

Correspondence Matrices; Algorithms for Propositional Logic.

Brian Droncheff

Email: Brian@Thinkinginnovative.com

Abstract

This paper gives a new way to represent logical expressions as well as a novel algorithm for the computation of logical expressions in propositional logic. The findings presented show that logical operators can be represented as binary matrices called correspondence matrices (CM's) and furthermore that many logical expressions can be computed by operating on their CM's. The linearity of CM's is proven and it is shown that CM's can be used as measurement operators in a similar fashion as is done in quantum computation. Representing logical operators as matrices allows for the use of matrix transformations and it is found that the negation, rotation and transformation of CM's allow for the transformation of a logical expression in to another. The notion of CM's are generalized in to logical measurement matrices (LM's) which are used to measure relationships between logical expressions. Other operations between logical expressions e.g. 'quotienting' and methods for the (de)composition of logical expressions are discussed as well. In finishing it is shown how LM's and their unique CM's can be constructed to compute finite length logical expressions.

0.1 Keywords

Keywords:

Logic, Propositional Logic, Matrix Operators, Quantum Computation, Logical Calculus, Matrix Mechanics, Circuit Design, Algorithms, Representation Theory, Distributive Algorithms, Parallel Processing.

1 Introduction

This paper gives a new way in which to represent and combine propositional logical expressions. Most notably, this paper shows that operators themselves can be represented by binary correspondence matrices (CM's) thus allowing logical expressions to be transformed via matrix transformations, furthermore that logical expressions can be combined based solely on their respective CM's. This new representation may reduce the time complexity of computation, not only due to the ability in which CM's can be combined but because CM's are binary. Direct manipulation of binary values may cut down on the computation and possible issues arising from having non-binary information representations. Also as CM operators can be broken apart both less complex and smaller matrices they give the ability for distributive computation and parallel processing.

To immediately see motivating examples: 1. Computation of a two variable logical expression using a CM [3]; 2. Rotation, Transposition and negation of CM's for computation [3.1]; 3. Quotienting of one logical expression by another [3.2.2]; 4. Combining logical expressions [3.2.3]; 5. Measurement by a LM [4.1.4]; 6. Computation of a four variable logical expression using a CM [5.1.1]; 7. Combining four variable logical expressions [5.4.2]. As correspondence matrices and the more general logical matrices are structures introduced in this paper, there is very little external material supporting them. Because of this, nearly everything in this paper was developed specifically to justify the derivation and implementation of CM's. Many other interesting faucents of these matrices were discovered in development and are added for completeness.

The algorithm presented has been developed in such a fashion as to be used in quantum computation. While quantum correlations have not be explicitly introduced in this paper, there are several ideas developed which have a relationship to postulates of quantum mechanics, mainly that in which the developed correspondence matrices act as a type of measurement matrix. As these correspondence matrices are linear and act similar to measurement matrices in quantum mechanics, they offer a great stepping stone in understanding some of the fundamental concepts in this area.

2 Fundamentals

2.1 Basic Logic

We will denote a *logical variable* or *literal* to be that which can be assigned or given a value of true or false. The assignment of truth to one or more logical variables is called a *truth assignment*. The value of being true is denoted by T or 1 and that of being false by F or 0. We can *operate* on one or more logical variables via a *logical operator* or *binary operator* in which the logical variables will be one or two *operands*. A *logical expression* \mathcal{L} is a correctly formatted combination of one or more logical sub-expressions and operators, a logical

sub-expression itself being a logical expression comprised of one or more logical variables and operators. Through the paper any single letter representing a variable will denote both a logical variable or expression i.e. X may refer to a single literal or, such as \mathcal{L} a logical expression, the context will be evident from the usage. A *logical operator variable* can be given the value of any logical operator. We will denote logical operator variables by greek letters, usually Θ .

We follow propositional logic^[1] let \neg be the negation operator and define the negation rules, $\neg 1 = 0$, $\neg 0 = 1$, $\neg(\mathcal{L}) = \neg\mathcal{L}$ and $\neg(\neg\mathcal{L}) = \mathcal{L}$. The negation of a logical expression is called its *compliment*. A non-negated literal X is called an *atom*. We will let the conjunction, denoted by the symbol \wedge , be a binary operator between two logical variables X and Y written $X \wedge Y$ or XY for short, and the disjunction \vee be a binary operator between X and Y written as $X \vee Y$. The equations denoted throughout the paper will work for both single and finite multi-variable logical expressions.

For reasons which will be explained later, the XOR operator will be denoted symbolically by \Downarrow instead of the more commonly used \oplus symbol. We will start with a few important relations:

$X \wedge 0 = 0$	$X \vee 0 = X$	$X \wedge 1 = X$	$X \vee 1 = 1$
$X \Downarrow 1 = \neg X$	$X \Downarrow 0 = X$	$\neg X \Downarrow 1 = X$	$\neg X \Downarrow 0 = \neg X$

The difference in XOR notation can be seen in the following application of DeMorgans law:

$$\boxed{\neg(X \Downarrow Y) = X \Leftrightarrow Y \quad \neg(X \Leftrightarrow Y) = X \Downarrow Y}$$

where \Leftrightarrow is the notation commonly used for the XNOR operator. We will call the representation of a logical expression in which each of its sub-expressions are XOR'ed with one-another the *XOR decomposition*. Each sub-expression of the XOR decomposition is called a *component* of the XOR decomposed expression. Likewise the *XOR composition* or *XOR sum* is the composition of a logical expression from components which are XOR'ed together. The XOR decomposition is in algebraic normal form (ANF) whereas the XOR composition does not have to be.

2.1.1 Valuations

A *valuation* \mathcal{V} is the result of the assignment of truth values to a logical expression or an operator to a logical operator variable. We will denote the valuation of the logical expression \mathcal{L} as $\mathcal{V}(\mathcal{L}) = V$ where $V \in \{T, F\}$ and say that an expression which is not valuated to be *unevaluated*. The *positive valuation* $\mathcal{V}(\mathcal{L}) = T$ and the *negative valuation* $\mathcal{V}(\mathcal{L}) = F$ are denoted by $\mathcal{V}^T(\mathcal{L})$ and $\mathcal{V}^F(\mathcal{L})$ respectively. Each $\mathcal{V}^T(\mathcal{L})$ and $\mathcal{V}^F(\mathcal{L})$ correspond to a set of truth assignments which make the expression \mathcal{L} respectively true or false. We say that the logical expression \mathcal{L} has been assigned a truth value or equivalently has a *truth assignment* V if $\mathcal{L} = V$ and denote this truth assignment as $\mathcal{V}_V(\mathcal{L})$. We

will let a $\mathcal{V}_T(\mathcal{L})$ be that of a *positive assignment* in which $\mathcal{L} = T$ and that of *negative assignment* denoted by $\mathcal{V}_F(\mathcal{L})$ in which $\mathcal{L} = F$. A positive or negative assignment of a logical expression \mathcal{L} assigns true or false to every non-negated variable (*atom*) or sub-expression respectively.

As indicated in the name, a logical operator variable Θ is itself a variable. We can apply a valuation in which we have that $\mathcal{V}(\Theta)$ is in the set of logical operators e.g. $\mathcal{V}(\Theta) = \wedge$. Like a logical expression which has not been assigned or valued to have a particular truth value, a logical operator variable doesn't have a particular value until assigned or valued to be a fixed operator. The valuation of a logical expression $\mathcal{L} = X\Theta Y$ has the form $\mathcal{V}(\mathcal{L}) = \mathcal{V}(X)\mathcal{V}(\Theta)\mathcal{V}(Y)$ or for a particular assignment value V , $\mathcal{V}_V(X)\Theta\mathcal{V}_V(Y)$. We will overload (abuse) notation for a logical variable operator Θ and call it a logical operator (as opposed to $\mathcal{V}(\Theta)$) when no confusion arises.

We can see from above $\mathcal{V}(\mathcal{L})$, that expressions have a particular valuation depending on their truth assignments, but also that we may have truth assignments being dependent on the expressions valuation. As an example, in one direction: $\mathcal{V}^T(X \wedge Y) \Rightarrow (\mathcal{V}_T(X) \wedge \mathcal{V}_T(Y))$ however it is also true that $(\mathcal{V}_T(X) \wedge \mathcal{V}_T(Y)) \Rightarrow \mathcal{V}^T(X \wedge Y)$. Combining the two expressions above and a similar computation shows that:

$$\mathcal{V}^T(X \wedge Y) \Leftrightarrow (\mathcal{V}_T(X) \wedge \mathcal{V}_T(Y)) \text{ and } \mathcal{V}^T(X \vee Y) \Leftrightarrow (\mathcal{V}_T(X) \vee \mathcal{V}_T(Y))$$

These equivalences are important in order to be able to distinguish between directly assigning a value (assignment) to logical variables i.e. $\mathcal{V}_T(X \vee Y)$ giving $X = Y = T$ and a valuation i.e. $\mathcal{V}^T(X \vee Y)$ where only one of X or Y must be true. In a valuation then it is possible that a particular variable may not be forced to have an assignment so as to make the expression true.

Concerning negation we have: $\neg\mathcal{V}^V(\mathcal{L}) = \neg(\mathcal{V}(\mathcal{L}) = V) \Leftrightarrow (\mathcal{V}(\mathcal{L}) \neq V)$ however $(\mathcal{V}(\mathcal{L}) \neq V) \Leftrightarrow (\mathcal{V}(\mathcal{L}) = \neg V)$ as V is binary, so $\neg\mathcal{V}^V(\mathcal{L}) \Leftrightarrow \mathcal{V}^{-V}(\mathcal{L})$. We also find that $\neg\mathcal{V}^V(\mathcal{L}) \Leftrightarrow \mathcal{V}^V(\neg\mathcal{L})$, hence $\mathcal{V}^{-V}(\mathcal{L}) = \mathcal{V}^V(\neg\mathcal{L})$ and we have that:

$$\neg\mathcal{V}^V(\mathcal{L}) \Leftrightarrow \mathcal{V}^{-V}(\mathcal{L}) \Leftrightarrow \mathcal{V}^V(\neg\mathcal{L})$$

With this equality we can re-write the negative valuation of a logical expression by the positive valuation of that same expressions negation.

2.2 Bra-Kets and State Vectors

To start, let a *logical state* be a negated or non-negated, valued or unvalued logical expression e.g. $1, 0, X$, and $\neg X$ are all logical states. We will call a *logical statevector* a matrix consisting of logical states. For example both $[1, 0]$ and $[X, \neg X]$ are logical statevectors.

The notion of a bra and ket^{[2],[3]} are ways to represent logical statevectors. Let Ψ be a statevector. We call bra Ψ the *bra-vector* for Ψ , denote it is as $\langle\Psi|$

and represent it by a $1 \times n$ matrix of logical expressions. We call ket Ψ the *ket-vector* for Ψ and denote it as $|\Psi\rangle$ the $n \times 1$ logical matrix. The bra of Ψ is related to ket of Ψ by $\langle\Psi|^T = |\Psi\rangle$ and both are logical statevectors. The inner product between $\langle\Psi|$ and $|\Psi\rangle$ is denoted $\langle\Psi|\Psi\rangle$, the outer product^[4] or tensor product by $|\Psi\rangle \otimes \langle\Psi| = |\Psi\rangle\langle\Psi|$, and for another statevector $|\Lambda\rangle$ we let $|\Psi\rangle|\Lambda\rangle = |\Psi\rangle \otimes |\Lambda\rangle$.

Let Ψ_X be a two element statevector. We denote the respective bra and ket vectors as such:

$$\langle\Psi_X| = [X_1, X_2] \text{ and } |\Psi_X\rangle = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

where $X_1 = X$ and $X_2 = \neg X$. We define $|\Psi_{\neg X}\rangle = [X_2, X_1]^T$ and have the relationship to $|\Psi_X\rangle$ via negation:

$$\neg|\Psi_X\rangle = \neg[X_1, X_2] = [\neg X_1, \neg X_2] = [X_2, X_1] = |\neg\Psi_X\rangle = |\Psi_{\neg X}\rangle = |\Psi_X\rangle_2$$

In general we find that $|\Psi_{X_i}\rangle = [X_i, \neg X_i]^T$ and $\neg|\Psi_{X_i}\rangle = |\Psi_{\neg X_i}\rangle = [\neg X_i, X_i]$. Because of the relationship to the subscript in the bra (and ket) statevector to the state-vector itself i.e. X in $|\Psi_X\rangle$, we change the notation. The new notation is better suited to propositional logic. In the rest of the paper when no confusion arises we will make the following changes:

$$|\Psi_{X_i}\rangle = |X_i\rangle$$

For example, $|\Psi_{X_1}\rangle = |X_1\rangle = |X\rangle$ and $|\Psi_{X_2}\rangle = |X_2\rangle = |\neg X\rangle$.

Let $[\Theta]$ be the 4-element square matrix with elements Θ_{ij} in which $\mathcal{V}(\Theta_{ij}) \in \{0, 1\}$. We will call $[\Theta]$ an *operator* when it acts upon any other matrix. The action of an operator is called an *operation* and the logical state vectors which are operated on are called *operands*. Operations are as follows:

$$\langle X|[\Theta] = \begin{bmatrix} X_i\Theta_{i1} \\ X_i\Theta_{i2} \end{bmatrix}^T \text{ and } [\Theta]|Y\rangle = \begin{bmatrix} \Theta_{1k}Y_k \\ \Theta_{2k}Y_k \end{bmatrix} \text{ where } [\Theta] = \begin{bmatrix} \Theta_{11} & \Theta_{12} \\ \Theta_{21} & \Theta_{22} \end{bmatrix}$$

where $\langle X|[\Theta]|Y\rangle = X_i\Theta_{ij}Y_j$ which is the Einstein notation^[5] for the XOR sum $\uparrow_{i,j} X_i\Theta_{ij}Y_j = X_i\Theta_{ij}Y_j = X_1\Theta_{11}Y_1 \uparrow \cdots \uparrow X_2\Theta_{22}Y_2$. It doesn't matter whether we operate first on $\langle X|$ or $|Y\rangle$:

$$(\langle X|[\Theta])|Y\rangle = \langle X|([\Theta]|Y\rangle) = X_i\Theta_{ij}Y_j$$

which is proved simply as we can distribute over XOR.

We can combine logical statevectors over the logical operator Φ in the following fashion:

$$|X\rangle \Phi |Y\rangle = \begin{bmatrix} X\Phi Y \\ \neg X\Phi \neg Y \end{bmatrix}$$

for example $|X\rangle \uparrow |\neg X\rangle = [1, 1]^T$ and $|X\rangle \uparrow |X\rangle = [0, 0]^T$.

3 Correspondence Matrices (CM's)

Correspondence matrices (CM's) are a new way to represent operators in propositional logic. Not only do these representations give a new way to visualize logical expressions, they allow for new types of logical operations. They offer the ability to quotient logical expressions by one-another [3.2.2] as well as constraints on ways in which logical expressions can be (de)composed. Possibly the most significant findings are that CM's allow matrix operations to be applied to logical expressions thus allowing the transformation [3.1] and computation of logical expressions [3.2.3]. Furthermore, correspondence matrices are shown to be linear and can be interpreted as measurement operators [4.1] operating on logical expressions, thus giving a possible use in fields such as quantum computation.

The foundation for the development of CM's is the fact that every propositional logical expression can be written as the XOR'ing of some set of logical sub-expressions. In particular, consider the logical expression $\mathcal{L} = X\Theta Y$ which can be written as a conjunction of some XOR combination of one or more elements of the set $\{XY, X\neg Y, \neg XY, \neg X\neg Y\}$:

$$X\Theta Y = \Theta_{11}XY \updownarrow \Theta_{12}X\neg Y \updownarrow \Theta_{21}\neg XY \updownarrow \Theta_{22}\neg X\neg Y = \Theta_{ij}X_iY_j$$

where $\mathcal{V}(\Theta_{ij}) \in \{0, 1\}$ and $X_i\Theta_{ij} = \Theta_{ij}X_i$ by the commutativity of \wedge . For each i, j combination $\Theta_{ij}X_iY_j$ is called a *component*. For example $X \Rightarrow Y$ can be re-written as the XOR combination of components: $1 \wedge XY \updownarrow 0 \wedge X\neg Y \updownarrow 1 \wedge \neg XY \updownarrow 1 \wedge \neg X\neg Y$ where $\mathcal{V}(\Theta_{11}) = \mathcal{V}(\Theta_{21}) = \mathcal{V}(\Theta_{22}) = 1$ and $\mathcal{V}(\Theta_{12}) = 0$.

We can write the expansion of $X\Theta Y$ more succinctly as:

$$X\Theta Y = \langle X | [\Theta] | Y \rangle = X_i \Theta_{ij} Y_j$$

in which each logical operator Θ can be uniquely represented as the matrix $[\Theta]$. This uniqueness can be seen by direct computation; that for two logical operators Θ_1 and Θ_2 that $\Theta_1 = \Theta_2$ if and only if $X_i\Theta_{1,ij}Y_j = X_i\Theta_{2,ij}Y_j$. We can build correspondence matrices from the ground up. Similar to the valuation of a logical measurement matrix as seen in 4.1, we can value a logical state-vector which in the single variable case we get;

$$|\mathcal{V}_T(X)\rangle = \begin{bmatrix} \mathcal{V}_T(X) \\ \mathcal{V}_T(\neg X) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |1\rangle \text{ and } |\mathcal{V}_F(X)\rangle = \begin{bmatrix} \mathcal{V}_F(X) \\ \mathcal{V}_F(\neg X) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |0\rangle$$

we will call $|1\rangle$ and $|0\rangle$ be the respective 1 and 0 *ket-vectors*. There are four possible tensor product combination of 1 and 0 ket-vectors;

$$|1\rangle\langle 1| = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, |1\rangle\langle 0| = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, |0\rangle\langle 1| = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, |0\rangle\langle 0| = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

which are the four CM *basis matrices*, each with a single *feature* or positive entry. The set of basis matrices constitutes a *basis* in which we can create any correspondence matrix. For example, we can derive the CM $[\vee]$ which is the matrix operator of the of logical OR (\vee);

$$|1\rangle\langle 1| \updownarrow |1\rangle\langle 0| \updownarrow |0\rangle\langle 1| = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \updownarrow \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \updownarrow \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} = [\vee]$$

Op.	Matrix	Op.	Matrix	Op.	Matrix	Op.	Matrix
$[\boxplus]$	$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$	$[0]$	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$	$[\Leftrightarrow]$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$[\updownarrow]$	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
$[\wedge]$	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	$[\neg\vee]$	$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$	$[\uparrow]$	$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$	$[\downarrow]$	$\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$
$[\Rightarrow]$	$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$	$[\Leftarrow]$	$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$	$[\vee]$	$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$	$[\neg\wedge]$	$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$
$[R]$	$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$	$[\neg R]$	$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$	$[L]$	$\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$	$[\neg L]$	$\begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$

Figure 1: The set of 16 distinct correspondence matrices

To see how to find the corresponding logical expression from a correspondence matrix, consider the CM $[\Rightarrow]$ which we use to operator on vectors $|X\rangle^T$ on the left side and $|Y\rangle$ on the right;

$$\langle X|[\Rightarrow]|Y\rangle = [X, \neg X] \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} Y \\ \neg Y \end{bmatrix} = [X \updownarrow \neg X, \neg X] \begin{bmatrix} Y \\ \neg Y \end{bmatrix} = Y \updownarrow \neg X \neg Y = X \Rightarrow Y$$

As was the case in which, for a logical operator variable Θ where $\mathcal{V}(\Theta)$ is in the set of logical operators, we will define $\mathcal{V}[\Theta]$ to be in the set of correspondence matrices where \mathcal{V} evaluates $[\Theta]$ index-wise. For short, we will call $[\Theta]$ a CM in that, as Θ can be fixed to a particular operator, $[\Theta]$ can be fixed to a particular CM. We will say that $[\Theta]$ is *commutative* if $[\Theta] = [\Theta]^T$ and can see that commutativity in CM's corresponds to commutativity in their corresponding logical operator e.g. $[\Leftrightarrow]$, $[\updownarrow]$, $[\wedge]$, and $[\vee]$ are all commutative. It is important to mention that operating on the logical bra-ket pairs can be done with disjunctions \vee instead of exclusive disjunctions \updownarrow in the resulting computation that is; for any logical expression \mathcal{L} we can write $\mathcal{L} = \vee_{i,j}(\Theta_{ij}X_iY_j)$.

The reasoning for representing the XOR operator as \updownarrow as opposed to the more commonly used \oplus can now be seen as $[\updownarrow]$, the CM corresponding to \updownarrow , is a 90° rotation (in either direction) of $[\Leftrightarrow]$ which corresponds to \Leftrightarrow . Indeed, we can see that $\neg[\Leftrightarrow] = [\updownarrow] = [\Leftrightarrow]^{90^\circ}$. Furthermore we can see that the *impax* CM $[\boxplus]$ is the overlap of both the CM's $[\updownarrow]$ and $[\Leftrightarrow]$ both logically as well as symbolically. The impax operator is tautological in that its operation on any two expressions give a tautology, likewise $[0]$ is contradictory; that is, $\langle X|[\boxplus]|Y\rangle = 1$ and $\langle X|[0]|Y\rangle = 0$ respectively.

Correspondence matrices are truth tables ‘rolled up’ and as such they are isomorphic to truth tables of logical operators by construction. Because of this isomorphism, if two logical expressions consist of the same variables and variable ordering, we can compare these expressions by comparing their respective CM’s. For example, consider $\mathcal{L}_1 = X \vee Y$ and $\mathcal{L}_2 = X \wedge Y$, as they both have the same variables and respective X and Y ordering, we can compare the CM’s $[\vee]$ and $[\wedge]$ associated with their respective operators \vee and \wedge . In doing so we can see that $[\vee] = [\wedge] \oplus [\updownarrow]$ and that $X \vee Y = (X \wedge Y) \updownarrow (X \updownarrow Y)$.

This ability to compare CM’s is useful as we can create a hierarchy of relations based on feature relationship. For example, above we can see that there is a relationship between \vee and \wedge as $[\vee]$ contains all the features in $[\wedge]$. More generally we say that any CM in a decomposition is *contained* in the decomposed CM if it does not have any features not in that which was decomposed. The significance of operator containment is that containment gives a quantifiable constraint on how logical operators can be related to one-another and may limit the search space for possible decompositions.

3.1 Correspondence Matrix Relationships

One of the benefits of logical CM’s is that, like any matrix, computations can be performed by rotations and transformations. There are some very interesting matrix operations which can help in computing logical expressions; to see this we will denote $[\Theta]^n$ as the clockwise rotation of the correspondence matrix $[\Theta]$ where $n \in \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$ and the transpose T is the usual swapping of the row with the column for each matrix element. Furthermore let $\langle X|[\Theta]^z|Y \rangle = \langle X|[\Theta]|Y \rangle^z$ for any combination of matrix operations z .

We have the five relation rules, each of which can be proved simply by expanding and showing equality:

1. $\langle Y|[\Theta]|X \rangle = \langle X|[\Theta]^T|Y \rangle = \langle X|[\Theta]|Y \rangle^T$

This rule follows from the general property of matrices and demonstrates the ability to transpose a logical expression solely by transposing the CM. As an example $Y \Rightarrow X = \langle Y|[\Rightarrow]|X \rangle = \langle X|[\Leftarrow]|Y \rangle = X \Leftarrow Y$.

2. $\neg \langle X|[\Theta]|Y \rangle = \langle X|[-\Theta]|Y \rangle$ where $[-\Theta] = \neg[\Theta]$ see section 3.3

This very powerful result tells us that every CM, hence logical operator, has a corresponding unique negation. As an example $\neg(X \wedge Y) = \neg \langle X|[\wedge]|Y \rangle = \langle X|[\neg\wedge]|Y \rangle = \neg X \vee \neg Y$. We can prove this by noting that $\neg \langle X|[\Theta]|Y \rangle = \neg(X_i \Theta_{ij} Y_j) = X_i \neg \Theta_{ij} Y_j$ after applying DeMorgans laws and rewriting the expression in conjunctive normal form (CNF).

3. $\langle X|[\Theta]| \neg Y \rangle = \langle X|[\Theta]^{T+90^\circ}|Y \rangle = \langle X|[\Theta]|Y \rangle^{T+90^\circ}$

where $[\Theta]^{T+90^\circ} = [[\Theta]^T]^{90^\circ}$ that is, the matrix transpose T is applied first and then the clockwise rotation of 90° is applied. To see an example con-

sider the expression $X \Rightarrow \neg Y$ where $\Theta = \Rightarrow$. After applying the computation $[\Rightarrow]^{T+90^\circ} = [\Leftarrow]^{90^\circ} = [\neg \wedge]$ we have that $X \Rightarrow \neg Y = \neg X \vee \neg Y$.

$$4. \langle \neg X | [\Theta] | Y \rangle = \langle X | [\Theta]^{T+270^\circ} | Y \rangle = \langle X | [\Theta] | Y \rangle^{T+270^\circ}$$

which is similar to the transformations in rule 3., the difference being that we rotate the expression 270° clockwise or 90° counter-clockwise. Note that the 180° difference in rotations between rule 3. and 4. is equivalent to the negation of both operands as shown in the next rule.

$$5. \langle \neg X | [\Theta] | \neg Y \rangle = \langle X | [\Theta]^{180^\circ} | Y \rangle = \langle X | [\Theta] | Y \rangle^{180^\circ}$$

This rule can be derived by applying both rules 1. and 2. in either order as $[[\Theta]^{T+90^\circ}]^{T+270^\circ} = [[\Theta]^{T+270^\circ}]^{T+90^\circ} = [\Theta]^{180^\circ}$. As an example $\neg X \vee \neg Y = \langle \neg X | [\vee] | \neg Y \rangle = \langle X | [\neg \wedge] | Y \rangle$.

Concerning these transformation rules we can see that there is consistency over negation and change of variables as, for example, applying rule 4. to $\langle \neg W | [\Theta] | Y \rangle = \langle W | [\Theta] | Y \rangle^{T+270^\circ}$, letting $W = \neg X$ and applying rule 4. again we find that $(\langle X | [\Theta] | Y \rangle^{T+270^\circ})^{T+270^\circ} = \langle \neg W | [\Theta] | Y \rangle$ as we would expect.

3.2 CM Properties

3.2.1 Linearity and Decomposition

One of the most useful attributes of the correspondence matrices is that they are linear over \uparrow . To show this we must show that:

$$\langle X | [\Theta_1] \uparrow [\Theta_2] | Y \rangle = \langle X | [\Theta_1] | Y \rangle \uparrow \langle X | [\Theta_2] | Y \rangle$$

We do this by first defining the function $\xi : [\Theta] \rightarrow \langle X | [\Theta] | Y \rangle$ and showing that ξ is a linear function over \uparrow , that is that $\xi([\Theta_1] \uparrow [\Theta_2]) = \xi([\Theta_1]) \uparrow \xi([\Theta_2])$. As the left side of the expression can be equivalently written as the XOR sum $X_i \Theta_{1,ij} \uparrow \Theta_{2,ij} Y_j$, $i, j \in \{1, 2\}$, \wedge distributes over XOR, and for example $X_i = X_i \wedge X_i$, we can distribute X_i and Y_j term by term and find that:

$$X_i (\Theta_{1,ij} \uparrow \Theta_{2,ij}) Y_j = X_i \Theta_{1,ij} Y_j \uparrow X_i \Theta_{2,ij} Y_j = \xi([\Theta_1]) \uparrow \xi([\Theta_2])$$

We then have that logical operators can be expanded in a linear fashion over XOR.

An important result of linearity over \uparrow is that any correspondence matrix which has more than a single positive entry can be re-written as a XOR decomposition of basis matrices:

$$[\Theta] = \Theta_{11}[\wedge] \uparrow \Theta_{12}[\uparrow] \uparrow \Theta_{21}[\Downarrow] \uparrow \Theta_{22}[\neg \vee]$$

This gives that any logical expression $X \Theta Y$ can be represented piece-wise:

$$\langle X | [\Theta] | Y \rangle = \langle X | [\Theta_{11}] | Y \rangle \uparrow \langle X | [\Theta_{12}] | Y \rangle \uparrow \langle X | [\Theta_{21}] | Y \rangle \uparrow \langle X | [\Theta_{22}] | Y \rangle = X_i \Theta_{ij} Y_j$$

which is the result as seen in section 3. As an example, the expression $\mathcal{L} = X \vee Y$ can be broken apart to $\mathcal{L} = \langle X | [\vee] | Y \rangle = \langle X | [\uparrow] | Y \rangle \Downarrow \langle X | [\Downarrow] | Y \rangle \Downarrow \langle X | [\wedge] | Y \rangle$. As the bra's $\langle X |$ and ket's $| Y \rangle$ are the same in each XOR'ed bra-ket expression we can find the decomposition (or composition) by looking solely at the correspondence matrices e.g. $[\vee] = [\uparrow] \Downarrow [\Downarrow] \Downarrow [\wedge]$.

We go a step further and find that we can decompose a logical expression $\mathcal{L} = X \Theta Y$ in to a combination of sub-expressions operated on by any logical operator Φ . That is, for $\Theta = \Theta_1 \Phi \Theta_2$:

$$X \Theta Y = \langle X | [\Theta] | Y \rangle = \langle X | [\Theta_1] | Y \rangle \Phi \langle X | [\Theta_2] | Y \rangle = (X \Theta_1 Y) \Phi (X \Theta_2 Y)$$

To see this, as every element Θ_{ij} in a $[\Theta]$ can be seen to be a combination of binary operands $\Theta_{1,ij}$ and $\Theta_{2,ij}$ operated on by some logical operator Φ i.e. $\Theta_{ij} = \Theta_{1,ij} \Phi \Theta_{2,ij}$, index-wise we have:

$$[\Theta] = \begin{bmatrix} \Theta_{1,11} \Phi \Theta_{2,11} & \Theta_{1,12} \Phi \Theta_{2,12} \\ \Theta_{1,21} \Phi \Theta_{2,21} & \Theta_{1,22} \Phi \Theta_{2,22} \end{bmatrix} = \begin{bmatrix} \Theta_{1,11} & \Theta_{1,12} \\ \Theta_{1,21} & \Theta_{1,22} \end{bmatrix} \Phi \begin{bmatrix} \Theta_{2,11} & \Theta_{2,12} \\ \Theta_{2,21} & \Theta_{2,22} \end{bmatrix} = [\Theta_1] \Phi [\Theta_2]$$

As $\langle X | [\Theta] | Y \rangle = X_i \Theta_{ij} Y_j$ and $\Theta_{ij} = \Theta_{1,ij} \Phi \Theta_{2,ij}$, we have $\langle X | [[\Theta_1] \Phi [\Theta_2]] | Y \rangle = X_i (\Theta_{1,ij} \Phi \Theta_{2,ij}) Y_j$. As $\Theta_{1,ij}$ and $\Theta_{2,ij}$ are themselves logical variables i.e. $\mathcal{V}(\Theta_{1,ij}) \in \{0, 1\}$ we have: $\Theta_{1,ij} \Phi \Theta_{2,ij} = \langle \Theta_{1,ij} | [\Phi] | \Theta_{2,ij} \rangle = (\Theta_{1,ij})_k \Phi_{kl} (\Theta_{2,ij})_l$. As Φ_{kl} is also a binary variable and \wedge is commutative we can distribute copies of X_i and Y_j over the binary operators giving the full expression $\mathcal{L} = (X_i (\Theta_{1,ij})_k Y_j) \Phi_{kl} (X_i (\Theta_{2,ij})_l Y_j)$. We can employ rule 2. [3.1] such that, for example $(X_i (\Theta_{1,ij})_k Y_j) = (X_i \Theta_{1,ij} Y_j)_k$ and have:

$$(X_i \Theta_{1,ij} Y_j)_k \Phi_{kl} (X_i \Theta_{2,ij} Y_j)_l = \langle X | [\Theta_1] | Y \rangle \Phi \langle X | [\Theta_2] | Y \rangle$$

which proves the result that any logical expression dependent on any operator Θ can be decomposed to some expression with sub-expressions dependent on operators Θ_1 and Θ_2 over a fixed operator Φ .

There are usually many possible decompositions for any CM. For example, consider the expression $X \Downarrow Y$ in which, for the operator \Downarrow , we have the two possible decompositions $[\Downarrow] = [\Leftrightarrow] \Rightarrow [\Downarrow]$ and $[\Downarrow] = [\Leftrightarrow] \Rightarrow [\Downarrow]$. The ambiguity in this example arises due to the fact that in propositional logic it is ambiguous as to whether a valid statement has the precedent being true or false. All one can say about the validity of a statement is that if it is false (negatively valuated), that the consequent is false while the precedent is true. There however are two operators, \Leftrightarrow and \Downarrow by which CM's can be decomposed uniquely. That is, letting $\Phi = \Downarrow$ or $\Phi = \Leftrightarrow$, then in the expression $[\Theta_1] \Phi [\Theta_2] = [\Theta_3]$ if we know any two of the 2×2 CM's $[\Theta_1]$, $[\Theta_2]$, or $[\Theta_3]$ then the third is unique.

3.2.2 Quotient

Another useful application of correspondence matrices is the ability to quotient one logical expression by another. Let $\mathcal{L}_1 = X \Theta_1 Y$ and $\mathcal{L}_2 = Y \Theta_2 Z$ be two logical expressions. We say that $\mathcal{L}_1 \setminus \mathcal{L}_2$ is \mathcal{L}_1 being *quotiented* by \mathcal{L}_2 and

that if \mathcal{L}_1 and \mathcal{L}_2 share the same feature (positive entry), then this feature is quotiented out. This idea of quotienting out features between \mathcal{L}_1 and \mathcal{L}_2 is simplified as the setminus operator $\setminus \equiv \wedge \neg$ where $\wedge \neg$ is defined to be itself a logical operator where $\mathcal{L}_1 \wedge \neg \mathcal{L}_2 = \mathcal{L}_1 \wedge \neg(\mathcal{L}_2)$. Letting $\mathcal{L}_1 = X\Theta_1Y$ and $\mathcal{L}_2 = X\Theta_2Y$ we have the relationship;

$$\mathcal{L}_1 \setminus \mathcal{L}_2 = X\Theta_1Y \setminus X\Theta_2Y = \langle X|[\Theta_1]|Y \rangle \setminus \langle X|[\Theta_2]|Y \rangle = \langle X|[\Theta_1] \setminus [\Theta_2]|Y \rangle$$

where

$$[\Theta_1] \setminus [\Theta_2] = \begin{bmatrix} \Theta_{1,11} \wedge \neg\Theta_{2,11} & \Theta_{1,12} \wedge \neg\Theta_{2,12} \\ \Theta_{1,21} \wedge \neg\Theta_{2,21} & \Theta_{1,22} \wedge \neg\Theta_{2,22} \end{bmatrix} = [\Theta_1] \wedge \neg[\Theta_2]$$

This quotient of logical expressions gives a sense of how different one logical expression is to another. Note that we can also perform this computation using the modulo operator:

$$[\Theta_1] \text{ mod } [\Theta_2] = \begin{bmatrix} \Theta_{1,11} \text{ mod } \Theta_{2,11} & \Theta_{1,12} \text{ mod } \Theta_{2,12} \\ \Theta_{1,21} \text{ mod } \Theta_{2,21} & \Theta_{1,22} \text{ mod } \Theta_{2,22} \end{bmatrix}$$

where we find that $\wedge \neg \equiv \text{mod}$.

We can verify that $\mathcal{L}_1 \setminus \mathcal{L}_2 = \langle X|[\Theta_1] \setminus [\Theta_2]|Y \rangle$ by first noting that $\mathcal{L}_1 = \Theta_{1,ij}X_iY_j$ and $\mathcal{L}_2 = \Theta_{2,ij}X_iY_j$ hence $\mathcal{L}_1 \setminus \mathcal{L}_2 = \Theta_{1,ij}X_iY_j \wedge \neg(\Theta_{2,ij}X_iY_j)$. By rule 2. [3.1], $\neg(\Theta_{2,ij}X_iY_j) = \neg(\Theta_{2,ij})X_iY_j$ which gives us the full expression: $\Theta_{1,ij}X_iY_j \neg\Theta_{2,ij}$ and gives that: $X_i\Theta_{1,ij}\neg\Theta_{2,ij}Y_j = \langle X|[\Theta_1] \wedge [\neg\Theta_2]|Y \rangle$. We then have our result:

$$\langle X|[\Theta_1]|Y \rangle \wedge \langle X|[\neg\Theta_2]|Y \rangle = \langle X|[\Theta_1] \setminus [\Theta_2]|Y \rangle$$

As an example consider the following quotient of logical expressions $X \vee Y$ by $X \wedge Y$; $(X \vee Y) \setminus (X \wedge Y) = X \updownarrow Y$ via operations on the bra-ket formulation:

$$\langle X|[\vee]|Y \rangle \setminus \langle X|[\wedge]|Y \rangle = \langle X|[\vee]|Y \rangle \wedge \neg \langle X|[\wedge]|Y \rangle = \langle X|[\vee] \wedge [\neg\wedge]|Y \rangle = \langle X|[\updownarrow]|Y \rangle$$

in which we see that $[\vee] \wedge [\neg\wedge] = [\updownarrow]$ and that $[\vee]$ and $[\wedge]$ have a $[\updownarrow]$ difference from one another, as seen previously $[\vee] = [\wedge] \updownarrow [\updownarrow]$.

Quotienting is important as it shows how CM's and their corresponding logical operators are related. Furthermore it gives a way to decompose the logical operator in to other logical operators.

3.2.3 Composition

We had seen the ability for logical decomposition in 3.2.1; one may ask then, how we might combine logical expressions? This leads us to one of the biggest findings in the paper; that if the left and right operands are the same for multiple logical expressions, then the expressions can be combined solely by combining their correspondence matrices!

Let X and Y be two logical expressions and Θ_1, Θ_2, Φ be logical operators, we find:

$$\langle X|[\Theta_1]|Y \rangle \Phi \langle X|[\Theta_2]|Y \rangle = \langle X|[\Theta_1]\Phi[\Theta_2]|Y \rangle$$

To derive this, we first transform the left hand side of the expression as follows:

$$\begin{aligned} \langle X|[\Theta_1]|Y \rangle \Phi \langle X|[\Theta_2]|Y \rangle &= \langle (X_i \Theta_{1,ij} Y_j) | [\Phi] | (X_k \Theta_{2,kl} Y_l) \rangle \\ &= (X_i \Theta_{1,ij} Y_j)_m \Phi_{mn} (X_k \Theta_{2,kl} Y_l)_n \end{aligned}$$

where $m, n \in \{1, 2\}$. The trick to combining the terms lies in the observation in which $\neg \langle X|[\Theta_p]|Y \rangle = \langle X|[\neg \Theta_p]|Y \rangle$, $p \in \{1, 2\}$ by rule 2. [3.1]. We then have that $\langle X|[\neg \Theta_p]|Y \rangle = X_i (\Theta_{p,ij})_2 Y_j$ where $(\Theta_{p,ij})_2 = \neg(\Theta_{p,ij})$. Remembering that $\Phi_{mn}, \Theta_{p,ij} \in \{0, 1\}$ for all indices and that \wedge is commutative we have:

$$\begin{aligned} (X_i (\Theta_{1,ij})_m Y_j) \Phi_{mn} (X_k (\Theta_{2,kl})_n Y_l) &= X_i X_k ((\Theta_{1,ij})_m Y_j \Phi_{mn} (\Theta_{2,kl})_n Y_l) \delta_{ik} \delta_{jl} \\ &= X_i ((\Theta_{1,ij})_m \Phi_{mn} (\Theta_{2,ij})_n) Y_j \end{aligned}$$

as we can factor out like variables over XOR sums. The resulting expression can be re-written as $X_i (\Theta_{1,ij} | [\Phi] | \Theta_{2,ij}) Y_j = X_i (\Theta_{1,ij} \Phi \Theta_{2,ij}) Y_j$ in which $\Theta_{1,ij} \Phi \Theta_{2,ij} = (\Theta_1 \Phi \Theta_2)_{ij}$, we then have:

$$X_i (\Theta_1 \Phi \Theta_2)_{ij} Y_j = \langle X|[(\Theta_1 \Phi \Theta_2)]|Y \rangle = \langle X|[[\Theta_1]\Phi[\Theta_2]]|Y \rangle$$

hence the result. This completes the proof and shows that, for logical expressions consisting of the same two logical variables as their respective left and right hand operands, combining or decomposing their corresponding logical expressions becomes a matter of operating solely on the operators!

To see examples of this amazing result consider the logical expression $\mathcal{L}_1 = (X \Rightarrow Y) \Downarrow (X \vee Y)$. Writing \mathcal{L} in CM form and performing the computation we find:

$$\mathcal{L}_1 = \langle X|[\Rightarrow]|Y \rangle \Downarrow \langle X|[\vee]|Y \rangle = \langle X|[[\Rightarrow] \Downarrow [\vee]]|Y \rangle = \langle X|[\neg L]|Y \rangle = \neg X$$

If sub-expressions are not in the same form, applying rules in [3.1] will help us get an expression in the proper form in which the bra-ket operands are the same in all the sub-expressions. For example, similar to \mathcal{L}_1 above let $\mathcal{L}_2 = (X \Rightarrow \neg Y) \Downarrow (X \vee Y)$. We use rule 3 to format $(X \Rightarrow \neg Y)$ and find that $\langle X|[\Theta]|Y \rangle = \langle X|[\Theta]|Y \rangle^{T+90^\circ}$ which operating on the CM we find $[\Rightarrow]^{T+90^\circ} = [\neg]$; the logical expression transformation then gives that $\langle X|[\Rightarrow]|Y \rangle = \langle X|[\neg]|Y \rangle$. We can then combine operators from the sub-expressions and find:

$$\langle X|[\neg]|Y \rangle \Downarrow \langle X|[\vee]|Y \rangle = \langle X|[[\Rightarrow] \Downarrow [\vee]]|Y \rangle = \langle X|[\Leftrightarrow]|Y \rangle = X \Leftrightarrow Y$$

For more than two logical variables we can still combine operators, though not all together. To see this consider the expression $\mathcal{L} = (\mathcal{L}_1 \Phi_1 \mathcal{L}_2) \Upsilon (\mathcal{L}_3 \Phi_2 \mathcal{L}_4)$ where $\mathcal{L}_1, \dots, \mathcal{L}_4$ are logical sub-expressions bound by the operators Φ_1, Φ_2 and Υ . We can re-write \mathcal{L} in the corresponding CM form and find:

$$\mathcal{L} = \langle \mathcal{L}_1 | [\Phi_1] | \mathcal{L}_2 \rangle \Upsilon \langle \mathcal{L}_3 | [\Phi_2] | \mathcal{L}_4 \rangle = \Upsilon_{rs} \langle \mathcal{L}_1 | [\Phi_1] | \mathcal{L}_2 \rangle_r \langle \mathcal{L}_3 | [\Phi_2] | \mathcal{L}_4 \rangle_s$$

Applying rule 2 we find; $\langle \mathcal{L}_1 | [\Phi_1] | \mathcal{L}_2 \rangle_r = \langle \mathcal{L}_1 | [\Phi_1]_r | \mathcal{L}_2 \rangle = (\Phi_{1,ij})_r (\mathcal{L}_1)_i (\mathcal{L}_2)_j$ and $\langle \mathcal{L}_3 | [\Phi_2] | \mathcal{L}_4 \rangle_s = (\Phi_{2,ij})_s (\mathcal{L}_3)_k (\mathcal{L}_4)_l$ where we employ that $(\Phi_1)_{r,ij} = (\Phi_{1,ij})_r$ and that $(\Phi_2)_{s,kl} = (\Phi_{2,kl})_s$ [3.3]. We have the resulting expression:

$$\Upsilon_{rs} (\Phi_{1,ij})_r (\Phi_{2,kl})_s (\mathcal{L}_1)_i (\mathcal{L}_3)_k (\mathcal{L}_2)_j (\mathcal{L}_4)_l$$

Next, let $\mathcal{L}_1 = W\Theta_1X$, $\mathcal{L}_2 = Y\Theta_2Z$, $\mathcal{L}_3 = W\Theta_3X$, and $\mathcal{L}_4 = Y\Theta_4Z$, we can combine particular sub-expressions and find equivalent expression representations: $(\mathcal{L}_1)_i (\mathcal{L}_3)_k = \langle W | [\Theta_1] | X \rangle_i \langle W | [\Theta_3] | X \rangle_k = \langle W | [\Theta_1]_i | X \rangle \langle W | [\Theta_3]_k | X \rangle = \langle W | [[\Theta_1]_i \wedge [\Theta_3]_k] | X \rangle = \langle W | [[\Theta_1]_i [\Theta_3]_k] | X \rangle$. Following the same process for $(\mathcal{L}_2)_j (\mathcal{L}_4)_l$ and combining, we find the full expression:

$$\langle \Phi_{1,ij} | [\Upsilon] | \Phi_{2,kl} \rangle \langle W | [[\Theta_1]_i [\Theta_3]_k] | X \rangle \langle Y | [[\Theta_2]_j [\Theta_4]_l] | Z \rangle$$

in which $[\Upsilon]$ takes as operands $\langle \Phi_{1,ij} |$ and $| \Phi_{2,kl} \rangle$. One significance is that, depending on the result of operating by $[\Upsilon]$, one may not need to perform any further computations. For example, if $\langle \Phi_{1,ij} | [\Upsilon] | \Phi_{2,kl} \rangle = 0$ then one may permute to the next set of index values. Another significance of the logical expression in the above form shows that we are able to operate on logical operators themselves! We will continue to examine the combining of expressions with 4 or more logical variables in 5.2.

3.3 Compliment Operators

As is seen in figure 1, there are many CM's corresponding to logical operators that are not commonly seen in logic. The success of correspondence matrices lies in that there is a complete set of 16 CM's which correspond to unique logical operators, each of which can be negated and corresponds to in a unique operator. Possibly the most impressive result being that the negation of a logical expression can be found by negating the operator!

Proving that the negation of a logical expression $\mathcal{L} = X\Theta Y$ gives rise to a unique negated operator; In one direction, starting with $\neg \langle X | [\Theta] | Y \rangle = (X_a \Theta_{ab} Y_b)_2$ which, by 3.1 is equivalent to $(X_a (\Theta_{ab})_2 Y_b) = \langle X | [-\Theta] | Y \rangle$. In the other direction to prove that $\neg \Theta$ is the compliment of Θ ; as $\langle X | [\Theta]_2 | Y \rangle = \langle X | [\ominus] \setminus [\Theta] | Y \rangle = \langle X | [\ominus] | Y \rangle \wedge \neg \langle X | [\Theta] | Y \rangle = (X_a \Theta_{ab} Y_b)_2 = \langle X | [\Theta] | Y \rangle_2$ by 3.2.2. This shows that $\neg \langle X | [\Theta] | Y \rangle = \langle X | [-\Theta] | Y \rangle$.

The negation of any CM $[\Theta]$ will be called its *compliment* CM and will be denoted by $\neg[\Theta]$ or $[-\Theta]$, the *compliment operator* to Θ is denoted by $\neg\Theta$ or $(\Theta)_2$. One can easily verify that $\neg(\neg[\Theta]) = [\Theta]$. The compliment operators are shown in figure 2 as compliment pairs $\Theta, \neg\Theta$ where Θ is compliment to $\neg\Theta$, and $\neg\Theta$ is compliment to $\neg(\neg\Theta) = \Theta$. As Θ_{ij} represents the element with the index ij in $[\Theta]$, we will represent the negation of the Θ_{ij} 'th element by $\neg(\Theta_{ij})$ or $(\Theta_{ij})_2$ and as $\neg(\Theta_{ij})$ negates each of the elements Θ_{ij} in the CM $[-\Theta]$ each of the negated

Logical and compliment Operators			
$\oplus, 0$	$\leftrightarrow, \updownarrow$	$\wedge, \neg\wedge$	$\vee, \neg\vee$
\Rightarrow, \uparrow	\Leftarrow, \downarrow	$R, \neg R$	$L, \neg L$

Figure 2: The set of eight operator and compliment operator pairs

elements are elements of the negated operator; $\neg(\Theta_{ij}) = (\neg\Theta_{ij})$. More precisely $(\Theta_{ij})_2 = (\Theta)_{2,ij}$ as negating each $\Theta_{ij} \in [\Theta]$ gives us that $\neg(\Theta_{ij}) \in \neg[\Theta] = [\neg\Theta]$ which consists of the elements $(\neg\Theta)_{ij} = (\Theta)_{2,ij}$.

For a familiar example showing an application of DeMorgans law we find that $\neg(X \wedge Y) = \neg\langle X | [\wedge] | Y \rangle = \langle X | [\neg\wedge] | Y \rangle$ which is the XOR composition of $X_1(\neg\wedge_{11})Y_1 \updownarrow X_1(\neg\wedge_{12})Y_2 \updownarrow X_2(\neg\wedge_{21})Y_1 \updownarrow X_2(\neg\wedge_{22})Y_2$. As $(\neg\wedge_{11}) = 0$ and $(\neg\wedge_{12}) = (\neg\wedge_{21}) = (\neg\wedge_{22}) = 1$ we have $X \neg Y \updownarrow \neg XY \updownarrow \neg X \neg Y = \neg X \vee \neg Y$.

4 Measurement

Measurement is the process in which an action on a system (that which is measured) gives a quality (the resulting measurement) about the system. In the quantum mechanical case it is used to give an amount of an observable such as spin, velocity, or in the case of Schrödinger’s cat^[6], life or death. In the logical case measurement reveals the truth of a system, an observable being something such as a valuation of the logical expression being measured.

In [3] we saw the derivation of $[\Theta]$, a correspondence matrix representation of a logical operator Θ and how it acts or operates on the bra-ket logical state vectors as operands. The action of CM’s as well as the logical matrices (LM’s) [4.1] can be viewed as ‘measurements’. More precisely, a matrix operates or *measures* one or more operands and the action of operating on or *measuring* operands is called a *measurement*. For example, the logical expression $\mathcal{L} = X\Theta Y$ can be seen as the measurement of the logical state-vectors $\langle X |$ and $| Y \rangle$ by the CM $[\Theta]$ in the equivalent representation in which $\mathcal{L} = \langle X | [\Theta] | Y \rangle$.

We will not pursue the quantum computational nature of CM’s and LM’s in depth, one however should note the similarities between quantum mechanical type measurements and logical measurements. First in that the valuation of a logical expression is a measurement, in particular; a measurement of an expressions truth. Before valuating a logical expression \mathcal{L} we do not know whether it is true or false, we might view this as a superposition^[7] of possible valuations. Upon measurement we then have a collapse^[8] to one possible valuation. Though probability densities are not incorporated, the action of a linear CM on a pair of logical state-vectors gives a XOR decomposition in which the positive valuation of one component leaves all others negatively valuated. This is a quantization of truth, that is; upon measurement one can not have any logical expression be

both true and false simultaneously. We might then view logical operators acting on logical expressions as linear measurements which reduce the possibilities of the computed expression.

4.1 Logical Measurement Matrices (LM's)

We can generalize the idea of correspondence matrices to the notion of a 'logical measurement matrix' or LM in which we find that CM's are particular valuations of. LM's are also important as seen in section 4 as the action (measurement) of a LM on a set of operands gives information about the operands relationship with one-another.

In short, a correspondence matrix is a positive valuation of a logical matrix which is itself a measurement operator. Letting $[\mathcal{M}_{X\Theta Y}] = \Theta_{ij}|X_i\rangle\langle Y_j|$ be a LM of logical expressions X and Y we can find the unique CM $[\Theta]$:

$$[\Theta] = \mathcal{V}_T([\mathcal{M}_{X\Theta Y}]) = \begin{bmatrix} \mathcal{V}_T(X)\Theta\mathcal{V}_T(Y) & \mathcal{V}_T(X)\Theta\mathcal{V}_T(\neg Y) \\ \mathcal{V}_T(\neg X)\Theta\mathcal{V}_T(Y) & \mathcal{V}_T(\neg X)\Theta\mathcal{V}_T(\neg Y) \end{bmatrix} = \Theta_{ij}[\mathcal{M}_{\mathcal{V}_T(X_i)\mathcal{V}_T(Y_j)}]$$

as $\mathcal{V}_T(X\Theta Y) = \mathcal{V}_T(X)\Theta\mathcal{V}_T(Y)$ is applied element-wise to $[\Theta]$. For an example; letting $\Theta = \Leftrightarrow$ and applying the positive assignments to all X and Y remembering that for example $\mathcal{V}_T(\neg X) = \neg\mathcal{V}_T(X) = 0$ we have $\mathcal{V}_T([\mathcal{M}_{X\Leftrightarrow Y}]) = [\Leftrightarrow]$.

4.1.1 Two Variable LM's

Looking more in depth we start by deriving LM's of two variables. Letting $\langle X_i|$ and $|Y_j\rangle$ be the bra and ket state vectors of the logical expressions X_i and Y_j respectively where $i, j \in \{1, 2\}$ we have that corresponding to the logical expression $\mathcal{L} = X_i Y_j$ we can *prepare* a unique *logical matrix*:

$$[\mathcal{M}_{X_i Y_j}] = (\langle X_i|^T)(|Y_j\rangle^T) = |X_i\rangle\langle Y_j|$$

which we will also refer to as a *logical measurement matrix* (LM). We say that $[\mathcal{M}_{X_i Y_j}]$ operates on or measures $\langle X_i|$ and $|Y_j\rangle$ by treating them as operands of $[\mathcal{M}_{X_i Y_j}]$ and that the measurement is denoted: $\langle X_i|[\mathcal{M}_{X_i Y_j}]|Y_j\rangle$. Applying the four possibilities for i and j and the corresponding negations i.e. $|Y\rangle_2 = \neg|Y\rangle = |\neg Y\rangle$ we have the four 2×2 *logical base matrices*:

$$[\mathcal{M}_{X_1 Y_1}] = [\mathcal{M}_{XY}], [\mathcal{M}_{X_1 Y_2}] = [\mathcal{M}_{X\neg Y}], [\mathcal{M}_{X_2 Y_1}] = [\mathcal{M}_{\neg X Y}], [\mathcal{M}_{X_2 Y_2}] = [\mathcal{M}_{\neg X \neg Y}]$$

where;

$$[\mathcal{M}_{XY}] = |X\rangle\langle Y| = \begin{bmatrix} XY & X\neg Y \\ \neg XY & \neg X\neg Y \end{bmatrix}, [\mathcal{M}_{X\neg Y}] = |\neg X\rangle\langle \neg Y| = \begin{bmatrix} \neg X\neg Y & \neg XY \\ X\neg Y & XY \end{bmatrix}$$

$$[\mathcal{M}_{X\neg Y}] = |X\rangle\langle \neg Y| = \begin{bmatrix} X\neg Y & XY \\ \neg X\neg Y & \neg XY \end{bmatrix}, [\mathcal{M}_{\neg X Y}] = |\neg X\rangle\langle Y| = \begin{bmatrix} \neg XY & \neg X\neg Y \\ XY & X\neg Y \end{bmatrix}$$

These four matrices are base matrices as every LM can be written as the XOR sum of some combination of these matrices and each of these four LM's can not be reduced to a XOR sum of a lesser number of matrices. Using the

same transformation rules for CM's [3.1] we can see the relationship between the above base via translations, rotations, and combinations thereof; for example $[\mathcal{M}_{XY}]^{T+90^\circ} = [\mathcal{M}_{X\bar{Y}}]$, where the rotation is clockwise.

We can generalize the above base LM's and include the logical operator, that is; for $\mathcal{L} = X\Theta Y$ we can build a logical measurement matrix $[\mathcal{M}_{X\Theta Y}]$ via the XOR sum of base LM's:

$$\Theta_{jk}[\mathcal{M}_{X_j Y_k}] = \Theta_{11}[\mathcal{M}_{X_1 Y_1}] \Downarrow \Theta_{12}[\mathcal{M}_{X_1 Y_2}] \Downarrow \Theta_{21}[\mathcal{M}_{X_2 Y_1}] \Downarrow \Theta_{22}[\mathcal{M}_{X_2 Y_2}] = [\mathcal{M}_{X\Theta Y}]$$

where Θ_{jk} is the corresponding element indexed in $[\Theta]$. For example, to find the LM $[\mathcal{M}_{X\leftrightarrow Y}]$ we first note that for $[\Theta] = [\leftrightarrow]$ that $\leftrightarrow_{11} = \leftrightarrow_{22} = 1$ and $\leftrightarrow_{12} = \leftrightarrow_{21} = 0$ hence we have the XOR sum:

$$\begin{aligned} [\mathcal{M}_{X\leftrightarrow Y}] &= \leftrightarrow_{11}[\mathcal{M}_{X_1 Y_1}] \Downarrow \leftrightarrow_{12}[\mathcal{M}_{X_1 Y_2}] \Downarrow \leftrightarrow_{21}[\mathcal{M}_{X_2 Y_1}] \Downarrow \leftrightarrow_{22}[\mathcal{M}_{X_2 Y_2}] \\ &= [\mathcal{M}_{X_1 Y_1}] \Downarrow [\mathcal{M}_{X_2 Y_2}] = [\mathcal{M}_{XY}] \Downarrow [\mathcal{M}_{\bar{X}\bar{Y}}] = \begin{bmatrix} X \leftrightarrow Y & X \Downarrow Y \\ X \Downarrow Y & X \leftrightarrow Y \end{bmatrix} \end{aligned}$$

In general we will leave out the terms in which $\Theta_{ij} = 0$. Extending the above example, we could use the LM $[\mathcal{M}_{X\leftrightarrow Y}]$ to construct the LM $[\mathcal{M}_{X\Rightarrow Y}]$:

$$[\mathcal{M}_{X\leftrightarrow Y}] \Downarrow [\mathcal{M}_{\bar{X}Y}] = [\mathcal{M}_{X\Rightarrow Y}] = \begin{bmatrix} X \Rightarrow Y & \bar{X} \vee \bar{Y} \\ X \vee Y & X \Leftarrow Y \end{bmatrix}$$

The generalized LM matrix representation is:

$$[\mathcal{M}_{X\Theta Y}] = \Theta_{ij}|X_i\rangle\langle Y_j| = \begin{bmatrix} X_i \Theta_{ij} Y_j & X_i \Theta_{ij} \bar{Y}_j \\ \bar{X}_i \Theta_{ij} Y_j & \bar{X}_i \Theta_{ij} \bar{Y}_j \end{bmatrix}$$

4.1.2 Single Variable LM's

In this paper thus far we have seen logical expressions of two and four variables. It is insightful to see how to build LM's of single variables and how multi-variable logical expressions can be built from them as well as the asymmetries introduced from doing so. Similar to multi-variables logical expressions, LM's can be XOR'ed together so as to remove variables as well as be factored into conjunctions of LM's of a single variable, these facts however are built on the foundation of treating logical expressions as vectors [2.2].

To start, we can see that it is simple to *expand* logical expressions of a single variable by other variables and in doing so create a multi-variable logical expression:

$$Y = Y(X_1 \Downarrow \bar{X}_1) = Y(X_1 \Downarrow \bar{X}_1) \cdots (X_n \Downarrow \bar{X}_n)$$

We see however that by expanding a logical variable in terms of other variables we have introduced an asymmetry i.e. $Y = (X \Downarrow \bar{X})Y = Y(X \Downarrow \bar{X})$. That asymmetry in LM's makes a difference can be seen simply as $[\mathcal{M}_{XY}] = |X\rangle\langle Y| \neq |Y\rangle\langle X| = [\mathcal{M}_{YX}]$. We can see that the multi-variable expanding of LM's are analogous to the logical expansions above:

$$[\mathcal{M}_{XY}] \Downarrow [\mathcal{M}_{\neg XY}] = (|X\rangle \Downarrow |\neg X\rangle)\langle Y| = |T\rangle\langle Y| = [\mathcal{M}_{Y_R}] = \begin{bmatrix} Y & \neg Y \\ Y & \neg Y \end{bmatrix}$$

and

$$[\mathcal{M}_{YX}] \Downarrow [\mathcal{M}_{Y\neg X}] = |Y\rangle(\langle X| \Downarrow \langle \neg X|) = |Y\rangle\langle T| = [\mathcal{M}_{Y_L}] = \begin{bmatrix} Y & Y \\ \neg Y & \neg Y \end{bmatrix}$$

where $|X\rangle \Downarrow |\neg X\rangle = |T\rangle = [1 \ 1]^T$ is the *tautological statevector* as it is true in both of its indices. We use the above asymmetric matrices to derive the symmetric LM of a single variable:

$$[\mathcal{M}_Y] = |Y\rangle\langle Y| = (|Y\rangle\langle T|)(\langle T|Y\rangle) = [\mathcal{M}_{Y_L}][\mathcal{M}_{Y_R}]$$

It can be proven directly that we can factor two variable expressions in to their asymmetric factors: $[\mathcal{M}_{XY}] = ([\mathcal{M}_{X_L}][\mathcal{M}_{Y_R}]) = [\mathcal{M}_{X_L}][\mathcal{M}_{Y_R}]$. We generalize this result and find that that for the LM corresponding to the logical expression $X\Theta Y$ we have:

$$[\mathcal{M}_{X\Theta Y}] = \Theta_{ij}[\mathcal{M}_{X_i Y_j}] = \Theta_{ij}[\mathcal{M}_{X_{i_L}}][\mathcal{M}_{Y_{j_R}}] = \Theta_{ij}[\mathcal{M}_{X_L}]_i[\mathcal{M}_{Y_R}]_j$$

that $[\mathcal{M}_{X_{i_L}}] = [\mathcal{M}_{X_L}]_i$ can be seen as negation is applied element-wise.

If a LM can be written as a conjunction of multiple LM's which do not share a variable then the matrix is said to be *seperable* and we say that we can separate the LM into *factors*. For example $[\mathcal{M}_{X_L}]$ and $[\mathcal{M}_{Y_R}]$ are factors of $[\mathcal{M}_{XY}]$. The ability to break apart LM's of conjunctive expressions such as XY into factors is useful as it shows that we can break apart some multi-variable LM's in to a conjunction of LM's with a single variable.

4.1.3 LM Properties

As might be expected we find that the same computations such as LM transformations, quotienting, and composition can be performed on LM's in the same way as was done with CM's. To show these properties first recall that $X_i\Theta_{ij}Y_j$ has an equivalent representation as $\langle X|[\Theta]|Y\rangle$ [3]. Using this we can apply the equivalent bra-ket LM representations and find:

$$[\mathcal{M}_{X\Theta Y}] = \begin{bmatrix} X_i\Theta_{ij}Y_j & X_i\Theta_{ij}\neg Y_j \\ \neg X_i\Theta_{ij}Y_j & \neg X_i\Theta_{ij}\neg Y_j \end{bmatrix} = \begin{bmatrix} \langle X|[\Theta]|Y\rangle & \langle X|[\Theta]|\neg Y\rangle \\ \langle \neg X|[\Theta]|Y\rangle & \langle \neg X|[\Theta]|\neg Y\rangle \end{bmatrix}$$

We then find, by rule 2. of transforming CM's i.e. $\neg\langle X|[\Theta]|Y\rangle = \langle X|[\neg\Theta]|Y\rangle$, that:

$$\neg[\mathcal{M}_{X\Theta Y}] = \begin{bmatrix} \langle X|[\neg\Theta]|Y\rangle & \langle X|[\neg\Theta]|\neg Y\rangle \\ \langle \neg X|[\neg\Theta]|Y\rangle & \langle \neg X|[\neg\Theta]|\neg Y\rangle \end{bmatrix} = \neg\Theta_{ij}|X_i\rangle\langle Y_j| = [\mathcal{M}_{X(\neg\Theta)Y}]$$

in which we can see that it doesn't matter if \neg is applied to the whole LM or just to Θ as $\neg(\Theta_{ij}[\mathcal{M}_{X_i Y_j}]) = (\neg\Theta_{ij})[\mathcal{M}_{X_i Y_j}]$.

We can then extend the notion of quotienting to logical matrices. To see this take the logical operators Θ_1 and Θ_2 for which we have the respective associated LM's $[\mathcal{M}_{X\Theta_1 Y}]$ and $[\mathcal{M}_{X\Theta_2 Y}]$. We can quotient the LM's and we find that:

$$[\mathcal{M}_{X\Theta_1 Y}] \setminus [\mathcal{M}_{X\Theta_2 Y}] = [\mathcal{M}_{X(\Theta_1 \setminus \Theta_2) Y}]$$

This relationship is proven by first remembering that as Θ_{ij} is a logical variable we have $\neg[\mathcal{M}_{X\Theta Y}] = \neg\Theta_{ij} \boxplus \neg[\mathcal{M}_{X_i Y_j}]$. We then find that the quotient $[\mathcal{M}_{X\Theta_1 Y}] \setminus [\mathcal{M}_{X\Theta_2 Y}] = \Theta_{1,ij} [\mathcal{M}_{X_i Y_j}] \wedge (\neg\Theta_{2,ij} \boxplus \neg[\mathcal{M}_{X_i Y_j}])$ which, as we can distribute over \wedge and \vee and that operations between LM's are performed between corresponding LM indices, we have the equivalent expression $(\Theta_{1,ij} \wedge \neg\Theta_{2,ij}) [\mathcal{M}_{X_i Y_j}]$. As $\Theta_{1,ij} \wedge \neg\Theta_{2,ij} = (\Theta_1 \setminus \Theta_2)_{ij}$ the result follows.

Similar to the combining of logical expressions by way of combining their respective CM's we can combine logical matrices and find that:

$$[\mathcal{M}_{X\Theta_1 Y}] \Phi [\mathcal{M}_{X\Theta_2 Y}] = [\mathcal{M}_{X(\Theta_1 \Phi \Theta_2) Y}]$$

To see this $[\mathcal{M}_{X\Theta_1 Y}] \Phi [\mathcal{M}_{X\Theta_2 Y}]$ can be written equivalently as:

$$\begin{bmatrix} \langle X | [\Theta_1] | Y \rangle & \langle X | [\Theta_1] | \neg Y \rangle \\ \langle \neg X | [\Theta_1] | Y \rangle & \langle \neg X | [\Theta_1] | \neg Y \rangle \end{bmatrix} \Phi \begin{bmatrix} \langle X | [\Theta_2] | Y \rangle & \langle X | [\Theta_2] | \neg Y \rangle \\ \langle \neg X | [\Theta_2] | Y \rangle & \langle \neg X | [\Theta_2] | \neg Y \rangle \end{bmatrix}$$

We can combine sub-expressions over Φ as operators can be combined i.e. $\langle X | [\Theta_1] | Y \rangle \Phi \langle X | [\Theta_2] | Y \rangle = \langle X | [\Theta_1 \Phi \Theta_2] | Y \rangle$ [3.1]; doing this for all indices we find:

$$\begin{bmatrix} \langle X | [\Theta_1 \Phi \Theta_2] | Y \rangle & \langle X | [\Theta_1 \Phi \Theta_2] | \neg Y \rangle \\ \langle \neg X | [\Theta_1 \Phi \Theta_2] | Y \rangle & \langle \neg X | [\Theta_1 \Phi \Theta_2] | \neg Y \rangle \end{bmatrix} = (\Theta_1 \Phi \Theta_2)_{ij} [\mathcal{M}_{X_i Y_j}] = [\mathcal{M}_{X(\Theta_1 \Phi \Theta_2) Y}]$$

which shows that we can combine LM's which share the same ordered sub-expressions.

4.1.4 Measurement using LM's

As was the case with correspondence matrices, logical matrices are built from the ground up to be measurement operators. Like CM's they measure the relationship between bra-ket state-vector pairs, however instead of returning an expression, the LM measures truth.

To see how measurement reveals truth, consider $\langle X_i | [\mathcal{M}_{X \Rightarrow Y}] | Y_j \rangle$ where $i, j \in \{1, 2\}$:

$\langle X [\mathcal{M}_{X \Rightarrow Y}] Y \rangle = 1$	$\langle X [\mathcal{M}_{X \Rightarrow Y}] \neg Y \rangle = 0$
$\langle \neg X [\mathcal{M}_{X \Rightarrow Y}] Y \rangle = 1$	$\langle \neg X [\mathcal{M}_{X \Rightarrow Y}] \neg Y \rangle = 1$

This measurement gives a more general measurement concerning the validity of a logical statement in which we can see that it is not valid to measure a logical statevector $\langle X |$ as a premise with a conclusion of $|\neg Y\rangle$ no matter the valuation of X and $\neg Y$.

Measuring logical state-vectors by a LM shows the logical operator's relationship explicitly:

$$\langle X_i | [\mathcal{M}_{X\Theta Y}] | Y_j \rangle = \Theta_{ij} \langle X_i | X_k \rangle \langle Y_l | Y_j \rangle \delta_{ik} \delta_{lj} = \Theta_{ij}$$

where $k, l \in \{1, 2\}$ and δ_{ab} is the Kronecker delta where $\delta_{ab} = 1$ if $a = b$ and 0 otherwise. We can then see that this type of measurement depends fully on the operator Θ . As an example, consider the LM $[\mathcal{M}_{X \Rightarrow Y}] = \Rightarrow_{kl}[\mathcal{M}_{X_k Y_l}]$, we have:

$$\langle X_i | [\mathcal{M}_{X \Rightarrow Y}] | Y_j \rangle = \Rightarrow_{ij} \langle X_i | X_k \rangle \langle Y_l | Y_j \rangle \delta_{ik} \delta_{lj} = \Rightarrow_{ij}$$

by remembering that $[\mathcal{M}_{X_k Y_l}] = |X_k\rangle\langle Y_l|$ and that as $\Rightarrow_{ij} \in \{0, 1\}$ we can factor it out.

We do not always have the result of the measurement being dependent solely on the logical operator, for example $\langle X | [\mathcal{M}_{X \Theta Y}] | Z \rangle = \Theta_{ij} \langle X | X_i \rangle \langle Y_j | Z \rangle = \Theta_{1j} \langle Y_j | Z \rangle$. More generally, in measuring the logical expressions \mathcal{L}_1 and \mathcal{L}_2 by the LM $[\mathcal{M}_{X \Theta Y}]$ we find that:

$$\mathcal{L} = \langle \mathcal{L}_1 | [\mathcal{M}_{X \Theta Y}] | \mathcal{L}_2 \rangle = \Theta_{rs} (\mathcal{L}_1 \Leftrightarrow X_r) \wedge (Y_s \Leftrightarrow \mathcal{L}_2)$$

or equivalently as $\Theta_{rs} \langle \mathcal{L}_1 | X_r \rangle \wedge \langle Y_s | \mathcal{L}_2 \rangle$. As, for example, the bra-ket $\langle \mathcal{L}_1 | X_r \rangle = \mathcal{L}_1 X_r \Leftrightarrow \neg \mathcal{L}_1 \neg X_r$ we have that the measured expression \mathcal{L}_1 is in a sense ‘polarized’ in respect to X_r that is; \mathcal{L} has a positive valuation depending on both $(\mathcal{L}_1 \Leftrightarrow X_r)$ and $(\mathcal{L}_2 \Leftrightarrow Y_s)$ being positively valued. Measurement by a LM gives a constraint on how logical sub-expressions can be related so as to allow the overall expression to be positively valued.

5 Higher Dimensional Matrices

We can create higher dimensional correspondence matrices in a manner similar to that of the 2×2 matrices. With the 2×2 LM’s there were only $2! = 2$ choices to create a logical matrix i.e. for logical expressions X and Y , only the LM’s $[\mathcal{M}_{XY}]$ or $[\mathcal{M}_{YX}]$ are possible. Things become more complex in the higher dimensional case in that for n logical variables there are $n!$ ways to create the $n \times n$ correspondence matrix. This section will derive a way to build higher dimensional LM’s and their corresponding CM’s.

5.1 4×4 Matrix Representation

Previously in section 4.1.1 we found the unique logical matrix $[\mathcal{M}_{AB}]$ corresponding to the logical expression $\mathcal{L} = A \wedge B$. We want embed elements of $[\mathcal{M}_{AB}]$ and the more generalized $[\mathcal{M}_{A \Theta B}]$ in to a larger matrix. To do so we start by defining a projection map p such that:

$$p : [\mathcal{M}_{AB}] \mapsto [[\mathcal{M}_A] \otimes [\mathcal{M}_B]]$$

in which $\forall e \in [\mathcal{M}_{AB}]$, $p(e) \in [[\mathcal{M}_A] \otimes [\mathcal{M}_B]]$ where $p(e) = e$. As both $[\mathcal{M}_A]$ and $[\mathcal{M}_B]$ are diagonal matrices, the map $p(e) = e$ gives elements on the diagonal in $[[\mathcal{M}_A] \otimes [\mathcal{M}_B]]$:

$$[\mathcal{M}_{AB}] = \begin{bmatrix} AB & A\bar{B} \\ \bar{A}B & \bar{A}\bar{B} \end{bmatrix} \xrightarrow{p} \begin{bmatrix} AB & 0 & 0 & 0 \\ 0 & A\bar{B} & 0 & 0 \\ 0 & 0 & \bar{A}B & 0 \\ 0 & 0 & 0 & \bar{A}\bar{B} \end{bmatrix} = [[\mathcal{M}_A] \otimes [\mathcal{M}_B]]$$

We can generalize this result to LM's of the form $[\mathcal{M}_{A\Phi B}] = \Phi_{tu}[\mathcal{M}_{A_t B_u}]$ where $t, u \in \{1, 2\}$ and find that for the logical expression $A\Phi B$:

$$\Phi_{tu}[\mathcal{M}_{A_t B_u}] \xrightarrow{p} \Phi_{tu}[[\mathcal{M}_{A_t}] \otimes [\mathcal{M}_{B_u}]]$$

We proceed to build a 4×4 LM representing a logical expression with four variables via the use of substitution. To do this consider the logical expression $\mathcal{L} = A \wedge B$ from above and let $A = W\Theta_1 X$ and $B = Y\Theta_2 Z$. As we can write $[\mathcal{M}_{W\Theta_1 X}] = \Theta_{1,ij}[\mathcal{M}_{W_i X_j}]$ and $[\mathcal{M}_{Y\Theta_2 Z}] = \Theta_{2,kl}[\mathcal{M}_{Y_k Z_l}]$ we have:

$$[[\mathcal{M}_A] \otimes [\mathcal{M}_B]] = [[\mathcal{M}_{W\Theta_1 X}] \otimes [\mathcal{M}_{Y\Theta_2 Z}]] = \Theta_{1,ij}\Theta_{2,kl} [[\mathcal{M}_{W_i X_j}] \otimes [\mathcal{M}_{Y_k Z_l}]]$$

which, as was the case with the two variable LM's, the operations $\Theta_{1,ij}$ and $\Theta_{2,kl}$, are binary and can be factored out of the tensor product. We then have that, like $[A \wedge B]$, the LM $[\mathcal{M}_{(W\Theta_1 X) \wedge (Y\Theta_2 Z)}]$ can be represented by a 4×4 logical matrix. The simplest example of a 4×4 LM is one in which $\Theta_1 = \Theta_2 = \wedge$ as this constrains the indices i, j, k, l to be 1. The resulting 4×4 matrix is:

$$[[\mathcal{M}_{WX}] \otimes [\mathcal{M}_{YZ}]] = \begin{bmatrix} WXYZ & WXY\bar{Z} & W\bar{X}YZ & W\bar{X}Y\bar{Z} \\ WX\bar{Y}Z & WX\bar{Y}\bar{Z} & W\bar{X}\bar{Y}Z & W\bar{X}\bar{Y}\bar{Z} \\ \bar{W}XYZ & \bar{W}XY\bar{Z} & \bar{W}\bar{X}YZ & \bar{W}\bar{X}Y\bar{Z} \\ \bar{W}X\bar{Y}Z & \bar{W}X\bar{Y}\bar{Z} & \bar{W}\bar{X}\bar{Y}Z & \bar{W}\bar{X}\bar{Y}\bar{Z} \end{bmatrix}$$

which is the LM corresponding to the logical expression $\mathcal{L} = WXYZ$.

The 4×4 LM just constructed is an example of only one of the $2^4 = 16$ possible base matrices for fixed Θ_1 and Θ_2 . The more general LM corresponding to the logical expression $\mathcal{L} = (W\Theta_1 X)\Phi(Y\Theta_2 Z)$ is:

$$[\mathcal{M}_{(W\Theta_1 X)\Phi(Y\Theta_2 Z)}] \xrightarrow{p} \Phi_{rs}[[\mathcal{M}_{W\Theta_1 X}]_r \otimes [\mathcal{M}_{Y\Theta_2 Z}]_s]$$

for the projection p as defined above. However as, for example, $[\mathcal{M}_{W\Theta_1 X}]_r = (\Theta_{1,ij})_r [W_i X_j]$ as seen by rule two in 3.1 we have the projection:

$$[\mathcal{M}_{(W\Theta_1 X)\Phi(Y\Theta_2 Z)}] \xrightarrow{p} \Phi_{rs}(\Theta_{1,ij})_r (\Theta_{2,kl})_s [[\mathcal{M}_{W_i X_j}] \otimes [\mathcal{M}_{Y_k Z_l}]]$$

where each $\Phi_{rs}, \Theta_{1,ij}, \Theta_{2,kl} \in \{0, 1\}$ and are commutative over conjunctions. We denote Φ as such to remind us that the sub-expressions $W\Theta_1 X$ and $Y\Theta_2 Z$ are computed first so as to remove ambiguity.

Consider the term $\Phi_{rs}(\Theta_{1,ij})_r (\Theta_{2,kl})_s$ in the above expression, this term has the equivalent representation:

$$\Phi_{rs}((\Theta_{1,ij})_r (\Theta_{2,kl})_s) = \langle (\Theta_{1,ij}) | [\Phi] | (\Theta_{2,kl}) \rangle$$

This shows that we can measure the relationship between operators! This is significant as, for a particular subscript valuation, the action of $[\Phi]$ on both $\langle \Theta_{1,ij} |$ and $|\Theta_{2,kl} \rangle$ tells us whether the LM $[[\mathcal{M}_{W_i X_j}] \otimes [\mathcal{M}_{Y_k Z_l}]]$ needs to be combined (XOR summed) with the other LM's in which the subscript permutations are such that $\mathcal{V}(\langle \Theta_{1,ij} | [\Phi] | \Theta_{2,kl} \rangle) \neq 0$. All LM's have a representation in which they can be conjoined or *modified* by some CM element e.g. $[\mathcal{M}_{X \Theta Y}] = \Theta_{ij} [\mathcal{M}_{X_i Y_j}]$ where $[\mathcal{M}_{X_i Y_j}]$ is modified by Θ_{ij} . We will call any expression which modifies a LM a *modifier*.

5.1.1 Measurement

As ordering is important for any logical expression hence all LM's, we must pick a consistent way in which the LM acts upon the left and right hand operands. To do this we first transform $[\mathcal{M}_{(W \Theta_1 X) \Phi (Y \Theta_2 Z)}]$ in to an equivalent representation. In pursuing this endeavor note the following:

$$|W\rangle|Y\rangle = \begin{bmatrix} W|Y\rangle \\ \neg W|Y\rangle \end{bmatrix} \text{ and } \langle Z|\langle X| = \begin{bmatrix} \langle Z|X\rangle \\ \langle Z|\neg X\rangle \end{bmatrix}^T$$

in which, as conjunction is commutative:

$$|W\rangle|Y\rangle \otimes \langle Z|\langle X| = \begin{bmatrix} WX|Y\rangle\langle Z| & W\neg X|Y\rangle\langle Z| \\ \neg WX|Y\rangle\langle Z| & \neg W\neg X|Y\rangle\langle Z| \end{bmatrix} = |W\rangle\langle X| \otimes |Y\rangle\langle Z|$$

We find the left hand operand to be $(|W\rangle|Y\rangle)^T = \langle Y|\langle W|$ and the right hand to be $(\langle Z|\langle X|)^T = |X\rangle|Z\rangle$ hence the overall measurement of the left and right operands by the LM is:

$$\langle Y|\langle W|[\mathcal{M}_{(W \Theta_1 X) \Phi (Y \Theta_2 Z)}]|X\rangle|Z\rangle$$

Finding the correspondence matrices from the logical ones is simply done by taking the positive valuation of each LM [4.1]. For the 4×4 matrix above we have;

$$\mathcal{V}_T[\mathcal{M}_{(W \Theta_1 X) \Phi (Y \Theta_2 Z)}] \xrightarrow{p} \langle \Theta_{1,ij} | [\Phi_{rs}] | \Theta_{2,kl} \rangle [\mathcal{V}_T([\mathcal{M}_{W_i X_j}]) \otimes \mathcal{V}_T([\mathcal{M}_{Y_k Z_l}])]$$

and find that modifying $\mathcal{V}_T([\mathcal{M}_{W \Theta_1 X}]_r \otimes [\mathcal{M}_{Y \Theta_2 Z}]_s)$ by $\Phi_{rs} \forall r, s$ allows the computation of 4×4 CM's. As an example of a 4×4 LM and the corresponding CM, consider the expression $\mathcal{L} = (W \Downarrow X) \Rightarrow (\neg Y \wedge Z)$. We first find the LM's:

$$[\mathcal{M}_{W \Downarrow X}] = \begin{bmatrix} W \Downarrow X & W \Leftrightarrow X \\ W \Leftrightarrow X & W \Downarrow X \end{bmatrix} \text{ and } [\mathcal{M}_{Y \Downarrow Z}] = \begin{bmatrix} \neg Y Z & \neg Y \neg Z \\ Y Z & Y \neg Z \end{bmatrix}$$

We can build the 4×4 logical matrix by taking the XOR sum relating to the operator \Rightarrow operating on $[\mathcal{M}_{W \Downarrow X}]$ and $[\mathcal{M}_{Y \Downarrow Z}]$:

$$\begin{aligned} [\mathcal{M}_{(W \Downarrow X) \Rightarrow (Y \Downarrow Z)}] &= \Rightarrow_{rs} [\mathcal{M}_{(W \Downarrow X)}]_r [\mathcal{M}_{(Y \Downarrow Z)}]_s \xrightarrow{p} \Rightarrow_{rs} [[\mathcal{M}_{(W \Downarrow X)}]_r \otimes [\mathcal{M}_{(Y \Downarrow Z)}]_s] \\ &= [[\mathcal{M}_{W \Downarrow X}] \otimes [\mathcal{M}_{Y \Downarrow Z}]] \Downarrow [\neg[\mathcal{M}_{W \Downarrow X}] \otimes [\mathcal{M}_{Y \Downarrow Z}]] \Downarrow [\neg[\mathcal{M}_{W \Downarrow X}] \otimes \neg[\mathcal{M}_{Y \Downarrow Z}]] \end{aligned}$$

in which the modifiers $\Rightarrow_{11} = \Rightarrow_{21} = \Rightarrow_{22} = 1$ and $\Rightarrow_{12} = 0$ have been applied and components with 0-modifiers have been removed. While we can find the CM from a positively valuated LM [4.1], we do not need to do so as a logical expression provides enough information.

To see how to construct the CM from the above logical expression we start by writing it in the equivalent form $\mathcal{L} = \langle W | [\Downarrow] | X \rangle \Rightarrow \langle Y | [\Downarrow] | Z \rangle$. As we have already chosen the order for the variables W, X, Y, Z we only need to compute the 4×4 matrix from the action of the modifier \Rightarrow on $[[\Downarrow] \otimes [\Downarrow]]$:

$$\Rightarrow_{rs} [[\Downarrow]_r \otimes [\Downarrow]_s] = [[\Downarrow]_1 \otimes [\Downarrow]_1] \Downarrow [[\Downarrow]_2 \otimes [\Downarrow]_1] \Downarrow [[\Downarrow]_2 \otimes [\Downarrow]_2] = \neg[[\Downarrow]_1 \otimes \neg[\Downarrow]_2]$$

where the right side of the equality is an equivalent computation done by noting that $(W \Downarrow X) \Rightarrow (Y \wedge \neg Z) = \neg((W \Downarrow Z) \wedge \neg(Y \wedge Z))$, we find:

$$\Rightarrow_{rs} [[\Downarrow]_r \otimes [\Downarrow]_s] = \mathcal{V}_T([\mathcal{M}_{(W \Downarrow X) \Rightarrow (Y \Downarrow Z)}]) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

which then measuring the operands $\langle Y | \langle W |$ and $| X \rangle | Z \rangle$ gives:

$$\langle Y | \langle W | \mathcal{V}_T[\mathcal{M}_{(W \Downarrow X) \Rightarrow (Y \Downarrow Z)}] | X \rangle | Z \rangle = (W \Downarrow X) \Rightarrow (\neg Y \wedge Z)$$

this makes sense as the CM $\mathcal{V}_T[\mathcal{M}_{(W \Downarrow X) \Rightarrow (Y \Downarrow Z)}]$ was constructed to measure logical state-vectors corresponding to variables W, X, Y, Z in a particular order. Like combinations of logical expressions with two variables, the true power of the higher dimensional CM's become evident when (de)composing logical expressions. An example of this can be seen in section 5.4.

5.2 Higher Dimensional Matrices

The ability to represent logical expressions as LM's is not limited to expressions of 4 or less logical variables. To see this, consider the logical expression $\mathcal{L} = ((X_1 \Theta_1 X_2) \Phi_1 (X_3 \Theta_2 X_4)) \Upsilon ((X_5 \Theta_3 X_6) \Phi_2 (X_7 \Theta_4 X_8))$; the corresponding LM is:

$$\Upsilon_{rs} (\Phi_{1,ij})_r (\Phi_{2,kl})_s (\Theta_{1,ab})_i (\Theta_{2,cd})_j (\Theta_{3,ef})_k (\Theta_{4,gh})_l \\ \otimes [[[\mathcal{M}_{(X_1)_a(X_2)_b}] \otimes [\mathcal{M}_{(X_3)_c(X_4)_d}] \otimes [[[\mathcal{M}_{(X_5)_e(X_6)_f}] \otimes [\mathcal{M}_{(X_7)_g(X_8)_h}]]]]]$$

While this expression has a lot of indices, it can be reduced to only eight; the same as the number of logical variables. To do this we employ the technique seen in section 3.3 such that for any logical operator Θ we have $(\Theta)_{t,uv} = (\Theta_{uv})_t$ where $t, u, v \in \{0, 1\}$. We can rearrange the operators and re-write combinations of operators in bra-ket form, for example as $(\Phi_{1,ij})_r = (\Phi_1)_{r,ij}$ we have $(\Theta_{1,ab})_i (\Phi_1)_{r,ij} (\Theta_{2,cd})_j = \langle \Theta_{1,ab} | [(\Phi_1)_r] | \Theta_{2,cd} \rangle$. After employing rule 2 [3.1] and finding that, for example, $\langle \Theta_{1,ab} | [(\Phi_1)_r] | \Theta_{2,cd} \rangle = \langle \Theta_{1,ab} | [\Phi_1] | \Theta_{2,cd} \rangle_r$ we can rewrite the above expression as:

$$\langle \langle \Theta_{1,ab} | [\Phi_1] | \Theta_{2,cd} \rangle | [\Upsilon] | \langle \Theta_{3,ef} | [\Phi_2] | \Theta_{4,gh} \rangle \rangle$$

$$[[[\mathcal{M}_{(X_1)_a(X_2)_b}] \otimes [\mathcal{M}_{(X_3)_c(X_4)_d}] \otimes [[\mathcal{M}_{(X_5)_e(X_6)_f}] \otimes [\mathcal{M}_{(X_7)_g(X_8)_h}]]]]$$

in which we can see that the modifier for the LM is completely composed of logical operators. While there are 2^8 possible 16×16 base LM's which may be XOR'ed together, we may fully remove any LM which has a modifier whose valuation is 0 thus reducing the overall computational time.

We build LM's of larger dimension so as to not have each indexed element of a smaller dimensional LM be a large logical expression. Using induction^[9] we can build larger LMs and show that any logical expression with 2^n logical sub-expressions (1) can be represented by a square $2^n \times 2^n$ LM and (2) all non-conjunctive operators can be pulled in to the LM modifier. First, we have shown by demonstration that for $n = 1$ and $n = 2$ ([4.1.1], and [5.1] respectively) that there are LM's satisfying the criteria (1) and (2) as stated. Assuming we can do this for the logical expression $\mathcal{L} = \mathcal{L}_1 \Theta_1 \mathcal{L}_2$ which can be represented by the $2^{n-1} \times 2^{n-1}$ LM $[\mathcal{M}_{\mathcal{L}_1 \Theta_1 \mathcal{L}_2}]$, we want to show that by appending another logical expression $X \Theta_2 Y$ over the operator Φ we can find a $2^n \times 2^n$ dimensional LM for $(\mathcal{L}_1 \Theta_1 \mathcal{L}_2) \Phi (X \Theta_2 Y)$ in which all non-conjunctive operators can be moved to the modifier. To do this, let $(X \Theta_2 Y)$ be a logical expression with only two variables X and Y ; we have the associated 2×2 LM and projection:

$$[\mathcal{M}_{(\mathcal{L}_1 \Theta_1 \mathcal{L}_2) \Phi (X \Theta_2 Y)}] \xrightarrow{p} \Phi_{uv} [[\mathcal{M}_{(\mathcal{L}_1 \Theta_1 \mathcal{L}_2)_u}] \otimes [\mathcal{M}_{(X \Theta_2 Y)_v}]]$$

As $[\mathcal{M}_{(\mathcal{L}_1 \Theta_1 \mathcal{L}_2)}] = (\Theta_1)_{ij} [\mathcal{M}_{(\mathcal{L}_1)_i (\mathcal{L}_2)_j}]$ and $[\mathcal{M}_{(X \Theta_2 Y)}] = (\Theta_2)_{kl} [\mathcal{M}_{X_k Y_l}]$ with that both $(\Theta_1)_{ij}_u$ and $(\Theta_2)_{kl}_v$ are binary values, we can factor them out in to the modifier proving (2). We have the result:

$$[\mathcal{M}_{(\mathcal{L}_1 \Theta_1 \mathcal{L}_2) \Phi (X \Theta_2 Y)}] \xrightarrow{p} \langle \Theta_1, ij | [\Phi] | \Theta_2, kl \rangle [[\mathcal{M}_{(\mathcal{L}_1)_i (\mathcal{L}_2)_j}] \otimes [\mathcal{M}_{X_k Y_l}]]$$

which as $Dim([\mathcal{M}_{(\mathcal{L}_1)_i (\mathcal{L}_2)_j}]) = 2^{n-1} \times 2^{n-1}$ and $Dim([\mathcal{M}_{X_k Y_l}]) = 2 \times 2$ we have that $Dim([\mathcal{M}_{(\mathcal{L}_1)_i (\mathcal{L}_2)_j}] \otimes [\mathcal{M}_{X_k Y_l}]) = 2^n \times 2^n$ thus proving (1).

It is not difficult to generalize the above proof to any logical expression of a finite number variables. To do so simply replace any logical variables which one would like removed with a 0 and the connecting binary operator with XOR. For example, to find the logical expression $\mathcal{L} = \mathcal{L}_1$ from $\mathcal{L} = \mathcal{L}_1 \Theta \mathcal{L}_2$ let $\Theta = \uparrow$ and $\mathcal{L}_2 = 0$, as $\mathcal{L}_1 \uparrow 0 = \mathcal{L}_1$ we have our result. Any finite LM can be constructed in a similar manner, we find that $[\mathcal{M}_{\mathcal{L}_1 \Theta \mathcal{L}_2}] = [\mathcal{M}_{\mathcal{L}_1 \uparrow 0}] = [\mathcal{M}_{\mathcal{L}_1}]$.

It is worth mentioning what $2^n \times 2^n$ basis matrices look like. For n logical expressions $\mathcal{L}_1 \cdots \mathcal{L}_n$ we find that for $\mathcal{L} = \mathcal{L}_1 \mathcal{L}_2 \cdots \mathcal{L}_n$, $n \geq 1$:

$$|\mathcal{L}_1\rangle \cdots |\mathcal{L}_{2i-1}\rangle = \otimes_{i=1}^{2n} |\mathcal{L}_{2i-1}\rangle \text{ and } \langle \mathcal{L}_{2i} | \cdots \langle \mathcal{L}_2 | = \otimes_{i=1}^{2n} \langle \mathcal{L}_{2i} |$$

we have the $2^n \times 2^n$ basis LM:

$$[\mathcal{M}_{\mathcal{L}_1 \cdots \mathcal{L}_n}] \xrightarrow{p} (\otimes_{i=1}^{2n} |\mathcal{L}_{2i-1}\rangle) (\otimes_{i=1}^{2n} \langle \mathcal{L}_{2i} |) = \otimes_{i=1}^{2n} |\mathcal{L}_{2i-1}\rangle \langle \mathcal{L}_{2i} | = \otimes_{i=1}^{2n} [\mathcal{M}_{\mathcal{L}_{(2i-1)} \mathcal{L}_{(2i)}}]$$

This however is only one of the possible 2^n basis matrices as there is a single possible non-zero modifier. The reason for this is that all operators between logical expressions are conjunctions.

We can measure these LM base matrices in the same way measurement was done for 4×4 LM's [5.1.1]. To do this, placing $\bigotimes_{i=1}^{2n}$ inside the bra and ket; $(\bigotimes_{i=1}^{2n} |\mathcal{L}_{2i-1}\rangle)^T = \langle \bigotimes_{i=1}^{2n} \mathcal{L}_{2i-1} |$ and $(\bigotimes_{i=1}^{2n} \langle \mathcal{L}_{2i} |)^T = | \bigotimes_{i=1}^{2n} \mathcal{L}_{2i} \rangle$:

$$\langle \bigotimes_{i=1}^{2n} \mathcal{L}_{2i-1} | \bigotimes_{i=1}^{2n} [\mathcal{M}_{\mathcal{L}_{(2i-1)}\mathcal{L}_{(2i)}}] | \bigotimes_{i=1}^{2n} \mathcal{L}_{2i} \rangle$$

5.3 Multivariable Computational Comparison

In this paper two different algorithms for computation have been developed. Both algorithms require the logical expression to be formatted in a particular fashion such that one can consistently measure the relationship between logical expressions [4.1.4]. The difference between the two algorithms is that one method (first method) does not continually build larger and larger measurement matrices while the other (second method) does. Notice that the order of operator measurement in both the algorithms, this will play an important role in how computation can be performed.

To see the difference in the computational methods consider logical expression \mathcal{L} with four variables where \mathcal{L} is written in the form:

$$\mathcal{L} = ((W\Theta_1 X)\Phi_1(Y\Theta_2 Z))\Upsilon((W\Theta_3 X)\Phi_2(Y\Theta_4 Z))$$

The first method utilizes the ability for composition of logical expressions based on their respective CM's [3.2.3]:

$$\mathcal{L} = \langle \Phi_{1,ij} | [\Upsilon] | \Phi_{2,kl} \rangle \langle W | [[\Theta_1]_i [\Theta_3]_k] | X \rangle \langle Y | [[\Theta_2]_j [\Theta_4]_l] | Z \rangle$$

This method computes by way of many 2×2 CM computations and has an advantage in that it allows the measurement on logical operators on both the left and right side of Υ which reveals false modifiers i.e. $\mathcal{V}(\langle \Phi_{1,ij} | [\Upsilon] | \Phi_{2,kl} \rangle) = 0$ thus lessening the number of overall computations.

In the second method for combining logical expressions of four variables, we combine the expressions 4×4 CM's for each $\mathcal{L}_1 = ((W\Theta_1 X)\Phi_1(Y\Theta_2 Z))$ and $\mathcal{L}_2 = ((W\Theta_3 X)\Phi_2(Y\Theta_4 Z))$ over Υ . The LM representation is:

$$\langle \langle \Theta_{1,ab} | [\Phi_1] | \Theta_{2,cd} \rangle | [\Upsilon] | \langle \Theta_{3,ab} | [\Phi_2] | \Theta_{4,cd} \rangle \rangle [[\mathcal{M}_{W_a X_b}] \otimes [\mathcal{M}_{Y_c Z_d}]]$$

To do the computation we make some simplifications; let $\Gamma_1 = \langle \Theta_{1,ab} | [\Phi_1] | \Theta_{2,cd} \rangle$, $\Gamma_2 = \langle \Theta_{3,ab} | [\Phi_2] | \Theta_{4,cd} \rangle$, and let $[\Gamma_n] = \Gamma_n \mathcal{V}_T([[\mathcal{M}_{W_a X_b}] \otimes [\mathcal{M}_{Y_c Z_d}]])$ for $n \in \{1, 2\}$. We can combine the individual 4×4 CM matrices $[\Gamma_1]$ and $[\Gamma_2]$ over Υ by the computation:

$$[\mathcal{M}] = \Upsilon_{uv}([\Gamma_1])_u([\Gamma_2])_v = \langle [\Gamma_1] | [\Upsilon] | [\Gamma_2] \rangle = [\Gamma_1] \Upsilon [\Gamma_2]$$

We can then measure the derived matrix $[\mathcal{M}]$ in which we find:

$$\mathcal{L} = \langle Y | \langle W | [\mathcal{M}] | X \rangle | Z \rangle$$

An example can be seen at the end of section 5.2.

5.4 Correspondence Matrix Computations

Here we give examples of how we can bring the tools developed throughout paper together. Particularly we demonstrate the transformations needed to combine logical expressions. We will see two examples; the first is how to combine a 2-variable logical expression with a single variable logical expression, the second shows how to combine a 4-variable with a 3-variable logical expression.

5.4.1 Two variable logical Computation

To combine two logical expressions each with two logical variables or less the expressions must be of the form $\mathcal{L} = \mathcal{L}_1 \Phi \mathcal{L}_2$ where $\mathcal{L}_1 = (X \Theta_1 Y)$ and $\mathcal{L}_2 = (X \Theta_2 Y)$. As found in section 3.2.3:

$$\mathcal{L} = (X \Theta_1 Y) \Phi (X \Theta_2 Y) = X \langle \Theta_1 | [\Phi] | \Theta_2 \rangle Y$$

where $\langle \Theta_1 | [\Phi] | \Theta_2 \rangle$ gives the operator $(\Theta_1 \Phi \Theta_2)$ which corresponds to the CM $[[\Theta]_1 \Phi [\Theta_2]]$.

As an example, consider the expression $\mathcal{L} = \neg((X \wedge \neg Y) \Leftrightarrow \neg X)$ which can be seen to be two logical expressions $(X \wedge \neg Y)$ and $\neg X$ operated on by \Leftrightarrow . To get the expression in the needed form we first write it in bra-ket notation: $\neg \langle \langle X | [\wedge] | \neg Y \rangle | [\Leftrightarrow] | \langle \neg X | [L] | Y \rangle \rangle$ which by using the rules [3.1] for matrix computations we have that by rule 3; $\langle X | [\wedge] | \neg Y \rangle = \langle X | [\uparrow] | Y \rangle$ and that $\langle \neg X | [L] | Y \rangle$ becomes $\langle X | [\neg L] | Y \rangle$ by application of rule 4. We then have that by rule 2. $\neg \langle \langle X | [\uparrow] | Y \rangle | [\Leftrightarrow] | \langle X | [\neg L] | Y \rangle \rangle = \langle \langle X | [\uparrow] | Y \rangle | [\Downarrow] | \langle X | [\neg L] | Y \rangle \rangle$. We can continue to combine the inner bra-ket expressions:

$$\langle X | [\uparrow] | Y \rangle \Downarrow \langle X | [\neg L] | Y \rangle = \langle X | [[\uparrow] \Downarrow [\neg L]] | Y \rangle = \langle X | [\neg \wedge] | Y \rangle = \neg X \vee \neg Y$$

which shows that $\mathcal{L} = \neg((X \wedge \neg Y) \Leftrightarrow \neg X) = \neg X \vee \neg Y$

5.4.2 Four variable logical Computation

To combine two expressions, each with four or less logical variables, the expressions must be of the form:

$$\mathcal{L} = \mathcal{L}_1 \Upsilon \mathcal{L}_2 \text{ where } \mathcal{L}_1 = (W \Theta_1 X) \Phi_1 (Y \Theta_2 Z) \text{ and } \mathcal{L}_2 = (W \Theta_3 X) \Phi_2 (Y \Theta_4 Z)$$

This paper presented two algorithms by which we may compute \mathcal{L} [5.3]; we will perform the computation using the second algorithm and utilize higher dimensional matrices.

As an example, let $\mathcal{L}_1 = (W \Downarrow X) \Rightarrow (\neg Y \wedge Z)$ from the example in 5.2 and $\mathcal{L}_2 = \neg((Y \wedge \neg Z) \Leftrightarrow \neg W)$, say we want to compute $\mathcal{L}_1 \Rightarrow \mathcal{L}_2$. We can write \mathcal{L} in the bra-ket form:

$$\mathcal{L} = \langle \langle \langle W | [\Downarrow] | X \rangle | [\Rightarrow] | \langle \neg Y | [\wedge] | Z \rangle \rangle | [\Rightarrow] | \neg \langle \langle Y | [\wedge] | \neg Z \rangle \rangle | [\Leftrightarrow] | \langle \neg W | [L] | X \rangle \rangle \rangle$$

We then format \mathcal{L} by applying the transformation rules e.g. $\langle Y | [\wedge] | \neg Z \rangle = \langle Y | [\uparrow] | Z \rangle$ so as to be in the format above. We have that:

$$\mathcal{L} = \langle\langle W|[\uparrow]X\rangle\rangle|[\Rightarrow]\langle Y|[\downarrow]Z\rangle\rangle|[\Rightarrow]\langle\langle Y|[\uparrow]Z\rangle\rangle|[\uparrow]\langle W|[-L]X\rangle\rangle$$

As \mathcal{L} is in the correct form we make the assignments: $[\Phi_1] = [\Rightarrow]$, $[\Phi_2] = [\uparrow]$, $[\Theta_1] = [\uparrow]$, $[\Theta_2] = [\downarrow]$, $[\Theta_3] = [-L]$, $[\Theta_4] = [\uparrow]$ and $[\Upsilon] = [\Rightarrow]$.

We can compute the above by noting that $\mathcal{L}_1 = \langle\langle W|[\uparrow]X\rangle\rangle|[\Rightarrow]\langle Y|[\downarrow]Z\rangle\rangle$ and $\mathcal{L}_2 = \langle\langle Y|[\uparrow]Z\rangle\rangle|[\uparrow]\langle W|[-L]X\rangle\rangle$ can be represented as unique 4×4 CM's of the form $[\Gamma_n] = \Gamma_n \mathcal{V}_T([\mathcal{M}_{W_i X_j}] \otimes [\mathcal{M}_{Y_k Z_l}])$ for $n \in \{1, 2\}$. We then find the respective CM's via the method discussed in 5.1.1 by operating directly on the CM's of \mathcal{L}_1 and \mathcal{L}_2 :

$$[\Gamma_1] = [[\uparrow]_1 \otimes [\downarrow]_1] \Downarrow [[\uparrow]_2 \otimes [\downarrow]_1] \Downarrow [[\uparrow]_2 \otimes [\downarrow]_2], [\Gamma_2] = [[\uparrow]_1 \otimes [-L]_2] \Downarrow [[\uparrow]_2 \otimes [-L]_1]$$

We can compute the measurement matrix $[\mathcal{M}]$ as we have that:

$$[\mathcal{M}] = \Rightarrow_{rs} ([\Gamma_1])_r ([\Gamma_2])_s = \langle [\Gamma_1] | [\Rightarrow] | [\Gamma_2] \rangle$$

which we will compute $[\mathcal{M}]$ both ways. The first way gives:

$$\begin{aligned} \Rightarrow_{rs} ([\Gamma_1])_r ([\Gamma_2])_s &= \\ &= [[\Gamma_1]_1 [\Gamma_2]_1] \Downarrow [[\Gamma_1]_2 [\Gamma_2]_1] \Downarrow [[\Gamma_1]_2 [\Gamma_2]_2] = \neg [[\Gamma_1]_1 [\Gamma_2]_2] = [\mathcal{M}] \end{aligned}$$

The second way to compute $[\mathcal{M}]$ gives:

$$\langle [\Gamma_1] | [\Rightarrow] | [\Gamma_2] \rangle = [\Gamma_1] \Rightarrow [\Gamma_2] = [\mathcal{M}]$$

This shows that $[\mathcal{M}]$ can be computed by the index-wise application of $\Rightarrow!$ The 4×4 CM's are:

$$[\Gamma_1] = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \quad [\Gamma_2] = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \quad [\mathcal{M}] = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

To finish the computation we need to measure $[\mathcal{M}]$ by the correct ordering of variables and find:

$$\begin{aligned} \langle Y | \langle W | [\mathcal{M}] | X \rangle | Z \rangle &= \begin{bmatrix} YW \\ -YW \\ Y\neg W \\ -Y\neg W \end{bmatrix}^T \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} XZ \\ X\neg Z \\ \neg XZ \\ \neg X\neg Z \end{bmatrix} \\ &= (-WXZ) \Downarrow ((W \Rightarrow Y)X\neg Z) \Downarrow ((W \Rightarrow Y)\neg XZ) \Downarrow ((Y \Rightarrow W)(\neg X\neg Z)) = \mathcal{L} \end{aligned}$$

6 Summary

The foundation of this paper is the introduction of the correspondence matrix (CM); a binary matrix which corresponds uniquely to a particular logical operator in propositional logic [3]:

Op.	Matrix	Op.	Matrix	Op.	Matrix	Op.	Matrix
\oplus	$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$	$[0]$	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$	$[\Leftrightarrow]$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$[\Downarrow]$	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
$[\wedge]$	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	$[\neg\vee]$	$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$	$[\Uparrow]$	$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$	$[\Downarrow]$	$\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$
$[\Rightarrow]$	$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$	$[\Leftarrow]$	$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$	$[\vee]$	$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$	$[\neg\wedge]$	$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$
$[R]$	$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$	$[\neg R]$	$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$	$[L]$	$\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$	$[\neg L]$	$\begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$

By representing logical operators as CM's we have introduced a mathematical framework in which we can not only perform computations on and between logical expressions, but computations on operators themselves! This paper explored many properties applications of CM's; It was found that CM's are linear [3.2.1] leading to the ability to break apart any logical expression in to the XOR decomposition of sub-expressions:

$$X\Theta Y = \langle X | [\Theta] | Y \rangle = X_i \Theta_{ij} Y_j$$

as well as decompose logical expressions by decomposing their unique operator's CM representations, furthermore that the decomposition of CM's allows for the decomposition of their corresponding logical operators:

$$[\Theta] = \Theta_{11}[\wedge] \Downarrow \Theta_{12}[\Uparrow] \Downarrow \Theta_{21}[\Downarrow] \Downarrow \Theta_{22}[\neg\vee]$$

We also found that linearity among CM's has allowed for the ability to quotient logical expressions [3.2.2]:

$$X\Theta_1 Y \setminus X\Theta_2 Y = \langle X | [\Theta_1] \setminus [\Theta_2] | Y \rangle = X(\Theta \wedge \neg\Theta_2)Y$$

where $(\Theta \wedge \neg\Theta_2)$ is the operator corresponding to the CM $[[\Theta_1] \wedge [\neg\Theta_2]]$.

We generalized the notion of a CM to that of a logical measurement matrix (LM) of which CM's are positive valuations of [4.1]:

$$[\Theta] = \mathcal{V}_T([\mathcal{M}_{X\Theta Y}]) = \begin{bmatrix} \mathcal{V}_T(X)\Theta\mathcal{V}_T(Y) & \mathcal{V}_T(X)\Theta\mathcal{V}_T(\neg Y) \\ \mathcal{V}_T(\neg X)\Theta\mathcal{V}_T(Y) & \mathcal{V}_T(\neg X)\Theta\mathcal{V}_T(\neg Y) \end{bmatrix}$$

where a LM is a matrix representation or metric of how logical variables are related to one another via a particular logical operator. Like any matrix computations can be performed by rotations, transpositions, and negations of LM's and their positively valuated CM's; for example, the foundational rule 2. [3.1]:

$$\neg(X\Theta Y) = \neg\langle X | [\Theta] | Y \rangle = \langle X | [\neg\Theta] | Y \rangle = X\neg(\Theta)Y$$

shows that by operating on matrices we can transform logical expressions.

A substantial finding is that we can interpret the relationship between two logical expressions as being the result of a measurement [4]:

$$\langle \mathcal{L}_1 | [\mathcal{M}_{X\Theta Y}] | \mathcal{L}_2 \rangle = \Theta_{rs}(\mathcal{L}_1 \Leftrightarrow X_r) \wedge (Y_s \Leftrightarrow \mathcal{L}_2)$$

where it was found that measuring two logical expressions by a LM tells whether the resulting expression is true or false or under what circumstances the result will be such. If we measure logical expressions via a CM, the resulting ‘polarized’ logical expression is one composed from components shared by both of the measured expressions and features of the operator.

This paper showed that by negating CM’s of logical operators we create compliment logical operators [3.3]. Using these complimented or negated operators we are able to give a proof of one of this papers most significant findings; that logical expressions can be computed by combining the operators of sub-expressions:

$$(X\Theta_1Y)\Phi(X\Theta_2Y) = \langle X|[\Theta_1]|Y\rangle \Phi \langle X|[\Theta_2]|Y\rangle = \langle X|[[\Theta_1]\Phi[\Theta_2]]|Y\rangle$$

thus giving that:

$$(X\Theta_1Y)\Phi(X\Theta_2Y) = X(\Theta_1\Phi\Theta_2)Y$$

where $\Theta_1\Phi\Theta_2$ is the operator corresponding to the CM $[[\Theta_1] \Phi [\Theta_2]]$ [3.2.3]. It was shown that this result was not restricted to two variable logical expressions [5.3].

We proved that any logical expression $\mathcal{L} = (\mathcal{L}_1\Theta_1\mathcal{L}_2)\Phi(X\Theta_2Y)$ having a finite number of variables can be projected in to some higher dimensional LM:

$$[\mathcal{M}_{(\mathcal{L}_1\Theta_1\mathcal{L}_2)\Phi(X\Theta_2Y)}] \xrightarrow{p} \langle \Theta_{1,ij} | [\Phi] | \Theta_{2,kl} \rangle [[\mathcal{M}_{(\mathcal{L}_1)_i(\mathcal{L}_2)_j}] \otimes [\mathcal{M}_{X_k Y_l}]]$$

Furthermore that all non-conjunctive operations in LM’s which are tensored together can be pulled out in to a matrix modifier [5.2]. In doing so we have shown the ability to measure relationships between logical operators, this allowing for deciding which LM’s need to be computed. Finally we showed how to derive $2^n \times 2^n$ basis LM’s for any $n \in \mathbb{N}$ where $n \geq 1$,

$$[\mathcal{M}_{\mathcal{L}_1 \dots \mathcal{L}_n}] \xrightarrow{p} (\otimes_{i=1}^{2n} |\mathcal{L}_{2i-1}\rangle)(\otimes_{i=1}^{2n} \langle \mathcal{L}_{2i}|) = \otimes_{i=1}^{2n} |\mathcal{L}_{2i-1}\rangle \langle \mathcal{L}_{2i}| = \otimes_{i=1}^{2n} [\mathcal{M}_{\mathcal{L}_{(2i-1)}\mathcal{L}_{(2i)}}]$$

and how to make measurements on these basis LM’s:

$$\langle \otimes_{i=1}^{2n} \mathcal{L}_{2i-1} | \otimes_{i=1}^{2n} [\mathcal{M}_{\mathcal{L}_{(2i-1)}\mathcal{L}_{(2i)}}] | \otimes_{i=1}^{2n} \mathcal{L}_{2i} \rangle$$

7 Thanks

The author would like to thank all those who make knowledge available to all, free of charge. A special thanks to Wikipedia and Wolfram Alpha. The author would also like to thank the Great Green Arkleseizure, without which this work would not be possible.

References

- [1] Oggier, Frédérique *Propositional Logic*. <http://www1.spms.ntu.edu.sg/~frederique/dm2.pdf>
- [2] Wikipedia contributors. *Bra-ket notation*. Wikipedia, The Free Encyclopedia; 2018 Apr 13, 15:58 UTC [cited 2018 Apr 28] https://en.wikipedia.org/w/index.php?title=Bra%E2%80%93ket_notation&oldid=836248310
- [3] Redish, Edward F. *Dirac notation*. <https://www.physics.umd.edu/courses/Phys374/fall05/files/DiracNotation.pdf>
- [4] Wikipedia contributors. *Outer Product*. Wikipedia, The Free Encyclopedia; 2018 May 10, 16:30 UTC [cited 2018 May 20] https://en.wikipedia.org/w/index.php?title=Outer_product&oldid=840550121
- [5] Stover, Christopher and Weisstein, Eric W. *Einstein Summation*. From MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/EinsteinSummation.html>
- [6] Wikipedia contributors. *Schrödinger's cat*. Wikipedia, The Free Encyclopedia; 2018 May 16, 17:27 UTC [cited 2018 May 20]. https://en.wikipedia.org/w/index.php?title=Schr/%C3%B6dinger/%27s_cat&oldid=841573463
- [7] Looking Glass Universe, 2015. *Interference in Quantum Mechanics*. Youtube [Accessed May 13th 2018]. <https://www.youtube.com/watch?v=tt8gVXDsh7Q>
- [8] Cresser, James. *Observables and Measurements in Quantum Mechanics*. <https://www.physics.umd.edu/courses/Phys374/fall05/files/DiracNotation.pdf>
- [9] Overbeek, Roy. *Tutorial on Mathematical Induction*. <http://www.few.vu.nl/~rbakhshi/teaching/induction.pdf>