Foundations for Quantum Computation and Quantum Information

Jake Xuereb

Last Updated - November 7, 2019

Abstract

During the academic year of 2018-2019 I became critically aware of the importance of quantum information in Physics. I mean I thought it was cool before, because of all of the great things quantum computing brings with it, but the far reaching ideas about the foundations of the universe we inhabit, which it brings with it is something I did not expect. Ideas of this sort, making use of quantum information as their basis, are relativistic quantum information, quantum metrology and entanglement space time to mention a few. These ideas shake me to my core and I want to see where they can take me.
To this end I'm reading a number of things on the subject. These are the notes I take whilst I read.

1 Tensor Products & Entanglement

1.1 Mathematical & Physical Motivations for the Tensor Product

A Conceptual Stance To describe singular quantum states we use singular vector spaces composed of their own orthonormal eigenbasis but what of multiple particles? Specifically, let us at first limit our view to two particles.
Using the introduction that is given by Zwiebach in [4], consider a Particle 1 whose phase space is described by a complex inner product space V with operators $T_1, T_2, \ldots$ and Particle 2 with similar space W and operators $S_1, S_2, \ldots$ .To describe the two particles in conjunction to one another one might be inclined to do as we do in mathematics with Cartesian Products and leading to ordered pairs $(v, w) \in V \times W$. Now we're not quite there yet what we wish to define is a new vector space $V \otimes W$ with elements denoted $v \otimes w$ which start out in $V \times W$ but then are surjectively mapped onto $V \otimes W$ using the two physically motivated rules, which in mathematics refers to bilinear composition.

1. Scaling either states of the states v, w is equivalent to scaling both states.

$$(av) \otimes w = v \otimes (aw) = a\,(v \otimes w)\,, \quad a \in \mathbb{C}.$$

2. If one of the two states is a superposition state so must be the tensor product of the two states. From a mathematical perspective this refers to the distributive nature of the tensor product over addition. From a physical point of view we will see that this means that if one of two particles is in a superposition state then the two-particle system is in a superposition state also. The astute reader will liken this property to that innate of homomorphisms.

$$(v_1 + v_2) \otimes w = v_1 \otimes w + v_2 \otimes w \qquad v \otimes (w_1 + w_2) = v \otimes w_1 + v \otimes w_2.$$

For V with basis $\{v_1, v_2, \dots v_n\}$ and W with basis $\{w_1, w_2, \dots w_m\}$ their tensor product $V \otimes W$ would be built up from basis elements $v_i \otimes w_j$ giving a basis looking $\{v_1 \otimes w_1, \dots, v_1 \otimes w_m, v_2 \otimes w_1, \dots, v_2 \otimes w_m, \dots \dots v_n \otimes w_1, \dots, v_n \otimes w_m\}$ visibly showing that

$$\dim V \otimes W = \dim V \times \dim W.$$

For the sake of pedantry consider $a \in V$ and $b \in W$. Their tensor product $a \otimes b \in V \otimes W$ is clearly a linear combination of the basis $v_i \otimes f_j$ since

$$a \otimes b = \sum_{i=1}^{n} (a \cdot v_i)\, v_i \otimes \sum_{j=1}^{m} (b \cdot w_j)\, w_j = \sum_{i,j} (a \cdot v_i)\, (b \cdot w_j)\, v_i \otimes w_j$$

clearly showing that $a \otimes b$ is a linear combination of $v_i \otimes w_j$.

An Operator defined to act on $V \otimes W$ must itself be a tensor product of operators defined on the subspaces V and W respectively. So for two operators $T \in \mathcal{L}(V)$ and $S \in \mathcal{L}(W)$, their tensor product $T \otimes S \in \mathcal{L}(V \otimes W)$. Operators in a tensor product space act on elements of this space, themselves tensor products in the following logical way.

$$T \otimes S\,(v \otimes w) = Tv \otimes Sw$$

where each operator acts on the subspace upon which it is defined and the tensor product of the new vectors in V and W is used to calculate the new tensor product. A tensor product operator which acts on only one of two component states in a bipartite state maybe constructed using the identity operator $\mathcal{I}$. Such operators are called *Upgraded* operators in the sense that if T is an operator which works in V then $T \otimes \mathcal{I}$ is the upgraded version of the operator which does the same thing as T but in the tensor space $V \otimes W$.

$$H_1 \otimes H_2 \neq H_1 \otimes \mathcal{I} + \mathcal{I} \otimes H_2.$$

Let H represent a hamiltonian in a phase space of one of two systems. On the right we are thinking of both systems together but there is no mixing between the two system and hence the sum of upgraded operators. On the left we have mixing.

A Pragmatic and Literal Stance with Examples Having understood the conceptual aspects of the tensor product we now ground it in reality by understanding how the tensor product is calculated. As described earlier the tensor product of two states is quite a bit like a cartesian product with the added aspect of multiplication so let's try this out.

Consider the states $|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}$ and $|1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$ where $|0\rangle$ and $|1\rangle$ correspond to the eigenkets of a two state (ex: Spin-$\frac{1}{2}$) system. The cartesian product $(\times)$ would looks like this

$$|0\rangle \times |1\rangle = \begin{bmatrix}(1,0)\\(1,1)\\(0,0)\\(0,1)\end{bmatrix}$$

and now multiplying the entries of the ordered pairs one gets what is known as the Kronecker Product which actualises the tensor product a matrix operation.

$$|0\rangle \otimes |1\rangle = \begin{bmatrix}(1)(0)\\(1)(1)\\(0)(0)\\(0)(1)\end{bmatrix} = \begin{bmatrix}0\\1\\0\\0\end{bmatrix}.$$

Great. We're now in good shape to try to describe a system comprised of two spin-$\frac{1}{2}$ particles. Let's call the state space of the first particle V_1 and of the second V_2 each with basis vectors $|\uparrow\rangle$ and $|\downarrow\rangle$. Tensor product space $V_1 \otimes V_2$ will have 4 basis vectors

$$|\uparrow_1\rangle \otimes |\uparrow_2\rangle \qquad |\uparrow_1\rangle \otimes |\downarrow_1\rangle \qquad |\uparrow_2\rangle \qquad |\downarrow_1\rangle \otimes |\downarrow_2\rangle$$

and considering the actual vector formulation allows us to calculate these new 4 dimensional basis vectors as

$$\begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} \qquad \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} \qquad \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} \qquad \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix}$$

giving

$$\begin{bmatrix}1\\0\\0\\0\end{bmatrix} \qquad \begin{bmatrix}0\\1\\0\\0\end{bmatrix} \qquad \begin{bmatrix}0\\0\\1\\0\end{bmatrix} \qquad \begin{bmatrix}0\\0\\0\\1\end{bmatrix}.$$

As such any state of the two particle system can be described as linear combination of these four vectors such that $|\psi\rangle \in V_1 \otimes V_2$ can be written as

$$|\psi\rangle = a_{11}\,|\uparrow_1\rangle \otimes |\uparrow_2\rangle + a_{12}\,|\uparrow_1\rangle \otimes |\downarrow_2\rangle + a_{21}\,|\downarrow_1\rangle \otimes |\uparrow_2\rangle + a_{22}\,|\downarrow_1\rangle \otimes |\downarrow_2\rangle$$

where $a_{ij} = \langle\psi|\,(|\uparrow_i\rangle \otimes |\downarrow_j\rangle) \in \mathbb{C}$ but we need to know how inner products work in tensor product spaces first, which gives me quite the neat segway to move forward.

Inner Products in Tensor Product Spaces The inner product of a bra and ket is of course as we know it, but in the tensor product of two state spaces $V \otimes W$ how is this going to work? Say I have

$$\langle v_1 \otimes w_1 | v_2 \otimes w_2 \rangle$$

where we are taking the inner product of $|v_1 \otimes w_1\rangle$ and $|v_2 \otimes w_2\rangle$. Now much like the case when we were figuring out how operators work, each piece of this product will interact with the piece which lives in its own space giving

$$\langle v_1 \otimes w_1 | v_2 \otimes w_2 \rangle = \langle v_1 | v_2 \rangle \langle w_1 | w_2 \rangle$$

where $\langle v_1 | v_2 \rangle$ is an inner product in V and $\langle w_1 | w_2 \rangle$ is an inner product in W.

Example We wish to find the normalised state in $V_1 \otimes V_2$ with zero total z and x-components, S_z^T and S_x^T of spin angular momentum. We will proceed by first finding the form of the state with zero total z-component and then find which the form of this state with zero x-component.

Our starting point is the generalised bipartite state

$$|\psi\rangle = a_{11} |\uparrow_1\rangle \otimes |\uparrow_2\rangle + a_{12} |\uparrow_1\rangle \otimes |\downarrow_2\rangle + a_{21} |\downarrow_1\rangle \otimes |\uparrow_2\rangle + a_{22} |\downarrow_1\rangle \otimes |\downarrow_2\rangle$$

To begin with, we remember that operators in tensor product spaces are upgraded and summed so as to enact on separate spaces in a way which which is mindful of both spaces. If the total z-component of angular momentum of a bipartite system is $S_{T_z} \in V_1 \otimes V_2$ and this can be represented as

$$S_{T_z} = S_{z_1} \otimes \mathcal{I} + \mathcal{I} \otimes S_{z_2}$$

Recall that S_z is characterised by the eigenket-eigenvalue relation $S_z |\uparrow\rangle = \frac{\hbar}{2} |\uparrow\rangle$ and $S_z |\downarrow\rangle = -\frac{\hbar}{2} |\downarrow\rangle$. Beginning the calculation

$$S_{T_z}(|\psi\rangle) = (S_{z_1} \otimes \mathcal{I} + \mathcal{I} \otimes S_{z_2}) |\psi\rangle = (S_{z_1} \otimes \mathcal{I}) |\psi\rangle + (\mathcal{I} \otimes S_{z_2}) |\psi\rangle$$

and working out the first bit

$$\begin{aligned}
(S_{z_1} \otimes \mathcal{I}) |\psi\rangle &= (S_{z_1} \otimes \mathcal{I}) (a_{11} |\uparrow_1\rangle \otimes |\uparrow_2\rangle + a_{12} |\uparrow_1\rangle \otimes |\downarrow_2\rangle + a_{21} |\downarrow_1\rangle \otimes |\uparrow_2\rangle + a_{22} |\downarrow_1\rangle \otimes |\downarrow_2\rangle) \\
(S_{z_1} \otimes \mathcal{I}) |\psi\rangle &= a_{11} S_{z_1} |\uparrow_1\rangle \otimes \mathcal{I} |\uparrow_2\rangle + a_{12} S_{z_1} |\uparrow_1\rangle \otimes \mathcal{I} |\downarrow_2\rangle + a_{21} S_{z_1} |\downarrow_1\rangle \otimes \mathcal{I} |\uparrow_2\rangle + a_{22} S_{z_1} |\downarrow_1\rangle \otimes \mathcal{I} |\downarrow_2\rangle \\
(S_{z_1} \otimes \mathcal{I}) |\psi\rangle &= \frac{\hbar}{2} (a_{11} |\uparrow_1\rangle \otimes |\uparrow_2\rangle + a_{12} |\uparrow_1\rangle \otimes |\downarrow_2\rangle - a_{21} |\downarrow_1\rangle \otimes |\uparrow_2\rangle - a_{22} |\downarrow_1\rangle \otimes |\downarrow_2\rangle) .
\end{aligned}$$

Similarly, the second bit will give us

$$(\mathcal{I} \otimes S_{z_2}) |\psi\rangle = \frac{\hbar}{2} (a_{11} |\uparrow_1\rangle \otimes |\uparrow_2\rangle - a_{12} |\uparrow_1\rangle \otimes |\downarrow_2\rangle + a_{21} |\downarrow_1\rangle \otimes |\uparrow_2\rangle - a_{22} |\downarrow_1\rangle \otimes |\downarrow_2\rangle) .$$

Putting everything together we have

$$S_z^T |\psi\rangle = \hbar (a_{11} |\uparrow_1\rangle \otimes |\uparrow_2\rangle - a_{22} |\downarrow_1\rangle \otimes |\downarrow_2\rangle) .$$

From this result is clear that S_z^T is 0 for $|\uparrow_1\rangle \otimes |\downarrow_2\rangle$ and $|\downarrow_1\rangle \otimes |\uparrow_2\rangle$ so now to find the state with zero S_x^T consider

$$|\psi\rangle = a_{12} |\uparrow_1\rangle \otimes |\downarrow_2\rangle + a_{21} |\downarrow_1\rangle \otimes |\uparrow_2\rangle$$

to which we enact S_x^T, again in pieces, recalling that $S_x |\uparrow\rangle = \frac{\hbar}{2} |\downarrow\rangle$ and $S_x |\downarrow\rangle = \frac{\hbar}{2} |\uparrow\rangle$.

$$(S_{x_1} \otimes \mathcal{I}) |\psi\rangle = \left(S_x^1 \otimes \mathcal{I}\right) (a_{12} |\uparrow_1\rangle \otimes |\downarrow_2\rangle + a_{21} |\downarrow_1\rangle \otimes |\uparrow_2\rangle)$$

$$(S_{x_1} \otimes \mathcal{I}) |\psi\rangle = \frac{\hbar}{2} (a_{12} |\downarrow_1\rangle \otimes |\downarrow_2\rangle + a_{21} |\uparrow_1\rangle \otimes |\uparrow_2\rangle)$$

Similarly,

$$(\mathcal{I} \otimes S_{x_2}) |\psi\rangle = \frac{\hbar}{2} (a_{21} |\downarrow_1\rangle \otimes |\downarrow_2\rangle + a_{12} |\uparrow_1\rangle \otimes |\uparrow_2\rangle)$$

so together we have

$$S_x^T |\psi\rangle = \frac{\hbar}{2} (a_{12} + a_{21}) (|\uparrow_1\rangle \otimes |\uparrow_2\rangle + |\downarrow_1\rangle \otimes |\downarrow_2\rangle) .$$

For this state to be have a 0 total x-component for angular momentum we thus require $a_{12} = -a_{21}$. Remembering that our earlier condition so that the z was 0 was $a_{11} = 0$ and $a_{22} = 0$ the state having both components equal to 0 must look like

$$|\psi\rangle = \alpha (|\uparrow_1\rangle \otimes |\downarrow_2\rangle - |\downarrow_1\rangle \otimes |\uparrow_2\rangle)$$

which is not normalised so let's make it so!

$$\langle\psi|\psi\rangle = \alpha^* \alpha (\langle\uparrow_1| \otimes \langle\downarrow_2| - \langle\downarrow_1| \otimes \langle\uparrow_2|) (|\uparrow_1\rangle \otimes |\downarrow_2\rangle - |\downarrow_1\rangle \otimes |\uparrow_2\rangle)$$

$$\langle\psi|\psi\rangle = |\alpha|^2 (\langle\uparrow_1 | \uparrow_1\rangle \otimes \langle\downarrow_2 | \downarrow_2\rangle - \langle\uparrow_1 | \downarrow_1\rangle \otimes \langle\downarrow_2 | \uparrow_2\rangle - \langle\downarrow_1 | \uparrow_2\rangle \otimes \langle\uparrow_2 | \downarrow_2\rangle + \langle\downarrow_1 | \downarrow_1\rangle \otimes \langle\uparrow_2 | \uparrow_2\rangle)$$

$$\langle\psi|\psi\rangle = |\alpha|^2 (1 - 0 - 0 + 1) \implies 2|\alpha|^2 = 1 \implies \alpha = \frac{1}{\sqrt{2}}.$$

This state,

$$|\psi\rangle = \frac{1}{\sqrt{2}} (|\uparrow_1\rangle \otimes |\downarrow_2\rangle - |\downarrow_1\rangle \otimes |\uparrow_2\rangle)$$

which we have shown to have total angular momentum 0 in the x and z directions and accepting on good faith that it has 0 in y also, has no total angular momentum and shall be our tool for inquiry in the coming section.

1.2 Describing Entangled States

When thinking of system with more than one particle we saw that the way to describe this situation would be by taking the tensor product of the state spaces of these respective particles. So for 2 particles we would have some $V \otimes W$. We also saw that the the basis vectors of this new space will be the tensor

products of the basis vectors v_i and w_j of form $v_i \otimes w_j$. Indeed we qualified the fact that for two spin-$\frac{1}{2}$ systems, their tensor product state space will have in general states of the type

$$|\psi\rangle = a_{11}\,|\uparrow_1\rangle \otimes |\uparrow_2\rangle + a_{12}\,|\uparrow_1\rangle \otimes |\downarrow_2\rangle + a_{21}\,|\downarrow_1\rangle \otimes |\uparrow_2\rangle + a_{22}\,|\downarrow_1\rangle \otimes |\downarrow_2\rangle\,.$$

But, this doesn't mean that there can't be states $|\phi\rangle$ which can be described by one tensor product $v_k \otimes w_l$ $\exists\, v_k \in V$, $w_l \in W$ which would mean that the first particle is in state v_k and the second is in w_l. That is, we can describe the particles independently! We can thus say that the two particles occupying state $|\phi\rangle$ are not entangled. Of course if there are no such v_k and v_l describing the multipartite state in terms of a single tensor product vector then we say that the particles occupying this state are entangled!

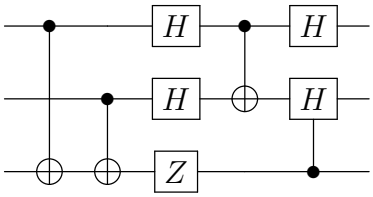
To clarify this idea, a non-entangled state is one where v_k is a linear combination

To be Continued

2 The Density Operator

3 Basic Quantum Computing and All That

Now that we've described entanglement and we want to do stuff with it, we need to learn about how to describe algorithms involving qubits. This is done using circuits. In circuits, lines represent qubits and gates transform these qubits.



3.1 Classical Computation

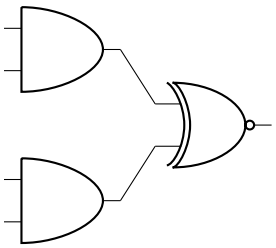
3.1.1 Gates & Circuits

In logic, by relating variables we construct boolean functions which will be true for some input and false for some other. We relate variables using logical operators such as the conjunctor $\wedge$, the adjuntor $\vee$, the negator $\neg$ and so on. This idea of a boolean function is what truly lies behind the idea of algorithms. To perform classical computations these logical operators are instead thought of as Gates, such as AND, NAND, XOR which are used to form logic circuits which represent algorithms such as what we have below, where the the out put of the AND gates is taken as the input of an XNOR.

Breaking this down logically, we would have four variable as input to a boolean function like so[1]

$$f(x_1, x_2, x_3, x_4) = (x_1 \wedge x_2) \oplus (x_3 \wedge x_4)\,.$$

[1] I am denoting the disjunctor or exclusive OR as $\oplus$.



That is, two relate four bits the following way one would throw them into such a circuit and end up with this relation. Algorithms are nothing more than variable relations and so can be represented as circuits and built from boolean functions.

Definition 3.1. Any function of form

$$f : \{0,1\}^n \to \{0,1\}$$

is known as a Boolean Function.

Binary In the world of classical computing we are tasked with representing things using bits, and as such we represent numbers in binary, which is just a fancy way of saying **base 2**. In general, we use base 10 to describe numbers such that each position represents an amount of powers of 10. For example

$$500 = 5 \times 10^2 + 0 \times 10^1 + 0 \times 10^0.$$

now in binary we instead want to work with powers of 2

$$500 = 2^8 + 2^7 + 2^6 + 2^5 + 2^4 + 0 \times 2^3 + 2^2 + 0 \times 2^1 + 0 \times 2^0 = 111110100.$$

So we would need 9 bits to represent 500.

The Half-adder Circuit To really drive this idea home let's see how we can add numbers together using nothing but boolean functions. To begin with we need to figure out how to do this for singular bits and then we can generalise this to n bits. The circuit below is called a half adder and is what we will use to add two bits together.

This takes two variables and relates them using an AND and an XOR gate.
As a boolean function we have two components

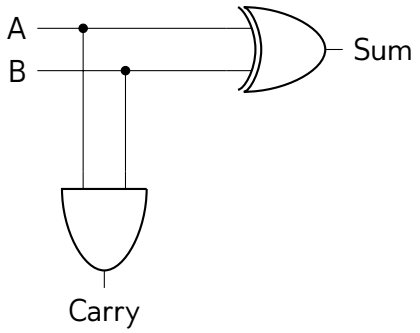
$$\text{Sum} = \text{A} \oplus \text{B} \qquad\qquad \text{Carry} = \text{A} \wedge \text{B}$$

giving a truth table as in Figure 1.

So now with this truth table in place, say we want to perform $1 + 1$, we let A$= 1$ & B$= 1$ giving $A \oplus B = 0$ and $A \wedge B = 1$. This means that their sum is a 0 in the position of the first bit but that we have to *carry* a 1 onto the position of the second bit. That is

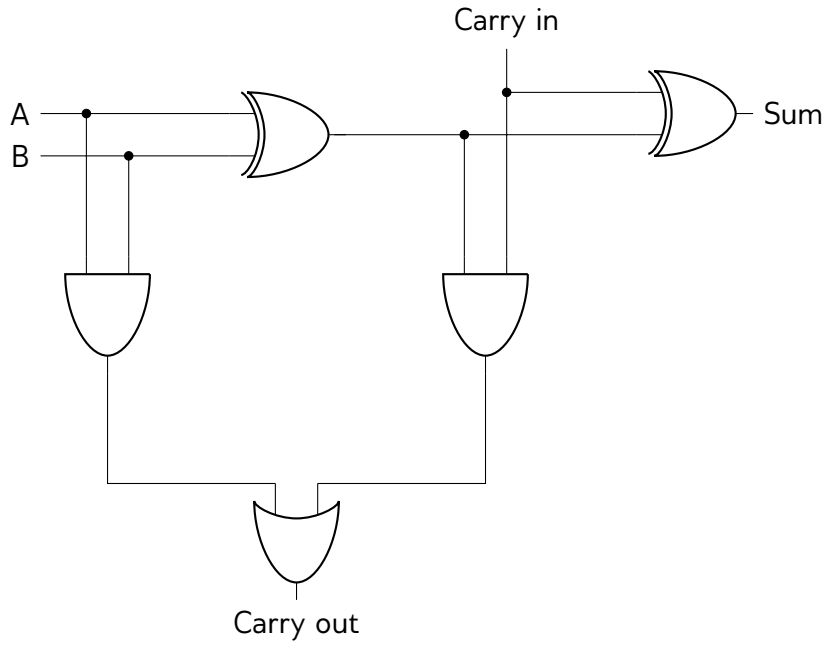
$$1 + 1 = 10 = 1 \times 2^1 + 0 \times 2^0 = 2$$

which is what we wanted.



A	B	$A \oplus B$	$A \wedge B$
1	1	0	1
1	0	1	0
0	1	1	0
0	0	0	0

Adding Numbers Great! Since n-bit integers are formed by single bits occupying places representing different powers of 2 then by using half-adders connected to each other sequentially we can add each power of 2 individually and then combine the result to form the number we require. This idea of combining half-adders results in the full-adder which is nothing more than two cascaded half-adders allowing us the ability to *Carry in* a number from a previous calculation beforehand and add it to the sum of our variables as well as collecting a *Carry Out* from these 3 contributors.



Let's try this out by performing $3 + 2 = 5$. In binary we have $3 = 11$ and $2 = 10$ so we begin by adding 1 and 0, which represent the values for 2^0 of each number, by chucking them into a Half-Adder. We have $1 \oplus 0 = 1$ and $1 \wedge 0 = 0$ so we will carry in 1×2^0 when adding 1×2^1 and 1×2^1 which in this case will be thought of as 1 and 1. We first perform $1 \oplus 1$ which gives 0 so our new number will have 0×2^1 and move on to carrying. Here we now have $(1 \oplus 1) \wedge 1 = 1$ which means we carry a 1 over onto the 2^2 position. All in all we have

$$1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 101 = 5 = 3 + 5$$

and we know how computers count.

Universality of NOR and NAND If you've done some introductory logic you have come across the ideas of De Morgan's Laws of equivalence. These allow us to rewrite logical statements in terms of other gates. For example the statement

$$a \wedge b \equiv \neg a \vee \neg b$$

where the expression on the right is how the NOR gate is defined in terms of $\{\neg, \wedge\}$ (a complete set also). In light of this we may introduce the idea of functional completeness or universality in that any boolean function may be stated in terms of the singletons $\{\downarrow\}$ and $\{|\}$, making use of the notation employed in [6], representing the negated inclusive adjunctor (NOR) and the negated conjunctor (NAND), shown below. These

ideas are explored much more rigorously and proven in [6] around page 50. A fun fact which drives this home is that the Apollo Guidance Computer which took us to the moon was produced out of 5600 NOR gates.

So what about addition on a Quantum Computer? You might be tempted to think that all of the above can just be done **as is** using qubits but whilst we can perform addition using qubits the process is quite different because the physical laws which govern our computations are now different.
To begin with even something as simple as passing information down a channel, a wire, is now difficult because your information may decohere or god forbid something measures it. Another key issue, in particular in Gate design, is that of Unitarity. In Quantum Mechanics all processes are required to be carried out by Hamiltonians which are Unitary or more simply reversible. Gates such as the AND and XOR gates are not invertible and so how we carry these processes out on a quantum computer shall be fundamentally different. Something like FANOUT, which would create copies of bits on classical computers is now impossible on a quantum computer because of the no-cloning theorem.

As a spoiler, you can do everything you can do with a classical computer on quantum computer, albeit in a more complicated way. Indeed, the Toffoli gate, a unitary gate which we will learn about soon enough, is equivalent to a NAND gate and so can produce any classical computation. Together with another gate, the Hadamard, these two gates form a universal set for quantum computation and so in some way, all of computation (see figure 1) ...

3.2 Down & Dirty with Quantum Computation

I'm going to assume the reader knows that bits look like the things on the left and qubits look like the thing on the right and operate under the assumption that we know how quantum objects behave or rather what humans have figured out about how they behave.

The main emphasis of what is to come is going to be a fairly pragmatic introduction into what quantum gates are, where they come from and what we can do with them.

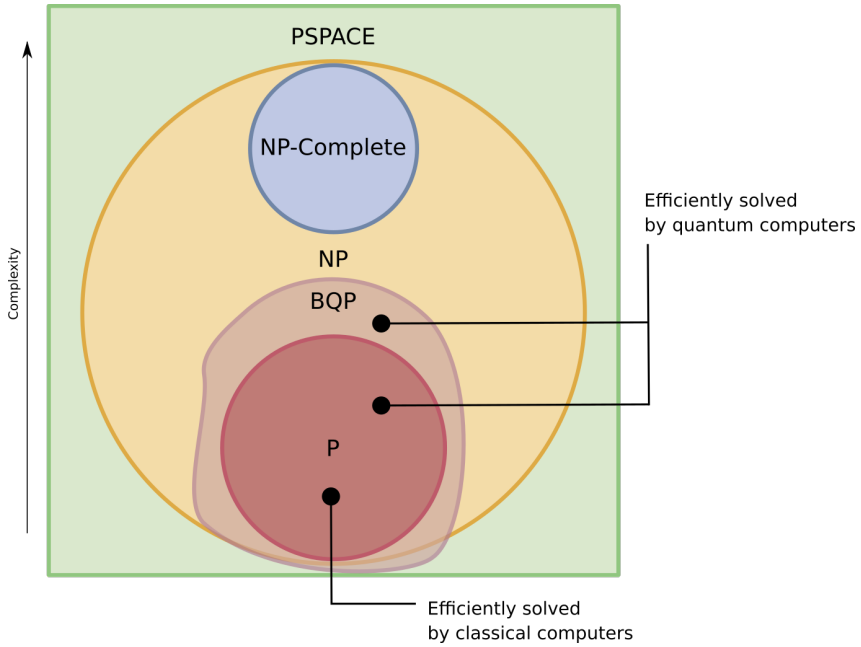


Figure 1: A Complexity Theoretic Digression, image by Dušan Petričić

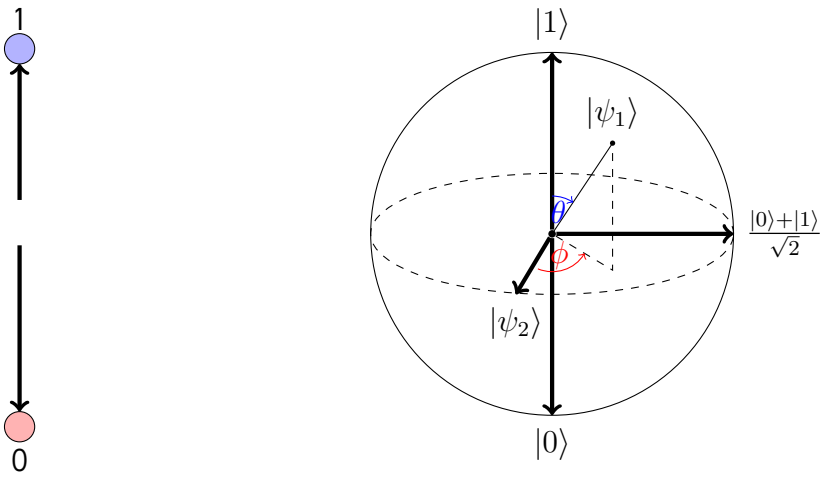
3.2.1 Reversible Computation

In classical computation, the physical processes which encode the logic we propose do not need to actually embody the logic which takes place. Think of it this way, if I start with two bits and take them through an and gate I have not in effect lost a bit because I end up with one bit of information as a result, but from a purely logical point of view I have, from a physical view I haven't. This is one of the main differences between classical and quantum computations, in quantum computing the qubits have to actually live out the computation, the have to entangle they have to be in a superposition, they have to experience a Hamiltonian and Hamiltonians are **Unitary**, reversible processes.
Indeed, any quantum circuit can be expressed as a Hamiltonian and particularly any gate should be thought of as a physical Hamiltonian.

Classical computational processes are carried out by permutation matrices which are a subgroup of the unitary matrices. This again alludes to P $\subset$ BQP although Scott Aaronson has been examining even more general computational models since 2004 based on linear operators more generally.

To achieve physical reversibility we need also logical reversibility which is given by gates which have the same number of input and output bits.
Definition 3.2. A Boolean gate G is said to be *reversible* if it has the same number of inputs as outputs, and its mapping from input strings is a bijection.

Figure 2: Difference between a bit and a qubit



X - A Quantum NOT Gate A classical NOT gate takes a $0 \to 1$ and a $1 \to 0$ but since a general quantum state $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \;:\; \alpha, \beta \in \mathbb{C}$ then we want a gate capable of

$$\alpha\,|0\rangle + \beta\,|1\rangle \to \beta\,|0\rangle + \alpha\,|1\rangle\,.$$

As one can see we are swapping the coefficients for the basis vectors, effectively flipping the state. Now, earlier we introduced the S_x operator which behaved like

$$S_x\,|\uparrow\rangle = \frac{\hbar}{2}\,|\downarrow\rangle \text{ and } S_x\,|\downarrow\rangle = \frac{\hbar}{2}\,|\uparrow\rangle$$

which is clearly *flippy* in nature as we wish. Now S_x from this relation can be found to have the form

$$S_x = \frac{\hbar}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

and defining

$$\sigma_x := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

which is the dimensionless version, also known as the Pauli X Operator. To our $|\phi\rangle$ this now carries out the desired transformation in the computational basis

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}.$$

In circuit form we write this as follows; $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ —[x]— $\begin{pmatrix} \beta \\ \alpha \end{pmatrix}$

Now the X Gate is clearly unitary since it's its own hermitian adjoint and multiplying it by itself gives I but also on a conceptual level it accepts one qubit as an input and outputs one qubit also.

To be Continued

References

[1] Nielsen, M., & Chuang, I. (2010). Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511976667

[2] Smith, G. & Yard, J. (2008). *Quantum Communication with Zero-Capacity Channels*, Science.

[3] MIT 6.004 - Computation Structures- Lecture 4

[4] Zwiebach, B. (2013) Quantum Mechanics II Course Notes. Chapter 8. *Multiparticle States and Tensor Products*

[5] Preskill, J. (2015) Lecture Notes for Ph219/CS219: *Quantum Information*.

[6] Enderton, H. (2001). A Mathematical Introduction to Logic. Academic Press.

[7] O' Donnell, R. (2015) Quantum Computation and Information Course Notes.

[8] Ekert, A, Hayden, P. & Hitoshi, I. (2008). Basic concepts in quantum computation.