Logic Gate Arithmetic and Quaternions

Christopher O'Neill

BAEC, Civic Centre, Main Street, Bray, Co. Wicklow

7 December 2020

Abstract

A new method for Logic Gate Arithmetic is proposed allowing sets to be added, subtracted, multiplied, and divided in an entirely new way leading to an alternate and expanded appreciation of Set Theory. 11 new Logic Operators are derived from the known Logic Operators: AND, OR, NOR, XOR, and XNOR and their relationships are described extensively. Utilising the principles of Dimensional Gate Operators (DGO), the new operators are shown to be directly related to dimensionality, both higher and lower, and the relationship to the Quaternions is given a fledgling treatment.

Background & Terminology

Logic Operators are an important part of modern Computer Systems and Set Theory. There are six primary Logic Gate Operators; AND, OR, NOR, XOR, XNOR and NOT, as they are traditionally perceived. More accurately, we could say that there are only 3 (AND, OR, and XOR), with the NOT operator being more of a decorator acting on these to create the other 2 (anti-operators), for a total of 5 in total.

In 'REIMAGINING COMPLEX NUMBERS'[1], I explored the possibility that these Logic Operators were incomplete and added 11 more — for a total of 16. Shortly after writing this, I watched a video by N J Wildberger called "Implication and 16 logical operations" and realised that he had arrived at the same conception nearly a full two years, before I had.[2] Wildberger derives his new set of logic operations in a formal algebraic sense using Bool Algebra and goes on to name some of them in relation to the Stoic Logic of Modus Ponens i.e. implications and others in relation to their own internal properties.

For example, the Logic or Truth table that corresponds to the binary number ‘0000’ (read right to left), he denotes as ‘0’ or ‘zero’. It’s counterpart ‘1111’ is therefore ‘1’. ‘IMP’ stands for implication and OT1 stands for ‘Original Term’.

0	NOR	NIMP$_2$	NOT$_2$	NIMP$_1$	NOT$_1$	XOR	NAND
‘0000’	‘0001’	‘0010’	‘0011’	‘0100’	‘0101’	‘0110’	‘0111’

AND	XNOR	OT$_1$	IMP1	OT$_2$	IMP$_2$	OR	1
‘1000’	‘1001’	‘1010’	‘1011’	‘1100’	‘1101’	‘1110’	‘1111’

Tables 1: Wildberger’s 16 Logic Operations

Once a naming convention is in place, it doesn’t really matter how it got there. However, these new logic operators lack Boolean Algebra symbols, like ‘^’, ‘V’, ‘Δ’ and ‘!Δ’.[3][2] To counteract this, I will propose my own naming convention. I developed this method, before realising that Wildberger had already given names to all of the logic operations — I must have skipped over that section of the video.

In any case, I’m not too concerned with the prospect of different naming conventions, this earlier on in my research. After all, the XOR and XNOR logic gates are sometimes called EOR and ENOR, without much complication, and there are numerous variations used for symbols throughout the literature, including ‘!’ and ‘¬’ for NOT etc.

More confusingly, however, is applying the binary numbers to the Truth Tables. Binary numbers are usually written horizontally, whereas Truth Tables are nearly always vertical (See the Appendix). When compiling Wildberger’s operations into Tables 1, it seemed necessary to list the operators, as though their binary numbers are being read from ‘right to left’. To avoid headaches, I will continue with this convention. And so, without further ado, I will layout the framework for how I derived all 16 Logic Operators from the original six.

“Logic, dear Watson”

The binary numbers ‘0000’ to ‘1111’ can be used to represent all 16 of the Dimensional Gate Operators. However, some of these operations don’t appear all that logical. To alleviate this concern, I decided to attempt to derive the other eleven operators from the original five.

?	NOR	?	?	?	?	XOR	NAND
‘0000’	‘0001’	‘0010’	‘0011’	‘0100’	‘0101’	‘0110’	‘0111’
AND	XNOR	?	?	?	?	OR	?
‘1000’	‘1001’	‘1010’	‘1011’	‘1100’	‘1101’	‘1110’	‘1111’

Table 2: “A rose by any other name?”

The first step was to combine the known logic gates; AND, OR, XOR, NAND, XNOR and NOR to see if we could create any knew terms. We can get to XOR, by adding AND and OR together, using the OR operator:

$$((\wedge)\ V\ (V)) = \Delta$$

But this is obvious and doesn’t lead us out of the loop into the unknown parts of the DGO. Similarly with ((^) ^ (V)) = ^. It is easy to see how this method will lead us in circles.

Taking a step back and reexamining the Truth Tables for these operations, I realised that there were two more ‘binary’ numbers staring me in the face: ‘1010’ and ‘1100’. These are equivalent to Wildberger’s OT_1 and OT_2. However, I identified them as RIGHT (or ‘R’) and LEFT (L), as in Right and Left Multiplication. Applying the NOR operator, !R (NOT RIGHT) and !L (NOT LEFT) are easily obtained.

From here, we can get to a new Logic Operator; ‘0100’, which is made from the combination of Δ and !L:

$$((\Delta) \wedge (!L)) = 0100$$

I’ve called *0100* ‘^U’ (meaning AND UP), because it is similar to ‘^’ (1000) shifted ‘up’ one space. The NOT version of this gate is *1011* giving us; ‘!^U’ (NOT AND UP). Similarly, !V shifted down one space gives us *0010*, so this is ‘!VD’ (NOR DOWN).

Therefore, ‘!VD’ is equal to *1101*.

Together with the NOT versions of these two new Logic Operations, we are able to fill 14 of the 16 arrangements in Table 2.

N	!V	!VD	!R	^U	!L	Δ	!^
‘0000’	‘0001’	‘0010’	‘0011’	‘0100’	‘0101’	‘0110’	‘0111’

^	!Δ	L	!^U	R	VD	V	!N
‘1000’	‘1001’	‘1010’	‘1011’	‘1100’	‘1101’	‘1110’	‘1111’

Table 3: All 16 operations with their names and symbols

Somewhere along the line, ‘0000’ and ‘1111’ were named: i.e. “N” and “!N” respectively (standing for “NONE” and “NOT NONE”).

These could equally be written as “NOT ALL” (NA) and “ALL” (A). Both ‘None’ and ‘A’ are already symbols in Boolean Algebra, so “NONE” (N) and “NOT NONE” (!N) are used instead.

Getting to ‘N’ and ‘!N’ from any of these gates is a simple matter. All you have to do is apply one gate together with its NOT version:

((V) Δ (!V)) = N

The reverse is given by:

((!V) Δ (!V)) = !N

Now that we have all 16 Dimensional Gate Operators and their Boolean Algebra symbols in place, we can explore the relationships between these and even write them in terms of one another. For example, all of these statements are TRUE:

((R) Δ (L)) = Δ ((!N) !Δ (L)) = !L (!VD) !Δ (L) = !^

((Δ) ^ (!L)) = ^U

This means that the last equation in this list can equally be written as:

((R) Δ (L)) (!(!VD) !Δ !(L)) ((!N) !Δ (L)) = ^U

This is a lot to take in and remember at first glance, so I’ve decided to organise all this information in the following tables:

N : [0000]

	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
V	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
VD	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
R	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
!^U	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
L	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
!Δ	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
^	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
!^	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Δ	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
!L	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
^U	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
!R	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
!VD	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
!V	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N

!V : [0001]

	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
V	!V	N	!V	N	!V	N	!V	N	!V	N	!V	N	!V	N	!V	N
VD	!VD	!VD	N	N	!VD	!VD	N	N	!VD	!VD	N	N	!VD	!VD	N	N
R	!R	!VD	!V	N	!R	!VD	!V	N	!R	!VD	!V	N	!R	!VD	!V	N
!^U	^U	^U	^U	^U	N	N	N	N	^U	^U	^U	^U	N	N	N	N
L	!L	^U	!L	^U	!V	N	!V	N	!L	^U	!L	^U	!V	N	!V	N
!Δ	Δ	Δ	^U	^U	!VD	!VD	N	N	Δ	Δ	^U	^U	!VD	!VD	N	N
^	!^	Δ	!L	^U	!R	!VD	!V	N	!^	Δ	!L	^U	!R	!VD	!V	N
!^	^	^	^	^	^	^	^	^	N	N	N	N	N	N	N	N
Δ	!Δ	^	!Δ	^	!Δ	^	!Δ	^	!V	N	!V	N	!V	N	!V	N
!L	L	L	^	^	L	L	^	^	!VD	!VD	N	N	!VD	!VD	N	N
^U	!^U	L	!Δ	^	!^U	L	!Δ	^	!R	!VD	!V	N	!R	!VD	!V	N
!R	R	R	R	R	^	^	^	^	^U	^U	^U	^U	N	N	N	N
!VD	VD	R	VD	R	!Δ	^	!Δ	^	!L	^U	!L	^U	!V	N	!V	N
!V	V	V	R	R	L	L	^	^	Δ	Δ	^U	^U	!VD	!VD	N	N
N	!N	V	VD	R	!^U	L	!Δ	^	!^	Δ	!L	^U	!R	!VD	!V	N

!VD : [0010]

	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!N	!N	V	VD	R	!^U	L	!Δ	^	!^	Δ	!L	^U	!R	!VD	!V	N
V	V	V	R	R	L	L	^	^	Δ	Δ	^U	^U	!VD	!VD	N	N
VD	VD	R	VD	R	!Δ	^	!Δ	^	!L	^U	!L	^U	!V	N	!V	N
R	R	R	R	R	^	^	^	^	^U	^U	^U	^U	N	N	N	N
!^U	!^U	L	!Δ	^	!^U	L	!Δ	^	!R	!VD	!V	N	!R	!VD	!V	N
L	L	L	^	^	L	L	^	^	!VD	!VD	N	N	!VD	!VD	N	N
!Δ	!Δ	^	!Δ	^	!Δ	^	!Δ	^	!V	N	!V	N	!V	N	!V	N
^	^	^	^	^	^	^	^	^	N	N	N	N	N	N	N	N
!^	!^	Δ	!L	^U	!R	!VD	!V	N	!^	Δ	!L	^U	!R	!VD	!V	N
Δ	Δ	Δ	^U	^U	!VD	!VD	N	N	Δ	Δ	^U	^U	!VD	!VD	N	N
!L	!L	^U	!L	^U	!V	N	!V	N	!L	^U	!L	^U	!V	N	!V	N
^U	^U	^U	^U	^U	N	N	N	N	^U	^U	^U	^U	N	N	N	N
!R	!R	!VD	!V	N	!R	!VD	!V	N	!R	!VD	!V	N	!R	!VD	!V	N
!VD	!VD	!VD	N	N	!VD	!VD	N	N	!VD	!VD	N	N	!VD	!VD	N	N
!V	!V	N	!V	N	!V	N	!V	N	!V	N	!V	N	!V	N	!V	N
N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N

Fig 0, 1 & 2

!R : [0011]

	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!N	!N	V	VD	R	!^U	L	!Δ	^	!^	Δ	!L	^U	!R	!VD	!V	N
V	!N	V	VD	R	!^U	L	!Δ	^	!^	Δ	!L	^U	!R	!VD	!V	N
VD	!N	V	VD	R	!^U	L	!Δ	^	!^	Δ	!L	^U	!R	!VD	!V	N
R	!N	V	VD	R	!^U	L	!Δ	^	!^	Δ	!L	^U	!R	!VD	!V	N
!^U	!N	V	VD	R	!^U	L	!Δ	^	!^	Δ	!L	^U	!R	!VD	!V	N
L	!N	V	VD	R	!^U	L	!Δ	^	!^	Δ	!L	^U	!R	!VD	!V	N
!Δ	!N	V	VD	R	!^U	L	!Δ	^	!^	Δ	!L	^U	!R	!VD	!V	N
^	!N	V	VD	R	!^U	L	!Δ	^	!^	Δ	!L	^U	!R	!VD	!V	N
!^	!N	V	VD	R	!^U	L	!Δ	^	!^	Δ	!L	^U	!R	!VD	!V	N
Δ	!N	V	VD	R	!^U	L	!Δ	^	!^	Δ	!L	^U	!R	!VD	!V	N
!L	!N	V	VD	R	!^U	L	!Δ	^	!^	Δ	!L	^U	!R	!VD	!V	N
^U	!N	V	VD	R	!^U	L	!Δ	^	!^	Δ	!L	^U	!R	!VD	!V	N
!R	!N	V	VD	R	!^U	L	!Δ	^	!^	Δ	!L	^U	!R	!VD	!V	N
!VD	!N	V	VD	R	!^U	L	!Δ	^	!^	Δ	!L	^U	!R	!VD	!V	N
!V	!N	V	VD	R	!^U	L	!Δ	^	!^	Δ	!L	^U	!R	!VD	!V	N
N	!N	V	VD	R	!^U	L	!Δ	^	!^	Δ	!L	^U	!R	!VD	!V	N

^U : [0100]

	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
V	N	!V	N	!V	N	!V	N	!V	N	!V	N	!V	N	!V	N	!V
VD	N	N	!VD	!VD	N	N	!VD	!VD	N	N	!VD	!VD	N	N	!VD	!VD
R	N	!V	!VD	!R	N	!V	!VD	!R	N	!V	!VD	!R	N	!V	!VD	!R
!^U	N	N	N	N	^U	^U	^U	^U	N	N	N	N	^U	^U	^U	^U
L	N	!V	N	!V	^U	!L	^U	!L	N	!V	N	!V	^U	!L	^U	!L
!Δ	N	N	!VD	!VD	^U	^U	Δ	Δ	N	N	!VD	!VD	^U	^U	Δ	Δ
^	N	!V	!VD	!R	^U	!L	Δ	!^	N	!V	!VD	!R	^U	!L	Δ	!^
!^	N	N	N	N	N	N	N	N	^	^	^	^	^	^	^	^
Δ	N	!V	N	!V	N	!V	N	!V	^	!Δ	^	!Δ	^	!Δ	^	!Δ
!L	N	N	!VD	!VD	N	N	!VD	!VD	^	^	L	L	^	^	L	L
^U	N	!V	!VD	!R	N	!V	!VD	!R	^	!Δ	L	!^U	^	!Δ	L	!^U
!R	N	N	N	N	^U	^U	^U	^U	^	^	^	^	R	R	R	R
!VD	N	!V	N	!V	^U	!L	^U	!L	^	!Δ	^	!Δ	R	VD	R	VD
!V	N	N	!VD	!VD	^U	^U	Δ	Δ	^	^	L	L	R	R	V	V
N	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N

!L : [0101]

	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
V	!V	!V	!V	!V	!V	!V	!V	!V	!V	!V	!V	!V	!V	!V	!V	!V
VD	!VD	!VD	!VD	!VD	!VD	!VD	!VD	!VD	!VD	!VD	!VD	!VD	!VD	!VD	!VD	!VD
R	!R	!R	!R	!R	!R	!R	!R	!R	!R	!R	!R	!R	!R	!R	!R	!R
!^U	^U	^U	^U	^U	^U	^U	^U	^U	^U	^U	^U	^U	^U	^U	^U	^U
L	!L	!L	!L	!L	!L	!L	!L	!L	!L	!L	!L	!L	!L	!L	!L	!L
!Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
^	!^	!^	!^	!^	!^	!^	!^	!^	!^	!^	!^	!^	!^	!^	!^	!^
!^	^	^	^	^	^	^	^	^	^	^	^	^	^	^	^	^
Δ	!Δ	!Δ	!Δ	!Δ	!Δ	!Δ	!Δ	!Δ	!Δ	!Δ	!Δ	!Δ	!Δ	!Δ	!Δ	!Δ
!L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
^U	!^U	!^U	!^U	!^U	!^U	!^U	!^U	!^U	!^U	!^U	!^U	!^U	!^U	!^U	!^U	!^U
!R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
!VD	VD	VD	VD	VD	VD	VD	VD	VD	VD	VD	VD	VD	VD	VD	VD	VD
!V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N

Fig 3, 4 & 5

Δ : [0110]

	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!N	!N	V	VD	R	!^U	L	!Δ	^	!^	Δ	!L	^U	!R	!VD	!V	N
V	V	!N	R	VD	L	!^U	^	!Δ	Δ	!^	^U	!L	!VD	!R	N	!V
VD	VD	R	!N	V	!Δ	^	!^U	L	!L	^U	!^	Δ	!V	N	!R	!VD
R	R	VD	V	!N	^	!Δ	L	!^U	^U	!L	Δ	!^	N	!V	!VD	!R
!^U	!^U	L	!Δ	^	!N	V	VD	R	!R	!VD	!V	N	!^	Δ	!L	^U
L	L	!^U	^	!Δ	V	!N	R	VD	!VD	!R	N	!V	Δ	!^	^U	!L
!Δ	!Δ	^	!^U	L	VD	R	!N	V	!V	N	!R	!VD	!L	^U	!^	Δ
^	^	!Δ	L	!^U	R	VD	V	!N	N	!V	!VD	!R	^U	!L	Δ	!^
!^	!^	Δ	!L	^U	!R	!VD	!V	N	!N	V	VD	R	!^U	L	!Δ	^
Δ	Δ	!^	^U	!L	!VD	!R	N	!V	V	!N	R	VD	L	!^U	^	!Δ
!L	!L	^U	!^	Δ	!V	N	!R	!VD	VD	R	!N	V	!Δ	^	!^U	L
^U	^U	!L	Δ	!^	N	!V	!VD	!R	R	VD	V	!N	^	!Δ	L	!^U
!R	!R	!VD	!V	N	!^	Δ	!L	^U	!^U	L	!Δ	^	!N	V	VD	R
!VD	!VD	!R	N	!V	Δ	!^	^U	!L	L	!^U	^	!Δ	V	!N	R	VD
!V	!V	N	!R	!VD	!L	^U	!^	Δ	!Δ	^	!^U	L	VD	R	!N	V
N	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N

!^ : [0111]

	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!N	!N	V	VD	R	!^U	L	!Δ	^	!^	Δ	!L	^U	!R	!VD	!V	N
V	!N	!N	VD	VD	!^U	!^U	!Δ	!Δ	!^	!^	!L	!L	!R	!R	!V	!V
VD	!N	V	!N	V	!^U	L	!^U	L	!^	Δ	!^	Δ	!R	!VD	!R	!VD
R	!N	!N	!N	!N	!^U	!^U	!^U	!^U	!^	!^	!^	!^	!R	!R	!R	!R
!^U	!N	V	VD	R	!N	V	VD	R	!^	Δ	!L	^U	!^	Δ	!L	^U
L	!N	!N	VD	VD	!N	!N	VD	VD	!^	!^	!L	!L	!^	!^	!L	!L
!Δ	!N	V	!N	V	!N	V	!N	V	!^	Δ	!^	Δ	!^	Δ	!^	Δ
^	!N	!N	!N	!N	!N	!N	!N	!N	!^	!^	!^	!^	!^	!^	!^	!^
!^	!N	V	VD	R	!^U	L	!Δ	^	!N	V	VD	R	!^U	L	!Δ	^
Δ	!N	!N	VD	VD	!^U	!^U	!Δ	!Δ	!N	!N	VD	VD	!^U	!^U	!Δ	!Δ
!L	!N	V	!N	V	!^U	L	!^U	L	!N	V	!N	V	!^U	L	!^U	L
^U	!N	!N	!N	!N	!^U	!^U	!^U	!^U	!N	!N	!N	!N	!^U	!^U	!^U	!^U
!R	!N	V	VD	R	!N	V	VD	R	!N	V	VD	R	!N	V	VD	R
!VD	!N	!N	VD	VD	!N	!N	VD	VD	!N	!N	VD	VD	!N	!N	VD	VD
!V	!N	V	!N	V	!N	V	!N	V	!N	V	!N	V	!N	V	!N	V
N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N

^ : [1000]

	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!N	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
V	N	N	!VD	!VD	^U	^U	Δ	Δ	^	^	L	L	R	R	V	V
VD	N	!V	N	!V	^U	!L	^U	!L	^	!Δ	^	!Δ	R	VD	R	VD
R	N	N	N	N	^U	^U	^U	^U	^	^	^	^	R	R	R	R
!^U	N	!V	!VD	!R	N	!V	!VD	!R	^	!Δ	L	!^U	^	!Δ	L	!^U
L	N	N	!VD	!VD	N	N	!VD	!VD	^	^	L	L	^	^	L	L
!Δ	N	!V	N	!V	N	!V	N	!V	^	!Δ	^	!Δ	^	!Δ	^	!Δ
^	N	N	N	N	N	N	N	N	^	^	^	^	^	^	^	^
!^	N	!V	!VD	!R	^U	!L	Δ	!^	N	!V	!VD	!R	^U	!L	Δ	!^
Δ	N	N	!VD	!VD	^U	^U	Δ	Δ	N	N	!VD	!VD	^U	^U	Δ	Δ
!L	N	!V	N	!V	^U	!L	^U	!L	N	!V	N	!V	^U	!L	^U	!L
^U	N	N	N	N	^U	^U	^U	^U	N	N	N	N	^U	^U	^U	^U
!R	N	!V	!VD	!R	N	!V	!VD	!R	N	!V	!VD	!R	N	!V	!VD	!R
!VD	N	N	!VD	!VD	N	N	!VD	!VD	N	N	!VD	!VD	N	N	!VD	!VD
!V	N	!V	N	!V	N	!V	N	!V	N	!V	N	!V	N	!V	N	!V
N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N

Fig 6, 7 & 8

!Δ : [1001]

	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!N	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
V	!V	N	!R	!VD	!L	^U	!^	Δ	!Δ	^	!^U	L	VD	R	!N	V
VD	!VD	!R	N	!V	Δ	!^	^U	!L	L	!^U	^	!Δ	V	!N	R	VD
R	!R	!VD	!V	N	!^	Δ	!L	^U	!^U	L	!Δ	^	!N	V	VD	R
!^U	^U	!L	Δ	!^	N	!V	!VD	!R	R	VD	V	!N	^	!Δ	L	!^U
L	!L	^U	!^	Δ	!V	N	!R	!VD	VD	R	!N	V	!Δ	^	!^U	L
!Δ	Δ	!^	^U	!L	!VD	!R	N	!V	V	!N	R	VD	L	!^U	^	!Δ
^	!^	Δ	!L	^U	!R	!VD	!V	N	!N	V	VD	R	!^U	L	!Δ	^
!^	^	!Δ	L	!^U	R	VD	V	!N	N	!V	!VD	!R	^U	!L	Δ	!^
Δ	!Δ	^	!^U	L	VD	R	!N	V	!V	N	!R	!VD	!L	^U	!^	Δ
!L	L	!^U	^	!Δ	V	!N	R	VD	!VD	!R	N	!V	Δ	!^	^U	!L
^U	!^U	L	!Δ	^	!N	V	VD	R	!R	!VD	!V	N	!^	Δ	!L	^U
!R	R	VD	V	!N	^	!Δ	L	!^U	^U	!L	Δ	!^	N	!V	!VD	!R
!VD	VD	R	!N	V	!Δ	^	!^U	L	!L	^U	!^	Δ	!V	N	!R	!VD
!V	V	!N	R	VD	L	!^U	^	!Δ	Δ	!^	^U	!L	!VD	!R	N	!V
N	!N	V	VD	R	!^U	L	!Δ	^	!^	Δ	!L	^U	!R	!VD	!V	N

L : [1010]

	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N
V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
VD	VD	VD	VD	VD	VD	VD	VD	VD	VD	VD	VD	VD	VD	VD	VD	VD
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
!^U	!^U	!^U	!^U	!^U	!^U	!^U	!^U	!^U	!^U	!^U	!^U	!^U	!^U	!^U	!^U	!^U
L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
!Δ	!Δ	!Δ	!Δ	!Δ	!Δ	!Δ	!Δ	!Δ	!Δ	!Δ	!Δ	!Δ	!Δ	!Δ	!Δ	!Δ
^	^	^	^	^	^	^	^	^	^	^	^	^	^	^	^	^
!^	!^	!^	!^	!^	!^	!^	!^	!^	!^	!^	!^	!^	!^	!^	!^	!^
Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
!L	!L	!L	!L	!L	!L	!L	!L	!L	!L	!L	!L	!L	!L	!L	!L	!L
^U	^U	^U	^U	^U	^U	^U	^U	^U	^U	^U	^U	^U	^U	^U	^U	^U
!R	!R	!R	!R	!R	!R	!R	!R	!R	!R	!R	!R	!R	!R	!R	!R	!R
!VD	!VD	!VD	!VD	!VD	!VD	!VD	!VD	!VD	!VD	!VD	!VD	!VD	!VD	!VD	!VD	!VD
!V	!V	!V	!V	!V	!V	!V	!V	!V	!V	!V	!V	!V	!V	!V	!V	!V
N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N

!^U : [1011]

	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N
V	!N	V	!N	V	!N	V	!N	V	!N	V	!N	V	!N	V	!N	V
VD	!N	!N	VD	VD	!N	!N	VD	VD	!N	!N	VD	VD	!N	!N	VD	VD
R	!N	V	VD	R	!N	V	VD	R	!N	V	VD	R	!N	V	VD	R
!^U	!N	!N	!N	!N	!^U	!^U	!^U	!^U	!N	!N	!N	!N	!^U	!^U	!^U	!^U
L	!N	V	!N	V	!^U	L	!^U	L	!N	V	!N	V	!^U	L	!^U	L
!Δ	!N	!N	VD	VD	!^U	!^U	!Δ	!Δ	!N	!N	VD	VD	!^U	!^U	!Δ	!Δ
^	!N	V	VD	R	!^U	L	!Δ	^	!N	V	VD	R	!^U	L	!Δ	^
!^	!N	!N	!N	!N	!N	!N	!N	!N	!^	!^	!^	!^	!^	!^	!^	!^
Δ	!N	V	!N	V	!N	V	!N	V	!^	Δ	!^	Δ	!^	Δ	!^	Δ
!L	!N	!N	VD	VD	!N	!N	VD	VD	!^	!^	!L	!L	!^	!^	!L	!L
^U	!N	V	VD	R	!N	V	VD	R	!^	Δ	!L	^U	!^	Δ	!L	^U
!R	!N	!N	!N	!N	!^U	!^U	!^U	!^U	!^	!^	!^	!^	!R	!R	!R	!R
!VD	!N	V	!N	V	!^U	L	!^U	L	!^	Δ	!^	Δ	!R	!VD	!R	!VD
!V	!N	!N	VD	VD	!^U	!^U	!Δ	!Δ	!^	!^	!L	!L	!R	!R	!V	!V
N	!N	V	VD	R	!^U	L	!Δ	^	!^	Δ	!L	^U	!R	!VD	!V	N

Fig 9, 10 & 11

R : [1100]

	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!N	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
V	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
VD	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
R	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!^U	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
L	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!Δ	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
^	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!^	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
Δ	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!L	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
^U	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!R	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!VD	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!V	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
N	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N

VD : [1101]

	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!N	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
V	!V	!V	!R	!R	!L	!L	!^	!^	!Δ	!Δ	!^U	!^U	VD	VD	!N	!N
VD	!VD	!R	!VD	!R	Δ	!^	Δ	!^	L	!^U	L	!^U	V	!N	V	!N
R	!R	!R	!R	!R	!^	!^	!^	!^	!^U	!^U	!^U	!^U	!N	!N	!N	!N
!^U	^U	!L	Δ	!^	^U	!L	Δ	!^	R	VD	V	!N	R	VD	V	!N
L	!L	!L	!^	!^	!L	!L	!^	!^	VD	VD	!N	!N	VD	VD	!N	!N
!Δ	Δ	!^	Δ	!^	Δ	!^	Δ	!^	V	!N	V	!N	V	!N	V	!N
^	!^	!^	!^	!^	!^	!^	!^	!^	!N	!N	!N	!N	!N	!N	!N	!N
!^	^	!Δ	L	!^U	R	VD	V	!N	^	!Δ	L	!^U	R	VD	V	!N
Δ	!Δ	!Δ	!^U	!^U	VD	VD	!N	!N	!Δ	!Δ	!^U	!^U	VD	VD	!N	!N
!L	L	!^U	L	!^U	V	!N	V	!N	L	!^U	L	!^U	V	!N	V	!N
^U	!^U	!^U	!^U	!^U	!N	!N	!N	!N	!^U	!^U	!^U	!^U	!N	!N	!N	!N
!R	R	VD	V	!N	R	VD	V	!N	R	VD	V	!N	R	VD	V	!N
!VD	VD	VD	!N	!N	VD	VD	!N	!N	VD	VD	!N	!N	VD	VD	!N	!N
!V	V	!N	V	!N	V	!N	V	!N	V	!N	V	!N	V	!N	V	!N
N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N

V : [1110]

	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N
V	V	!N	V	!N	V	!N	V	!N	V	!N	V	!N	V	!N	V	!N
VD	VD	VD	!N	!N	VD	VD	!N	!N	VD	VD	!N	!N	VD	VD	!N	!N
R	R	VD	V	!N	R	VD	V	!N	R	VD	V	!N	R	VD	V	!N
!^U	!^U	!^U	!^U	!^U	!N	!N	!N	!N	!^U	!^U	!^U	!^U	!N	!N	!N	!N
L	L	!^U	L	!^U	V	!N	V	!N	L	!^U	L	!^U	V	!N	V	!N
!Δ	!Δ	!Δ	!^U	!^U	VD	VD	!N	!N	!Δ	!Δ	!^U	!^U	VD	VD	!N	!N
^	^	!Δ	L	!^U	R	VD	V	!N	^	!Δ	L	!^U	R	VD	V	!N
!^	!^	!^	!^	!^	!^	!^	!^	!^	!N	!N	!N	!N	!N	!N	!N	!N
Δ	Δ	!^	Δ	!^	Δ	!^	Δ	!^	V	!N	V	!N	V	!N	V	!N
!L	!L	!L	!^	!^	!L	!L	!^	!^	VD	VD	!N	!N	VD	VD	!N	!N
^U	^U	!L	Δ	!^	^U	!L	Δ	!^	R	VD	V	!N	R	VD	V	!N
!R	!R	!R	!R	!R	!^	!^	!^	!^	!^U	!^U	!^U	!^U	!N	!N	!N	!N
!VD	!VD	!R	!VD	!R	Δ	!^	Δ	!^	L	!^U	L	!^U	V	!N	V	!N
!V	!V	!V	!R	!R	!L	!L	!^	!^	!Δ	!Δ	!^U	!^U	VD	VD	!N	!N
N	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N

Fig 12, 13 & 14

!N : [1111]

	N	!V	!VD	!R	^U	!L	Δ	!^	^	!Δ	L	!^U	R	VD	V	!N
!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N
V	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N
VD	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N
R	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N
!^U	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N
L	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N
!Δ	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N
^	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N
!^	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N
Δ	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N
!L	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N
^U	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N
!R	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N
!VD	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N
!V	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N
N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N	!N

Fig 15

We have already shown that we can do the basic rules of arithmetic with these gates (like bringing terms across the equals sign), so we should be able to do more complicated affairs, like addition, subtraction and/or multiplication and division. For example, we can try to rewrite R Δ L to produce (!^));

R Δ L = Δ

(R) Δ (Δ) = ((VD) Δ (!^))

(R) Δ (Δ) / ((VD) Δ) = (!^)

Another more convoluted example, this time rewriting ((Δ) ^ (!L)) for (VD):

(Δ) ^ (!L) = ^U

((Δ) ^ (^U)) V ((Δ) V (^U)) = ^U

!(((Δ) ^ (^U)) V ((Δ) V (^U))) = !^U

(!Δ)^(!^U) V (!Δ)V(!^U)/(!Δ) ^ = VD

Later on, we will apply these simple methods to Set Theory, as a whole.

Some Notable Gates

In previous research papers we have come across Dimensional Gate Operators like Δ (XOR), which represents the rules of Real Number Arithmetic and !Δ (XNOR), which represents the rules of Complex Arithmetic. This shift in how we do arithmetic allows us to go from the 1-2 dimensional lines of the Real Numbers and into the 2-3 dimensional realm of the ‘Complex Plane’.

Unlike, the numbers of the Complex Plane, however, there are no algebraic numbers here. This is because the $\sqrt{-1}$ (or *i*) is equal to ±1 in !Δ. Therefore, they are simply ordinary numbers with a different arithmetic rule set. You might be tempted to think — as I did — that Δ also governs the rules of the Quaternions and Octonions. But this is not the case.

Unlike, the numbers of the Complex Plane, however, there are no algebraic numbers here. This is because the $\sqrt{-1}$ (or *i*) is equal to ±1 in !Δ. Therefore, they are simply ordinary numbers with a different arithmetic rule set. You might be tempted to think — as I did — that Δ also governs the rules of the Quaternions and Octonions. But this is not the case.

One important feature of the Quaternions is that they are non-commutative; that is A(B) =/= B(A).

Where do we see this feature in the DGO?

We see it in places like !VD, !R, !^U, and VD, where A(-B) =/= B(-A). Quaternions are governed by a mix of !VD and ^U logic, because they are non-commutative for different valued quaternions, whilst retaining the !Δ rule set of the imaginary numbers, when values *i*, *j*, *k* are the same:

$$i^2 = j^2 = k^2 = -1$$

$$ij = k, ji = -k$$
$$jk = i, kj = -i$$
$$ki = j, ik = -j$$

0	0	0
0	1	1
1	0	1
1	1	0

0	0	0
0	1	0
1	0	1
1	1	0

0	0	0
0	1	1
1	0	0
1	1	0

Left to Right: Δ, ^U, and !VD

The Quaternions are made from two groups of !VD and one of ^U. This asymmetry allows for the loop to be closed and also explains why these higher-dimensional algebras can only be order 2^n.

Dimensional Spaces

Let's return to Fig 0-15. What do you notice about these 16 graphs? Other than how great they all look… That's right. It is clear that some of them fit together like jigsaw pieces. For instance, it is clear that Fig 1, 2, 4, & 8 form a 'Set B'. Similarly, we can make 'Set C' from the figures 7, 11, 13 and 14.

The horizontal and vertical lines appear to make another set, so we will call that 'Set D'. That just leaves the final four graphs; {0, 6, 9, 15} or 'Set A'.

There is a distinct pattern emerging in how these sets are dispersed across the entire binary number line, here represented in their decimal form:

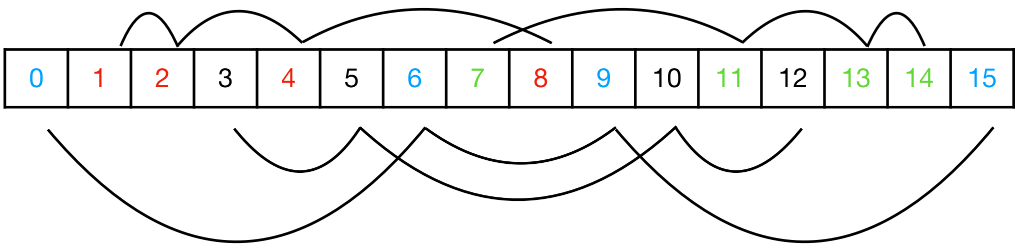


Fig 16: All four sets; A, B, C, & D. This pattern explains the frequency of the NOT values, as they are represented in the DGO.

The first composite graph (Fig 17), shows the domains of N (in the bottom lefthand corner), Δ immediately above that, !Δ in the lower righthand corner and !N in the top right.

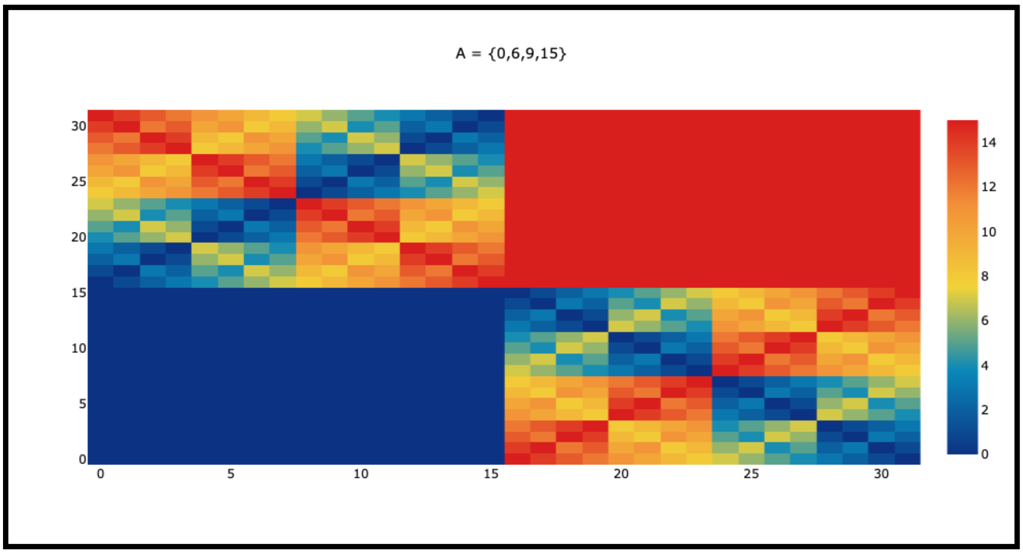


Fig 17: A = {N, Δ, !Δ, !N}

What matters here is not the order, so much, as the grouping and what this tells us. Set A can be rewritten as {N, Δ, !Δ, !N}, which stands for the paths between Dimensions 0, 2, 3 and 1 respectively. To understand what I mean by this, look at our Quaternion logic gates: ^U and !VD.

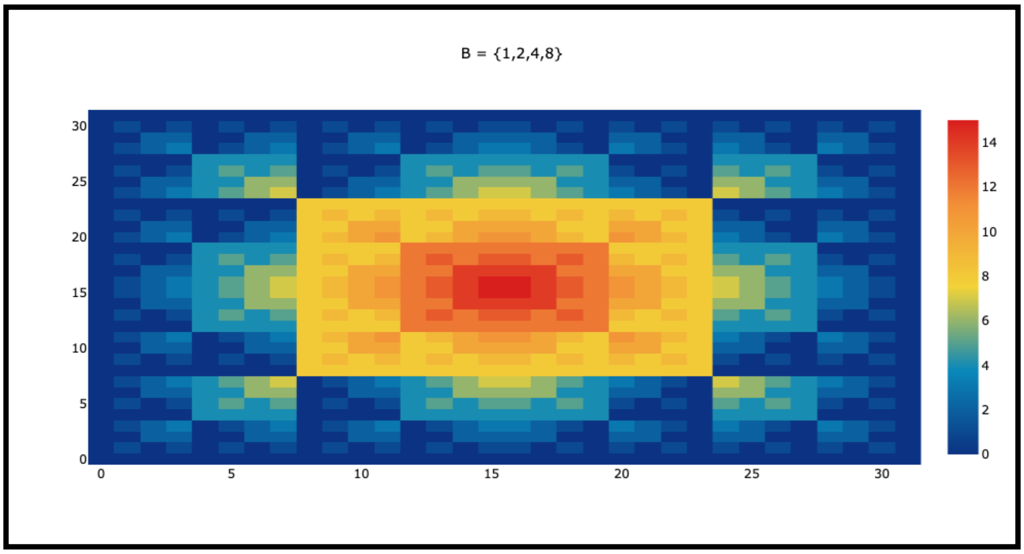


Fig 18: Set B = {!^, !^U, !VD, !V}

If we multiply these terms by ¬Δ (or Imaginary rule set) we get (!VD) ¬Δ (!^U) = Δ (our Real number logic). This shows the path by which Quaternions collapse down into the rule set of both the Imaginary and Real numbered spaces of dimensions 2-3. We can then continue this process, in the usual manner, multiplying (Δ) !Δ (!Δ) and (Δ) Δ (!Δ) to get the dimensions beneath them:

(Δ) !Δ (!Δ) = N

(Δ) Δ (!Δ) = !N

Leading us to conclude that N and !N refer to dimensions 0 and 1, respectively. In this sense, the Real Numbers and Imaginary Numbers live in Set A, along with dimensions 1 and 0. Whereas the Quaternions live jointly in Set B and C, along with some other more traditional logic gates. The Octonions, Sedenions, and (potentially) other higher dimensional spaces exist scattered among the other sets, although this is something I have to investigate further.

While this way of thinking about logic gates could provide a method for travelling from one rule set (i.e. dimensional space) to the other, it should not be taken too literally. In one sense, applying the Δ or !Δ rule set to the

Quaternions does not lead to Δ or !Δ, but rather to hybrid spaces; the Real Quaternions and the Imaginary Quaternions, and neither of these two systems cancel out to Dimension 0 when summed in Δ. More on this in up-coming research.

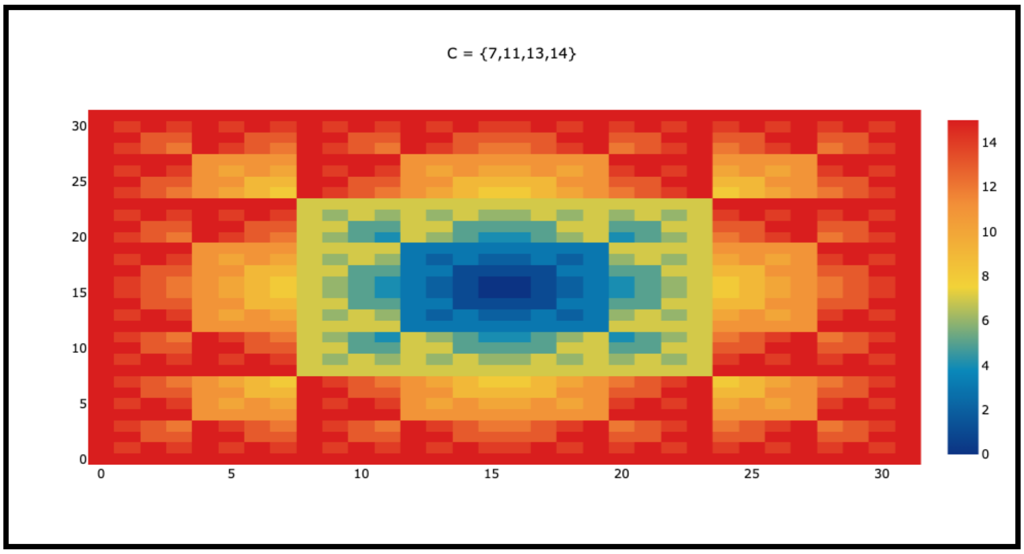


Fig 19: Set C = {V, VD, ^U, ^}

Based on the arrangements of the graphs in Fig 17, we can see that it starts in the lower lefthand corner and proceeds in a zigzag fashion to the upper-right. The graphs in Fig 18 go in the reverse direction and the pattern is repeated in the next two graphs in Fig 19 & 20.

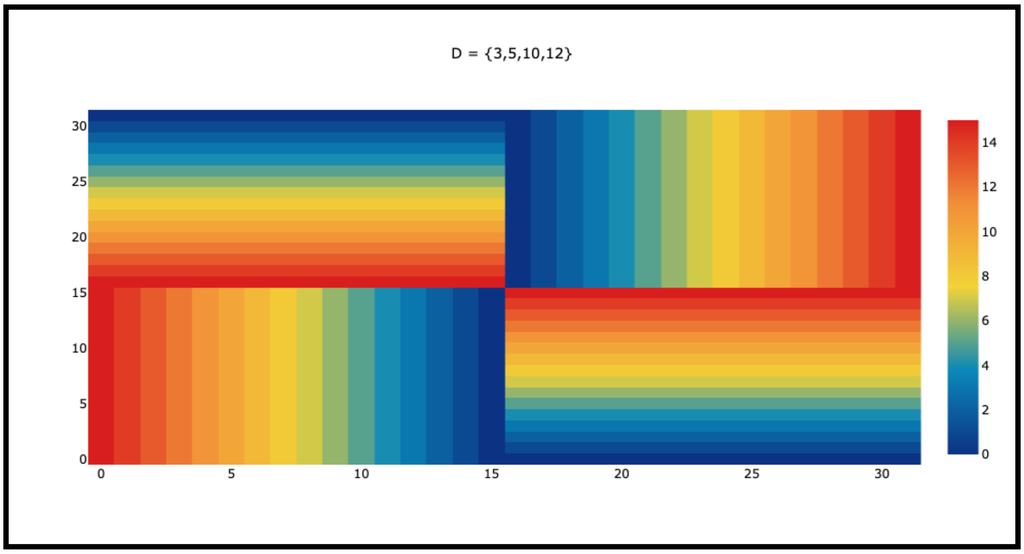


Fig 20: Set D= {!R, !L, L, R}

Using the pattern of Fig 17 as the 'base arrangement', we can then arrange Fig 17, 18, 19 & 20 into a single graph (Fig. 21). But there are 16! possible arrangements.[4] How do we know we have the right one?

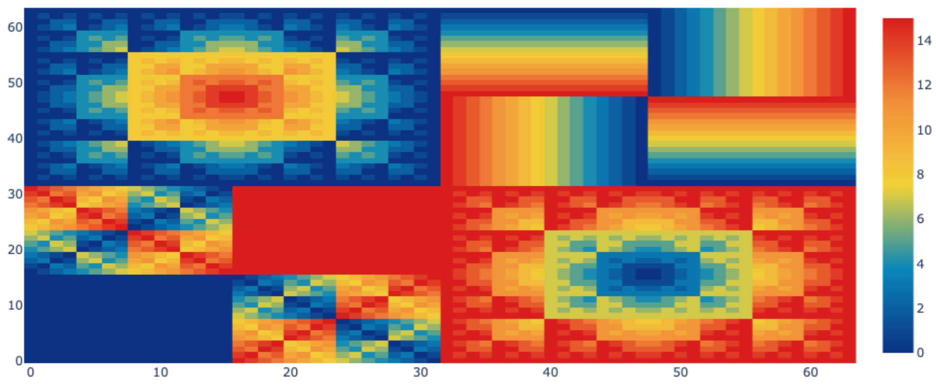


Fig 21: Preliminary grouping of sets.

We don’t. But we can deduce from a lack of connectivity between the different regions that a more accurate arrangement is possible. We shouldn’t expect to see such harsh delineating lines from such inter-related sets. Earlier it was stated that the Quaternions and the Imaginary Quaternions follow the rules laid out in Set B, while the Real Quaternions are sitting in Set C. This can’t be right. If we move the Real Quaternions into Set B with the other Quaternions, this forces all of the more commonly known logic gates: ^, !^, V and !V together, which is much neater.

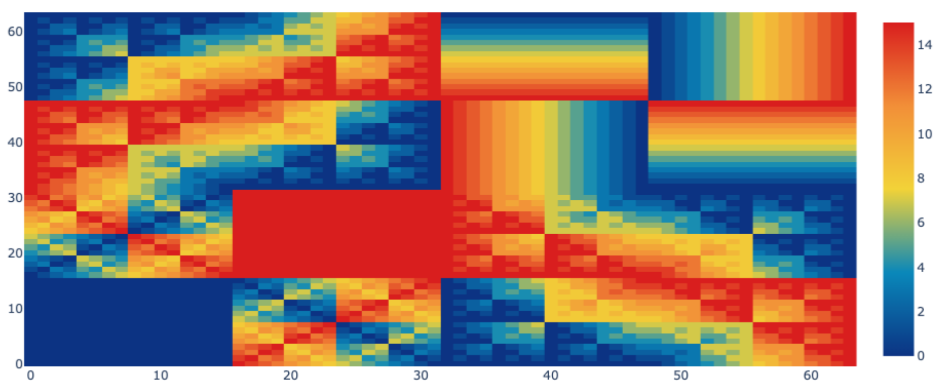


Fig 21: Final grouping.

Now, we see a much greater connectivity between different regions. At this point, we are basically making pretty pictures without much substantive

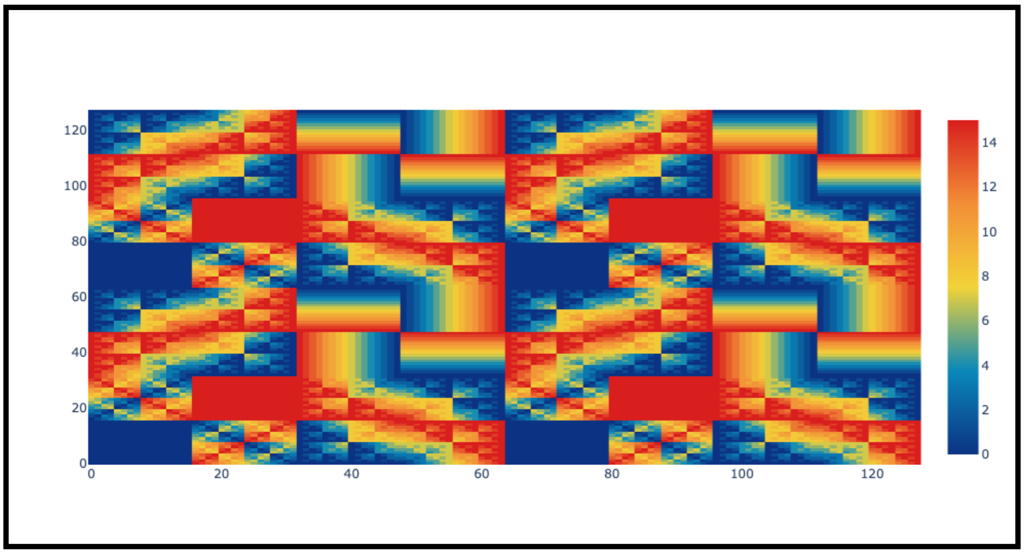


Fig 22: A more tiled version of Fig. 21.

conclusions. However, I think it was worthwhile to explore this line of inquiry, as it served as a useful vehicle to explain some of the relationships between the Reals, Imaginary, Quaternions and Logic Gates.

Set Theory Arithmetic

Since we have already shown that is possible (in principle, at least) to divide and multiply set operators together, I'm wondering if it is possible to apply these kinds of operations to the actual sets of Set Theory. Unlike traditional methods of arithmetic with sets e.g. *dividing the elements of one set into another*, here we will be dividing the operators acting on the sets. Why would anyone want to do this, you might ask?

It is a good question and one that I can only speculate on.

Using DGO methods on Set Theory will open up new ways to get from one partition of a group of sets to another, and will lead to operations which were not previously possible under the former laws of the Boolean Algebra.

Why should this be so?

Doesn't the current operators cover all aspects and combinations of Set Theory? The answer to this is: *Yes, it does*. But, if we recall that the !VD operator governs the arithmetic of the quaternions, then this opens up the possibility of doing Set Theory in higher order dimensions.

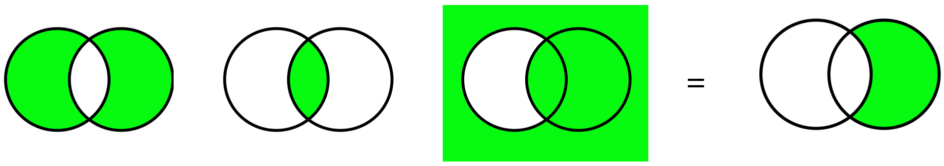
Why would we need to do this?

That is not so clear. We might find that quaternions are indispensable for describing some aspect of Quantum Mechanics (see my various up-coming papers, keywords: ‘Quaternions’, ‘Quarks’ and ‘Polyhedra’ for more on that), in which case describing sets of these particles via the rules of the 4-dimensional space in which they live might be desirable. Alternatively, 4D Set Theory might find application in the weird and wonderful of Quantum Computing, where a bit can be a ‘1’ and ‘0’ at the same time.

Let’s begin with a simple example. We will define our sets; A and B:

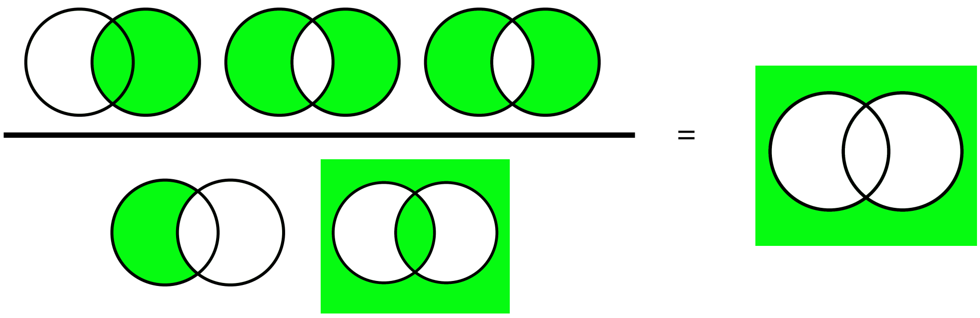
(A Δ B) ^ (A !L B) = (A ^U B)

This operation would correspond with the following Venn Diagrams:



A slightly more complex operation would be the following:

(A R B) Δ (A !Δ B) / (!VD)!Δ = (A !V B)



An important aspect of the above two equations is that both of them feature non-commutative logic gates, specifically: !L, R and !VD. This means that different outcomes for the equation would be expected depending on which set we choose to be on the *right* and which appears on the *left*.

Since sets have no particular orientation in space and since either one can be on the ‘left’ or on the ‘right’, this poses something of a problem, especially as we move into higher-dimensions. Or at least it would, if we were expecting to form a closed space algebra from it, but as we shall see in the follow-up preprints and research papers, creating closed field algebras might not always be the best approach…

Conclusion

Dimensional Logic Gate Operators have the potential for some very weird applications. I would reiterate Wildberger's proposal that they could find application in computer circuitry, set theory and Bool algebra, and add to it Quantum Computing, 4-dimensional (and higher) Set Theory and Set Theory Arithmetic.

Citations and Footnotes

[1] www.researchgate.net/publication/346527686_REIMAGINING_COMPLEX_NUMBERS
[2] Implication and 16 logical operations | Math Foundations 258 | N J Wildberger: https://www.youtube.com/watch?v=XkqmuUg_yFs (Wildberger's video is dated December 22 2018.)
[3] Wildberger does not appear too concerned with this, as he is evidently more taken by the use of 'Bool Algebra' rather than 'Boolean Algebra', which is "more practical for sets with multiple components". [2]
[4] That is *20,922,789,888,000* possible permutations.

Appendix

N

0	0	0
0	1	0
1	0	0
1	1	0

!V

0	0	1
0	1	0
1	0	0
1	1	0

!VD

0	0	0
0	1	1
1	0	0
1	1	0

!R

0	0	1
0	1	1
1	0	0
1	1	0

^U

0	0	0
0	1	0
1	0	1
1	1	0

!L

0	0	1
0	1	0
1	0	1
1	1	0

Δ

0	0	0
0	1	1
1	0	1
1	1	0

!^

0	0	1
0	1	1
1	0	1
1	1	0

^

0	0	0
0	1	0
1	0	0
1	1	1

!Δ

0	0	1
0	1	0
1	0	0
1	1	1

L

0	0	0
0	1	1
1	0	0
1	1	1

!^U

0	0	1
0	1	1
1	0	0
1	1	1

R

0	0	0
0	1	0
1	0	1
1	1	1

VD

0	0	1
0	1	0
1	0	1
1	1	1

V

0	0	0
0	1	1
1	0	1
1	1	1

!N

0	0	1
0	1	1
1	0	1
1	1	1

The colour-coding here pertains to the different sets: A = Red, B = Green, C = Blue, and D = Yellow. The Quaternions are created from !VD and ^U and the Real Quaternions are created from !^U and VD.