3(N_3\,u\otimes v) \ , \ u,v\in\{s,n,h\} \, .$

Using the properties of the Kronecker product, we have

$$L_3(u \otimes v) = D_3(N_3 \otimes I_3)(u \otimes v)$$

and we can define the matrix implication L_3 as:

$$L_3 = D_3(N_3 \otimes I_3).$$

Computing the outcomes from this operator, we can easily determine its structure. In our compact notation, we have

$$L_3 = R(snss) + I(hsshh),$$

indicating that here we have a matrix version of Kleene's implication.

5.4 EQUIVALENCE AND EXCLUSIVE-OR

A natural departure point for constructing a three-valued vector logic equivalence is to adopt the classical equivalence $p \equiv q$ eq $(p \rightarrow q) \land (q \rightarrow p)$. We begin using as vector implication the operator L₃ defined previously, and we construct the vectorial equivalence EQU(u,v) as

 $EQU(u,v) = C_3[L_3(u \otimes v) \otimes L_3(v \otimes u)] , u, v \in \{s,n,h\}.$

Kronecker factorization produces:

 $EQU(u, v) = C_3 (L_3 \otimes L_3)(u \otimes v \otimes v \otimes u).$

Remark that $C_3(L_3 \otimes L_3)$ is a matrix of order $Q \times Q^4$. This operator can be computed directly, and the set of outputs obtained can be used to select a $Q \times Q^2$ matrix operator

capable to generate the same outputs (a projection on a lower dimension of the mapping produced by EQU acting on u and v). The result of this projection is a correspondence $EQU(u,v) \mapsto E_3(u \otimes v) ,$

where E_3 is an equivalence matrix given by the expression

 $E_3 = R(snns) + I(hhhhh)$.

The exclusive-or (or XOR) matrix X_3 can be immediately obtained using the expression $X_3 = N_3E_3$. The result is

 $X_3 = R(nssn) + I(hhhhh)$.

5.5 POLISH VECTOR LOGIC

A matrix for the Lukasiewicz's implication L_P (P for "Polish"), can be directly written as

$$L_P = R(snss) + I(hsshs)$$
.

Using this matrix Lukasiewicz's implication, L_P , we can construct Lukasiewicz versions for the disjunction D_P and the conjunction C_P . After that, we can define the equivalence E_P , and the exclusive-or X_P . We proceed calculating the detailed expressions for the following equations:

$$\begin{split} D_{P} &= L_{P}(N_{3} \otimes I_{3}) \\ C_{P} &= N_{3}D_{P}(N_{3} \otimes N_{3}) \\ EQU_{P}(u,v) &= C_{P}(L_{P} \otimes L_{P})(u \otimes v \otimes v \otimes u) \\ EQU_{P}(u,v) &\mapsto E_{P}(u \otimes v) \\ X_{P} &= N_{3}E_{P}. \end{split}$$

(In order to check the consistency inside this "polish" vector logic, note that $D_p(N_3 \otimes I_3) = L_P(N_3 \otimes I_3)(N_3 \otimes I_3) = L_P(N_3^2 \otimes I_3^2) = L_P).$

The results of these operations are

 $D_{P} = R(sssn) + I(sshhs)$ $C_{P} = R(snnn) + I(hhnns)$ $E_{P} = R(.snns) + I(hhhhs)$

 $X_{p} = R(nssn) + I(hhhn).$

5.6 MODALITIES IN THREE-VALUED VECTOR LOGIC.

We showed in Section 4.2 that an infinite-valued logic can be the result of linear combinations of two truth values processed by a dyadic (ie. Booelean) matrix, with the modal operators "possibility" and "necessity" arising naturally as the limits of recursive processes.

In this section we analyze how we can represent the logical modalities "Possibility" and "Necessity" in the framework of three-valued vector logic. It is an important historical fact that Lukasiewicz three-valued logic was directly derived from investigations concerning modal propositions, and from the searching of truth-functional representations for the operators "possibility" and "necessity" [16]. Let us quote some passages of his important 1930 article (in the following quotation, the expression "matrix method" refers to the characterization of logical operations using mathematical functions defined by "truth-tables"). Lukasiewicz wrote: "When I became aware of the incompatibility of traditional theorems of modal propositions in 1920, I was in the process of establishing a normal bivalent propositional calculus based on the matrix method . At that time I was convinced that it was possible to demonstrate all the thesis of the ordinary propositional calculus assuming that propositional variables could take on only two values, "0" (false), and "1" (true). (...) Our whole system of logic is based on the law of bivalence, even though it has

been fiercely disputed since ancient times. Aristotle knew this law, but he questioned its validity as it referred to future contingent propositions. The law of bivalence was flatly rejected by the Epicureans. Chrysippus and the Stoics were the first ones to develop it fully and incorporate it as a principle of their dialectic, the equivalent of modern day propositional calculus. The arguments regarding the law of bivalence have metaphysical overtones: its supporters are resolute determinists; whereas its opponents generally have an indeterministic Weltanschauung. Thus, we are once again in the area of concepts of possibility and necessity." [16].

In his article, Lukasiewicz demonstrates that the operators possibility (pos) and necessity (nec) acting on the dyadic truth-valued set $\tau_2 = \{t, f\}$, degenerate into identical expressions:

pos(t) = t, pos(f) = f; nec(t) = t, nec(f) = f.

On the contrary, inside a three-valued set $\tau_3 = \{t, f, u\}$, Lukasiewicz [16] shows how the mathematical definitions for the modal operators pos and nec split into meaningful expressions:

pos(t) = pos(u) = t, pos(f) = f; nec(t) = t, nec(u) = nec(f) = f.

Inside the three-valued vector logic, based on the set {s,n,h}, we can define the matrix operators possibility, Pos, and necessity, Nec, directly from the previous truth-functional Lukasiewicz's definitions. We obtain the following matrices:

 $Pos = I_2 + sh^T ,$ $Nec = I_2 + nh^T ,$

where $I_2 = ss^T + nn^T$, is the two-valued logical identity. Note that the operator Pos satisfies the condition of a modal possibility:

Pos s = Pos h = s, Pos n = n.

The operator Nec produces

Nec s = s, Nec h = Nec n = n.

These results correspond with the truth-tables established in Lukasiewicz [16] from his three-valued logic. It is interesting to note that these matrix operators verify the Aristotle's theorem:

,

```
Nec = N_3 Pos N_3,
```

where N_3 is the three-valued matrix negation.

We want to emphasize that with this three-valued vector logic formalism, the modal operators "possibility" and "necessity" are simple monadic operators represented by square matrices. This is an important difference with two-valued vector logic because in the enlarged logical space recursive processes are no longer necessary.

It is interesting to analyze Lewis' "strict implication", a kind of implication that connects classical "material" implication with modalities; this operation is executed by a matrix Lew [22]:

 $Lew = Nec L_3$

This matrix strict implication only produces outputs inside the set $\{s,n\}$. An alternative Lewis matrix can be constructed using Lukasiewicz (Polish) matrix implication:

 $Lew_P = Nec L_P$.

Using our compact notation, we have

Lew = R(snss) + I(nssnn)Lew P = R(snss) + I(nssns)

The sole difference between Lew and Lew_P is the response respect to input $h \otimes h$: Lew($h \otimes h$) = n, but Lew_P($h \otimes h$) = s.

The logical matrix operators transform many fundamental theorems of propositional calculus in natural algebraic operations: This was previously illustrated for the De Morgan Law. Other interesting case is given by following the logical equivalence

$$p \rightarrow q eq \neg q \rightarrow \neg p$$
.

This important property of logical implication, can be proved inside the two-valued, the three-valued and the three-valued Polish vector logics using the normal properties of matrix operations and Kronecker products. Let $\rho = 2$, 3 and P, and let $\lambda = 2$ if $\rho = 2$ and $\lambda = 3$ if $\rho = 3$ or P. Then, the proofs are unified using the following equations:

$$L_{\rho}(u \otimes v) = D_{\rho}(N_{\lambda} \otimes I_{\lambda})(u \otimes v) = D_{\rho}(N_{\lambda}u \otimes I_{\lambda}v) .$$

This equality assumes that $L_{\rho} = D_{\rho}(N_{\lambda} \otimes I_{\lambda})$. Considering that $D_{\rho}(a \otimes b) = D_{\rho}(b \otimes a)$ (for $a, b \in \{s, n, h\}$) and $I_{\lambda} = N_{\lambda}N_{\lambda}$, we have:

$$\begin{split} &D_{\rho}(N_{\lambda}u\otimes I_{\lambda}v) = D_{\rho}(I_{\lambda}v\otimes N_{\lambda}u) = \\ &= D_{\rho}(N_{\lambda}N_{\lambda}v\otimes N_{\lambda}u) = D_{\rho}(N_{\lambda}\otimes I_{\lambda})(N_{\lambda}v\otimes N_{\lambda}u) \,. \end{split}$$

Consequently,

 $L_{\rho}(u \otimes v) = L_{\rho}(N_{\lambda}v \otimes N_{\lambda}u).$

6. EXTENSIONS

The original motivation for translating the formalism of elementary logic into the matrix format of vector logic, were neural network theories. Nevertheless, the emerging formalism is interesting in itself, and allows to state with a novel approach problems out of the neural domain. As an example, we recently employed the vector logic formalism to analyze the emergence of complex dynamics in a class of one-dimensional cellular automata [24]. Here we are going to translate into this formalism, some points concerning the use of complex numbers in logic, and the matrix representations of the Fredkin gate.

6.1 VECTOR LOGIC AND COMPLEX OPERATORS.

Recently, different motivations lead to propose algebraic logics that use complex numbers. The work by Dick [8] shows the construction of a fuzzy logic in the complex domain that becomes capable to introduce contextual conditions to improve inferential rules. The theory of quantum computation naturally leads to the extension of logical theory, including reversible logical gates [10] and vector variables in the complex domain [9].

Nevertheless, considering the intrinsic formal interest of the extension of the logic theory to the complex domain, in the present section we are going to analyze an important operator coming from the theory of quantum gates. This "quantum" operator is the square root of NOT, symbolized by $\sqrt{\text{NOT}}$, and represents one of the pivots of the quantum computing theory [6], [13]. Here, we are not directly concerned with quantum computation, and our interest is to explore a gate (with a specific usefulness for the quantum computation theory) that allows extending the formalism to the complex domain. It seems clear that $\sqrt{\text{NOT}}$ is a symbolic multidimensional version of the imaginary number $\sqrt{-1}$, and that quantum physics allows a concrete implementation of this formal fact.

a) The square root of $\,\mathrm{N}_2^{}$.

We begin looking for an operator $\sqrt{N_2}~$. Let

$$(\sqrt{N_2})s = \alpha s + \beta n$$
,
 $(\sqrt{N_2})n = \alpha' s + \beta' n$,
with $\alpha + \beta = \alpha' + \beta' = 1$.

The fact that

$$(\sqrt{N_2}) \left[(\sqrt{N_2}) s \right] = N_2 s = n ,$$

$$(\sqrt{N_2}) \left[(\sqrt{N_2}) n \right] = N_2 n = s ,$$

imposes the equations

$$\begin{cases} \alpha^{2} + \beta \alpha' = 0\\ \alpha \beta + \beta \beta' = 1 \end{cases}$$
$$\begin{cases} \alpha' \alpha + \beta' \alpha' = 1\\ \alpha' \beta + \beta'^{2} = 0 \end{cases}$$

We can also put

$$\begin{bmatrix} \alpha & \alpha' \\ \beta & \beta' \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$\begin{bmatrix} \alpha & \alpha' \\ \beta & \beta' \end{bmatrix} \begin{bmatrix} \alpha' \\ \beta' \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

It is easy to prove from these equations that the symmetry conditions $\alpha = \beta'$ and $\beta = \alpha'$ must hold; hence

$$\label{eq:alpha} \begin{split} \alpha^2 + \beta^2 &= 0 \,. \\ Assuming \\ \alpha &= u + i v \quad ; \quad \beta = u' + i v' \quad (\,i = \sqrt{-1}\,), \\ we \ obtain \ u &= u' \ and \ v &= -v' \ , and \\ \alpha &= u + i v \,, \\ \beta &= u - i v \,. \end{split}$$

Finally, being $\alpha + \beta = 1$, we have

$$\mathbf{u} = \mathbf{v} = \frac{1}{2} \ .$$

Consequently,

$$\begin{aligned} \alpha &= \frac{1}{2}(1+i) \ ; \ \beta &= \frac{1}{2}(1-i) \\ \alpha' &= \beta \ ; \ \beta' &= \alpha \ , \end{aligned}$$

and

$$(\sqrt{N_2})s = \frac{1}{2}(1+i)s + \frac{1}{2}(1-i)n,$$

 $(\sqrt{N_2})n = \frac{1}{2}(1-i)s + \frac{1}{2}(1+i)n.$

We can see here that the operator $\sqrt{N_2}$ produces an output formally similar to a fuzzy one (see Section 4.2), but with complex coefficients.

Let us define the matrix

$$\mathbf{A} = \begin{bmatrix} \alpha & \alpha' \\ \beta & \beta' \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(1+i) & \frac{1}{2}(1-i) \\ \frac{1}{2}(1-i) & \frac{1}{2}(1+i) \end{bmatrix}.$$

In what follows we are going to apply the well known property of matrix calculus $Ma_1 = b_1$ and $Ma_2 = b_2 \implies M[a_1a_2] = [b_1b_2]$

Using the appropriate vales for matrix M components) we obtain

$$\begin{bmatrix} \frac{1}{2}(1+i) & \frac{1}{2}(1-i) \\ \frac{1}{2}(1-i) & \frac{1}{2}(1+i) \end{bmatrix}^2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Defining

$$\mathbf{P} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

the previous equation becomes

$$A^2 = P .$$

Defining U = [s n], we obtain

$$(\sqrt{N_2})U = UA$$

and if vectors s and n are linearly independent, we have an exact solution of the form

$$\sqrt{N_2} = UAU^+$$
,

where U^+ is the right Moore-Penrose pseudo-inverse

$$\mathbf{U}^{+} = (\mathbf{U}^{\mathrm{T}}\mathbf{U})^{-1}\mathbf{U}^{\mathrm{T}}$$

Note that second member of the matrix equation $A^2 = P$ is the two-dimensional NOT used in the theory of quantum gates (hence A is in a sense the minimal matrix square root of NOT), but the formalism that produces the complex parameters α and β depends on Qdimensional vectors s and n included in matrix U.

b) The square root of N_3 .

In the framework of the three-valued vector logic, the search of the operator \sqrt{NOT} can be stated as follows: we look for a matrix operator that satisfies

$$(\sqrt{N_3})^2 = N_3$$
 ,

This operator $\sqrt{N_3}$, must be based in the extended set $\{s,n,h\}$. Let us assume that this operator satisfies the equations

$$(\sqrt{N_3})s = \alpha s + \beta n + \gamma h$$
$$(\sqrt{N_3})n = \alpha' s + \beta' n + \gamma' h$$
$$(\sqrt{N_3})h = \alpha'' s + \beta'' n + \gamma'' h,$$

with $\alpha+\beta+\gamma=\alpha{'+}\beta{'+}\gamma{'=}\alpha{''+}\beta{''+}\gamma{''=}1$. The consequence is

$$(\sqrt{N_3})[(\sqrt{N_3})s] = N_3 s = n$$
$$(\sqrt{N_3})[(\sqrt{N_3})n] = N_3 n = s$$
$$(\sqrt{N_3})[(\sqrt{N_3})h] = N_3 h = h.$$

Reproducing the procedure followed in the two-valued case, we obtain the matrix equation

$$B^2 = R,$$

with

$$\mathbf{B} = \begin{bmatrix} \alpha & \alpha' & \alpha'' \\ \beta & \beta' & \beta'' \\ \gamma & \gamma' & \gamma'' \end{bmatrix} \text{ and } \mathbf{R} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

A solution for this matrix equation is

$$\mathbf{B} = \begin{bmatrix} \frac{1}{2}(1+i) & \frac{1}{2}(1-i) & 0\\ \frac{1}{2}(1-i) & \frac{1}{2}(1+i) & 0\\ 0 & 0 & 1 \end{bmatrix}.$$

This solution generates the following complex "fuzzy" representation:

$$(\sqrt{N_3})s = \frac{1}{2}(1+i)s + \frac{1}{2}(1-i)n$$

 $(\sqrt{N_3})n = \frac{1}{2}(1-i)s + \frac{1}{2}(1+i)n$
 $(\sqrt{N_3})h = h$.

Now we can calculate the matrix $\sqrt{N_3}$. Remark that

$$(\sqrt{N_3})s = \alpha s + \beta n + \gamma h = \begin{bmatrix} s & n & h \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}.$$

Using the analogous expressions for $(\sqrt{N_3})n$ and $(\sqrt{N_3})h$, and defining the matrix V = [s n h], we have

$$(\sqrt{N_3})V = VB$$

(matrix V is of order $Q \times 3$). If the set of truth values $\{s,n,h\}$ is linearly independent, then a solution is provided by

$$\sqrt{N_3} = VBV^+$$

where V^+ is the right Moore-Penrose pseudo-inverse

$$\mathbf{V}^+ = (\mathbf{V}^{\mathrm{T}}\mathbf{V})^{-1}\mathbf{V}^{\mathrm{T}}.$$

6.2 VECTOR LOGIC, FREDKIN GATES AND "REVERSIBILITY".

The looking for reversible computers, with ideally a quasi zero energy consumption is an exiting frontier in contemporary research. Recent advances in this direction come from the field of optics, with the search of optical reversible computer devices. In this domain, the vector representation of logical variables becomes completely natural [29].

The Fredkin gate is a Boolean function with three inputs and three outputs that is widely used in the investigations on reversible computers. It can be defined as follows: (t a b) \rightarrow (t a b), (f a b) \rightarrow (f b a) with a,b \in { t , f } [10] [9] [23] [24]. The selection of an output channel (the second or the third outputs) and the imposition of a given format to the inputs, allow this gate to produce a set of dyadic logical functions, including AND, OR and implication (IMPL), and also the monadic function NOT. The generation of dyadic functions using this gate requires the selection of an output channel and a forced configuration of the input. For instance, in order to obtain the conjunction, we can select the second output position and introduce the input (x y f), with x,y \in { t , f }. The implication is obtained in the second output introducing as input the tern (x t y).

In two-valued vector logic, the Fredkin gate is given by the matrix F that satisfies the following equations ($a, b \in V_2 = \{s, n\}$):

 $F(s \otimes a \otimes b) = s \otimes a \otimes b$ $F(n \otimes a \otimes b) = n \otimes b \otimes a.$

This matrix Fredkin gate can be reprented by the equation

$$\mathbf{F} = \mathbf{s}\mathbf{s}^{\mathrm{T}} \otimes \mathbf{I}_{2}^{[2]} + \mathbf{n}\mathbf{n}^{\mathrm{T}} \otimes \mathbf{R}$$
(14)

(for details see [23]) . $I_2^{[2]} = I_2 \otimes I_2$ (I_2 is the two-valued identity) and R is an interchange matrix such that

$$\mathbf{R}(\mathbf{a}\otimes\mathbf{b})=\mathbf{b}\otimes\mathbf{a}\;.$$

This matrix has the structure

$$\mathbf{R} = \mathbf{s} \otimes \mathbf{I}_2 \otimes \mathbf{s}^{\mathrm{T}} + \mathbf{n} \otimes \mathbf{I}_2 \otimes \mathbf{n}^{\mathrm{T}} \quad . \tag{15}$$

The reversibility of this matrix Fredkin gate means that for three vectorial inputs u,v,w, is $F[F(u \otimes v \otimes w)] = u \otimes v \otimes w$.

The selection of an output position can be performed with a filtering matrix of the form

$$\Phi_2 = (s+n)^T \otimes I_2 \otimes (s+n)^T.$$

The format of the input depends on the logical function we are looking for. The conjunction requires the format matrix

$$\Sigma_{C_2} = I_2 \otimes I_2 \otimes n;$$

the implication requires the format matrix

$$\Sigma_{L_2} = I_2 \otimes s \otimes I_2.$$

Consequently, we obtain the two-valued conjunction and implication operators pre- and post-filtering the Fredkin matrix as follows:

 $C_2 = \Phi_2 F \Sigma_{C_2},$

$$L_2 = \Phi_2 F \Sigma_{L_2} \, .$$

An important problem is the potential cost of these filtering processes in a physical implementation of this reversible gate.

The extension of reversible gates for three-valued logics, has been recently explored and a variety of reversible three-valued logical gates has been analyzed [3], a fact that certainly is important for logic, mathematics and physics. In this context, it is interesting that the direct extension of the Fredkin gate using the same philosophy that led to built up a Kleene three-valued logic destroys reversibility. These "philosophy of construction" implies to impose uncertainty whenever the different choices of the uncertain input do not cancel the uncertainty. For instance, the choice of the conjunction between "true" and "uncertain" is uncertain, because the substitution of "uncertain" by "true" or "false" produces two different outcomes. On the contrary, the conjunction between "false" and "uncertain" is necessarily false. The resulting "Kleenean" three-valued Fredkin gate is a matrix F_K satisfying the following equations ($a, b \in V_3 = \{s, n, h\}$):

$$\begin{split} F_K \left(s \otimes a \otimes b \right) &= s \otimes a \otimes b \\ F_K \left(n \otimes a \otimes b \right) &= n \otimes b \otimes a \\ F_K \left(h \otimes a \otimes b \right) &= \begin{cases} h \otimes h \otimes h \text{ iff } a \neq b \\ h \otimes z \otimes z \text{ iff } a = b \equiv z \end{cases} \end{split}$$

As can be directly proved, an interesting representation for this matrix is

$$\mathbf{F}_{\mathbf{K}} = \mathbf{s}\mathbf{s}^{\mathrm{T}} \otimes \mathbf{I}_{3}^{[2]} + \mathbf{n}\mathbf{n}^{\mathrm{T}} \otimes \mathbf{R}_{3} + \mathbf{h}\mathbf{h}^{\mathrm{T}} \otimes (\mathbf{U}\mathbf{n}\mathbf{c} + \mathbf{F}\mathbf{i}\mathbf{x}),$$

being

$$I_3^{[2]} = I_3 \otimes I_3 ,$$

$$\begin{split} R_{3} &= s \otimes I_{3} \otimes s^{T} + n \otimes I_{3} \otimes n^{T} + h \otimes I_{3} \otimes h^{T}, \\ \text{Unc} &= (h \otimes h) \left[(s \otimes n)^{T} + (n \otimes s)^{T} + (s \otimes h)^{T} + (h \otimes s)^{T} + (n \otimes h)^{T} + (h \otimes n)^{T} \right], \\ \text{Fix} &= (s \otimes s) (s \otimes s)^{T} + (n \otimes n) (n \otimes n)^{T} + (h \otimes h) (h \otimes h)^{T}. \end{split}$$

This matrix F_K is no longer a reversible gate, but if it retains the capacity of bringing the operators of Kleene three-valued logic, under the same selection of the outputs and the same input configuration that work in the two-valued case. For instance, defining

$$\begin{split} \Phi_{\text{K3}} &= (s+n+h)^{\text{T}} \otimes I_3 \otimes (s+n+h)^{\text{T}} \,, \\ \Sigma_{\text{KC}_3} &= I_3 \otimes I_3 \otimes n \,, \end{split}$$

we obtain

$$C_3 = \Phi_{K3} F_K \Sigma_{KC_3}$$

It is interesting the fact that this "Kleenean" Fredkin gate retains its capacity of displaying a variety of three-valued logical gates but, at the same time, loses its reversibility. In a sense, the uncertainty represented by the third logical value h is a kind of "logical friction" or "informational friction", with a superficial analogy with the mechanical friction that makes irreversible the mechanical behaviour of a harmonic oscillator.

7. CONCLUSIONS AND PERSPECTIVES

Vector logic allows submerging the operations of logical calculus into the mechanics of matrix algebra. In particular, this matrix-vector representation allows exploiting the rich properties displayed by Kronecker product factorization. The results of this algebraic formalism is the existence of a set of well defined correspondences between logical and matrix operations. Moreover, many equivalences between logical expressions become identities involving matrix operators. In vector logic, fuzziness was first introduced as linear combinations of vectorial truth values, and in a second step the inputs are processed

by matrix operators. In the case of three-valued vector logic, the third vector h introduces the uncertainty as a specific truth value. Using this third truth value, we can define some basic operators and to built up the implication and the equivalence as derived operators.

Vector logic shows similarities with the matrix models for associative memories, and this fact suggest that the formalism can be an interesting instrument for the modeling of processes of reasoning. It can also help to improve the statement of still unsolved challenging problems (eg: how to model the neural basis of theorem proving?).

A cardinal discover by Lukasiewicz [16] was the existence, in his three-valued logic, of truth tables capable of describing modalities. This discover has a clear manifestation in our linear algebra formalism. In the two-valued vector logic (as in the classical logic), the modal operators "possibility" and "necessity" are obtained as the limit of recursive processes. On the contrary, in the three-valued vector logic these modalities are computed in a single step using a monadic operator, a matrix-vector version of Lukasiewicz's result.

The theory of quantum computation teaches us that the matrix logical operator NOT can be expressed as a product of complex matrices [6, 13]. Vector logic formalism shows natural ways to extend logical formats to the domain of complex numbers in non-quantum systems. On one hand, complex numbers itself are susceptible of being represented as pairs of real numbers, and this fact produce natural vector representations of complex magnitudes using reals. On the other hand, a vector can take its values from the complex domain. In particular it is an interesting fact that the looking for an expression for $\sqrt{N_3}$ generates linear combination of vectors s, n and h, a kind of extended fuzziness with complex coefficients. Another interesting remark is that the third truth value h, being a fixed point ($h = (\sqrt{N_3})h = N_3 h$), prevents complex fuzziness. This excursion into nonquantum complex operators can posses some interest due to the ubiquity, in a variety meso or macroscopic physical devices, of oscillatory behaviors susceptible of being described using complex quantities. These oscillatory systems can naturally confer a potential computational interest to these complex vectorial representations. The development of "complex logics" can be of particular interest for the representations of decision processes that need to include the phases of quasi cyclic fuzzy events [8].

The reversible logical gates provide, at the same time, interesting mathematical constructs and important physical challenges. Logical reversibility requires to select a precise position to read the output, to format the input, and in addition to store the discarded "garbage" in an ordered memory. Many three-valued reversible gates have been reported and carefully investigated [3], and the existence of these reversible gates has a great potential importance from both formal and physical research. The extension of two-valued Fredkin gate to a three-valued Fredkin gate in agreement with "Kleenean" criteria destroys reversibility, and this is an interesting fact: it illustrates the loss of order associated with the loss of precision in the information. In the three-valued Fredkin gate described in Section 6.2, reversibility is preserved in the subcases having s or n as the control input. On the contrary, when the control input is the uncertain vector h, uncertainty invades the transformation and prevents reversibility (except for three fixed points).

Finally, we would like to comment that the matrix-vector formalism presented here provides a compact representation of gates that perform parallel computing processes. Consequently, this formalism could inspire new approaches to search efficient algorithms capable - as happens in the theory of quantum computation (eg. Shor's algorithm) - of exploiting the massive parallelism. If this will be the case, the possibility that these new algorithms could be implemented using classical technologies remains open.

Acknowledgements. The author wishes to thanks two anonymous referees for their constructive comments. This work was partially supported by PEDECIBA-Uruguay and CSIC-UdelaR.

APPENDIX: NUMERICAL ILLUSTRATION

We describe here the structure of some two- and three-valued vector logic matrix operators for the simplest case, with the truth values being normal 2- and 3-dimensional unit vectors.

a) Two-valued matrix operators

The basic vectors are $s = [1 0]^T$ and $n = [0 1]^T$.

The logical identity and the negation are given by the matrices

$$\mathbf{I}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad ; \qquad \mathbf{N}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \; .$$

The conjunction is

 $\mathbf{C}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \ ,$

and the disjunction is given by

$$\mathbf{D}_2 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The implication is

$$\mathbf{L}_2 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \ .$$

Finally, the equivalence and the exlusive-or are given by

$$E_2 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix},$$
$$X_2 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

b) Three-valued matrix operators

Let
$$s = [1 \ 0 \ 0]^T$$
; $n = [0 \ 1 \ 0]^T$; $h = [0 \ 0 \ 1]^T$.

The monadic identity I and negation N matrices are

$$\mathbf{I}_{3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} ; \mathbf{N}_{3} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Note that $N_3N_3 = I_3$

The conjunction C_3 and the disjunction D_3 are

$$\begin{split} \mathbf{C}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \\ \mathbf{D}_3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}. \end{split}$$

Using these matrices, De Morgan's Law $C_3 = N_3D_3(N_3 \otimes N_3)$ can be numerically verified.

The implication L_3 given by $L_3 = D_3(N_3 \otimes I_3)$ is given by

	1	0	0	1	1	1	1	0	0
L ₃ =	0	1	0	0	0	0	0	0	0
	0	0	1	0	0	0	0	1	1

On the other hand, Lukasiewicz implication matrix L_P is given by

$$\mathbf{L}_{\mathbf{P}} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The matrices corresponding to equivalence E_3 and exclusive-or X_3 , are

 $E_{3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$ $X_{3} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$

We have previously showed that in three-valued vector logic the modalities possibility and necessity are computed by monadic operators, represented by the square matrices Pos and Nec. In the simple orthonormal basis used in this section, they are

$$Pos = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}; Pos = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

We can immediately confirm Aristotle's theorem: $Nec = N_3 Pos N_3$.

Lewis' "strict implication", is given by

 $Lew = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

The alternative Lewis matrix constructed using Lukasiewicz matrix implication, Lew_P = Nec L_P, is

 $Lew_{P} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

REFERENCES

[1] G. Boole, *An Investigation of the Laws of Thought, on which are Founded the Theories of Logic and Probabilities.* London: Macmillan, 1854; New York: Dover Reedition, 1958.

[2] J.L.Borges, "El ruiseñor de Keats", in *Obras Completas*, pp. 717-719, Buenos Aires,
 EMECE, 1974 (English translation: "The Nightingale of Keats" Other Inquisitions, Texas:
 University of Texas Press, 1964).

[3] G. Cattaneo, A. Leporati and R. Leporini, "Quantum conservative gates for finite-valued loics", *International Journal of Theoretical Physics*, vol. 43, pp. 1769-1791, 2004.

[4] L.N. Cooper, "Memories and memory: a physicist's approach to the brain", *International Journal of Modern Physics*, vol.15, pp. 4069-4082, 2000.

[5] I.M. Copilovich, "Matrix development of the calculus of relations", Journal of Symbolic Logic, vol. 13, pp. 193-203, 1948.

[6] D. Deutsch, A. Ekert, and R.Lupacchini, "Machines, logic and quantum physics", *Bulletin of Symbolic Logic*, vol. 3, pp. 265-283, 2000.

[7] S.B. Diagne, Boole: L'Oiseau de Nuit en Plein Jour. Paris: Belin, 1989

[8] S. Dick, "Towards complex fuzzy logic", *IEEE Transactions on Fuzzy Systems*, vol. 15, pp. 405-414, 2005.

[9] R.P. Feynman, Lectures on Computation, Cambridge: Perseus, 1999.

[10] E. Fredkin and T. Toffoli, "Conservative logic", *International Journal of Theoretical Physics*, vol. 21, pp. 219-253, 1982.

[11] A. Graham, *Kronecker Products and Matrix Calculus with Applications*, Chichester: Ellis Horwood, 1981.

[12] N. Houser, "Algebraic logic from Boole to Schröder, 1840-1900", in *Companion Encyclopedia of the History and Philosophy of the Mathematical Sciences (Vol. 1)*, I. Grattan-Guinness (Ed.), pp. 600-615, London: Routledge, 1994.

[13] B. Hayes, "The square root of NOT", American Scientist, vol. 83, pp. 304-308 1995.

[14] G.J. Klir and B. Yuan, Fuzzy Sets and Fuzzy Logic, New Jersey: Prentice-Hall, 1995.

[15] T. Kohonen, *Associative Memory: A System-Theoretical Approach*, New York: Springer-Verlag, 1977.

[16] J. Lukasiewicz, "Philosophical remarks on many-valued logic", 1930; reprinted in *J. Lukasiewicz, Selected Works*, edited by L. Borkowski, pp.153-178, Amsterdam: North-Holland, 1980.

[17] W. S. McCulloch and W. Pitts, "A logical calculus of the ides immanentin nervous activity", *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115-133, 1943

[18] M.L. Minsky and S.A. Papert, *Perceptrons* (expanded edition), Cambridge MA: MIT Press, 1988.

[19] E. Mizraji, "Context-dependent associations in linear distributed memories", *Bulletin* of Mathematical Biology, vol. 50, pp. 195-205, 1989.

[20] E. Mizraji, "Vector logics: the matrix-vector representation of logical calculus", *Fuzzy Sets and Systems*, vol. 50, pp. 179-185, 1992.

[21] E. Mizraji, "Topics in vector logic", in Proceedings of the IX Latin American Symposium on Mathematical Logic (Part 2), *Notas de Lógica Matemática*, vol. 39, pp. 159-165, 1994.

[22] E. Mizraji, "Modalities in vector logic", *Notre Dame Journal of Formal Logic*, vol. 35, pp. 272-283, 1994.

[23] E. Mizraji, "The operators of vector logic", *Math. Log. Quart.*, vol. 42, pp. 27-39, 1996.

[24] E. Mizraji, "The parts and the whole: inquiring how the interaction of simple subsystems generates complexity", *International Journal of General Systems*, vol. 35 (4), pp. 395-415, 2006.

[25] E. Mizraji and J. Lin, "The dynamics of logical decisions", *Physica D*, vols.168-169, pp. 386-396, 2002.

[26] A. Pomi and E. Mizraji, "A cognitive architecture that solves a problem stated by Minsky", *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 31, pp. 729-734, 2001.

[27] A. Pomi and E. Mizraji, "Semantic graphs and associative memories", *Physical Review E*, vol.70, pp. 066136(1-6), 2004.

[28] N. Rescher, Many-Valued Logic, New York: McGraw-Hill, 1969

[29] J. Westphal and J. Hardy, "Logic as a vector system", *Journal of Logic and Computation*, vol. 15, pp. 751-765, 2005.