Refutation of quantum logic as tautologous

We assume the method and apparatus of Meth8/VŁ4 with `T`autology as the designated *proof* value, `F` as contradiction, `N` as truthity (non-contingency), and `C` as falsity (contingency). Results are a 16-valued truth table in row-major and horizontal.

Section 1.

LET bk_ "| 〉" bra-ket;
p, q, r, s: |0〉 bk_0, |1〉 bk_1, 2^0.5 (√2), |+〉 bk_+; ~s | -〉 bk_- ;
~ Not; & And; + Or; - Not Or; > Imply.

From: Wright, J. (2015). Lecture 2: Quantum math basics.
cs.cmu.edu/~odonnell/quantum15/lecture02.pdf

|+〉 = 1/√2 |0〉 + 1/√2 |1〉 is rewritten as √2 |+〉 = |0〉 + |1〉 (1.1)

(r&s)=(p+q) ; `TFFF TFFF TFFF FTTT` (1.2)

|-〉 = 1/√2 |0〉 - 1/√2 |1〉 is rewritten as √2 |-〉 = |0〉 - |1〉 (2.1)

(r&~s)=(p-q) ; `FTTT TFFF FTTT FTTT` (2.2)

We ask: Does the positive sign qubit (Eq. 1) imply the negative sign qubit (Eq. 2), as its conjugate, as a theorem? (3.1)

((r&s)=(p+q))>((r&~s)=(p-q)) ; `FTTT TTTT FTTT TTTT` (3.2)

In Section 1, Eqs. 3.2 as rendered is *not* tautologous. This means the implication operator for quantum logic is refuted and by extension, so also quantum logic.

Consequently, we evaluate a less technical description of quantum logic aimed for a different audience.

Section 2.

From:
medium.com/@decodoku/quantum-computation-with-the-simplest-maths-possible-c23ff6563964

LET p, q, r, s, t: up (upness), down (downness), overlap, S (superposition);
% possibility, for one or some; # necessity, for all or every;
(%p>#p) truthity, ordinal 1; (%p<#p) falsity, not ordinal 1, such as ordinal 0.

Quantum Computation with the simplest maths possible

... [I]t would be useful to have some way of quantifying how similar two states are. We'll call this the *overlap*. The states **up** and **down** are completely different, so these should have an overlap of 0 (this is the actual number zero this time). For states that are 100% the same, let's say that the overlap is 1.

For the two states **up** and **down**, there are only four possible overlaps to calculate and we know what they should be already.

overlap of **up** and **up** = 1 (1.1)

(r&(p&p))=(%p>#p) ; CCCC CNCN CCCC CNCN (1.2)

overlap of **up** and **down** = 0 (2.1)

(r&(p&q))=(%p<#p) ; NNNN NNNC NNNN NNNC (2.2)

overlap of **down** and **up** = 0 (3.1)

(r&(q&p))=(%p<#p) ; NNNN NNNC NNNN NNNC (3.2)

overlap of **down** and **down** = 1 (4.1)

(r&(q&q))=(%p>#p) ; CCCC CCNN CCCC CCNN (4.2)

Now we need to work out overlaps for superposition states. There are many different possible superpositions of up and down, which differ by how biased they are towards one or the other. This means we need two numbers, let's call them the upness and downness, that describe how much up and down there is in a superposition.

It would also be nice to have a shortened name for the superposition state that we are trying to describe. Let's just call it **S**. Now we need to write down the fact that **S** is a superposition of **up** and **down** and also what its upness and downness are, in a way that looks mathsy. How about

S = (upness of **S**) × **up** + (downness of **S**) × **down** (5.1)

s=(((p&s)&p)+((q&s)&q)) ; TTTT TTTT FTTT FTTT (5.2)

This nicely puts all the required information on one line. It even has has an + and some ×'s in to make it look like maths. These look suspiciously like addition and multiplication. But what does it even mean to multiply a state by a number? Or to add two states? These aren't the addition and multiplication that we are used to. It will turn out that they will follow similar rules to the normal ones, though. So that's why we use these symbols.

Now, what is the overlap between our superposition state S and the state up? We still haven't made up enough rules to actually calculate this, so we have to choose something. We have just introduced the notion of upness, which is how much **up** there is in **S**. This seems to be pretty much the same thing as the overlap between **S** and **up**, and it wouldn't contradict any of the rules we have already if they were the same thing. So let's just make up the rule that says they are the same thing.

overlap of **S** and **up** = upness of **S** (6.1)

(r&(s&p))=(p&s) ; TTTT TTTT TFTF TTTT (6.2)

There's a more complicated way we can write this, that can help us understand a little more about what is going on.

overlap of **S** and **up** = (upness of **S**) × (overlap of **up** and **up**)

+ (downness of **S**) × (overlap of **down** and **up**) (7.1)

(r&(s&p))=(((p&s)&(r&(p&p)))+((q&s)&(r&(q&p)))) ;

`TTTT TTTT TTTT TTTT` (7.2)

Here the overlap of **S** and **up** is a sum of two things. The first is the contribution from the **up** part of **S**

(upness of **S**) × (overlap of **up** and **up**) = (upness of **S**) × 1 = upness of **S** (8.1)

(((p&s)&(r&(p&p)))=((p&s)&(%p>#p)))=(p&s) ; `FFFF FFFF FCFC FNFN` (8.2)

This tells us that the **up** part of **S** contributes the upness (obviously), and it contributes it fully because the overlap between the **up** part of **S** and **up** is 1.

The second contribution is from the down part of **S**

(downness of **S**) × (overlap of **down** and **up**) = (downness of **S**) × 0 = 0 (9.1)

(((q&s)&(r&(q&p)))=((q&s)&(%p<#p)))=(%p<#p) ;

`CCCC CCCC CCFF CCFT` (9.2)

This tells us that the down part of **S** would contribute the downness if it contributed anything. But it doesn't actually contribute it because the overlap between the **down** part of **S** and **up** is 0.

We get a similar equation for the overlap of **S** and **down**.

overlap of **S** and **down** = (upness of **S**) × (overlap of **up** and **down**)

+ (downness of **S**) × (overlap of **down** and **down**) (10.1)

(r&(s&q))=(((p&s)&(r&(p&q)))+((q&s)&(r&(q&q)))) ;

`TTTT TTTT TTTT TTTT` (10.2)

This time the overlaps of **up** and **down** ensure that the downess contributes fully, and the upness doesn't contribute at all.

What about the overlap with something else? If we look at the overlap between **S** and **down**, and the overlap for **S** and **up**, the only difference is that one has **up** in and the other has **down**. So maybe we can just replace that with anything else too. Let's invent a new state and call it **T**, for no other reason but it coming after **S** in the alphabet. The overlap of **S** and **T** is then

overlap of **S** and **T** = (upness of **S**) × (overlap of **up** and **T**)

+ (downness of **S**) × (overlap of **down** and **T**) (11.1)

(r&(s&t))=(((p&s)&(r&(p&t)))+((q&s)&(r&(q&t)))) ;

`TTTT TTTT TTTT TTTT,   FTTT TTTT TTTT TTTT` (11.2)

In these equations we have × and +, multiplying and adding normal numbers. These are indeed the multiplication and addition that we are used to. From these equations you can maybe see why I used × and **+** before. Compare the equation for **S** with the equation for its overlap with **T**

S = (upness of **S**) × **up** **+** (downness of **S**) × **down**

s=(((p&s)&p)+((q&s)&q)) ; `TTTT TTTT FTTT FTTT` (12.1)

overlap of **S** and **T** = (upness of **S**) × (overlap of **up** and **T**)

+ (downness of **S**) × (overlap of **down** and **T**)

(r&(s&t))=(((p&s)&(r&(q&t)))+((q&s)&(r&(q&t)))) ;

`TTTT TTTT TTTT TTTT,   TTTT TTTT TTTT FTTT` (12.2)

These are pretty much the same. The only difference is that each state in the first one has been replaced by the overlap of that state and T in the second. This means that the second one just has normal numbers in. So the weird multiplication and addition in the first one become normal in the second. So, whatever **x** and **+** are, they must be some version of multiplication and addition that work with the states of qubits, and just become normal multiplication and addition once we just start calculating with numbers. We won't need to think much more about this, though.

Let's think more about the overlap between **S** and our new state **T**. Firstly, just like **S** we should be able to write **T** as

T = (upness of **T**) × **up** **+** (downness of **T**) × **down** (13.1)

t=(((p&t)&p)+((q&t)&q)) ;

`TTTT TTTT TTTT TTTT,   FTTT FTTT FTTT FTTT` (13.2)

Earlier we made a rule that the upness of a state is the same thing as its overlap with **up**. This rule lets us write the equation for the overlap of **S** and **T** in a simpler way.

overlap of **S** and **T** = (upness of **S**) × (upness of **T**)

+ (downness of **S**) × (downness of **T**) (14.1)

(r&(s&t))=(((p&s)&(p&t))+((q&s)&(q&t))) ;

`TTTT TTTT TTTT TTTT,   TTTT TTTT TFFF FFFT` (14.2)

This lets us work out the overlap of **S** and **T** using their upness and doweness, which are just numbers that we know.

Now let's ask a question for which we already know the answer. What is the overlap between **S** and itself? Using the maths above

overlap of **S** and **S** = (upness of **S**) × (upness of **S**)

+ (downness of **S**) × (downness of **S**)

= (upness of **S**)^2 + (downness of **S**)^2 (15.1)

((r&(s&s))=(((p&s)&(p&s))+((q&s)&(q&s))))=

(((p&s)&(p&s))&((q&s)&(q&s))) ; `FFFF FFFF FTTF TFFT` (15.2)

Since we are looking at the overlap between two states that are exactly the same, the answer should come out to be 1. So now we know something about the relationship between the upness and downness for any quantum superposition

$$\text{upness}^2 + \text{downness}^2 = 1 \quad (16.1)$$

((p&p)+(q&q))=(%p>#p) ; `CNNN CNNN CNNN CNNN` (16.2)

This makes a lot of sense. The more a state is biased towards up, the less it must be biased towards down. For example, a state with an upness of 1 (and so with an upness^2 of 1 too) is completely up, and so has no downness. The first concrete fact that our quantum maths has told us isn't weird at all. See, quantum mechanics isn't so strange.

Well, maybe it is a little bit strange. Note that we don't just add upness and downness here. Instead we square them first. One thing we know from school is that negative numbers square to the same value as positive ones. $(-1)^2 = 1$ just like $1^2 = 1$, for example. So maybe this equation is telling us that its okay for the upness and downness to be negative, even though this would be a bit weird, because these numbers only need to be sensible after we've squared them.

In Section 2, Eqs. 7.2 and 10.2 (2 of 16) as rendered are tautologous with the others not. This confirms the conclusion from Section 1 that quantum logic is *not* tautologous.