A Quantum Abacus for Teaching Quantum Algorithms

Dan-Adrian German
Luddy School of Informatics
Computing and Engineering
Indiana University
Bloomington, Indiana, USA
Email: dgerman@indiana.edu

Marcelo Pias
Federal University of Rio Grande
Computer Science Centre (C3)
Rio Grande do Sul, Brazil

Qiao Xiang
Xiamen University
School of Informatics
Xiamen, Fujian, China

Sreesha Srinivasan Kuruvadi
Luddy School of Informatics
Computing and Engineering
Indiana University
Bloomington, Indiana, USA

***Abstract*—At the time of this writing more than 60 (sixty) companies in the world are building quantum computers. These computers, based on quantum physics principles, are radically different from those that operate according to the more familiar principles of classical physics. A quantum algorithm takes a number of classical bits as its input, manipulates them so as to create a superposition of all their possible states, further manipulates this exponentially large superposition to obtain the final quantum result, and then measures the result to get (with the appropriate probability distribution) the same number of output bits as in its input. For the middle phase, there are elementary operations which count as one step and yet manipulate all the exponentially many amplitudes of the superposition. The natural language of these quantum gates is that of linear algebra in a complex (Hilbert) vector space. Since 2017 it is known that it is possible to replace the linear algebra with some string-rewriting rules which are no more complicated than the basic rules of arithmetic. The original system was introduced by Terry Rudolph and has been promoted and disseminated in large-scale outreach projects (among others) by Diana Franklin (University of Chicago) and Sofia Economou and Ed Barnes (Virginia Tech) as well as several other educators at the high-school level. In this paper we show how a slightly modified (though still very elementary) system can be used to communicate a visual and entirely operational understanding of key quantum computation concepts such as: superposition, probability, entanglement, phase, interference and unitary state evolution, as they occur in well-known quantum algorithms. We give concrete examples of proving properties for quantum gates and quantum circuits without resorting at all to complex numbers or matrix multiplication. Only simple, abacus-like operations are used—hence the title of the paper. The system we present allows a novice learner to actually trace a quantum algorithm as if it were a classical computation, which is a rare (and, frankly, borderline incredible) luxury in the area of quantum computation, where traditional debugging is impossible. Examples include the phase kickback phenomenon and the famous Deutsch-Josza algorithm. We end with a discussion (and more examples) of how this approach can create a genuine bridge to the mathematics of quantum computation, that is, of vector and tensor algebras in complex spaces for students who may have little or no proper mathematical background.**

I. Introduction

A qubit is of the form $\alpha|0\rangle + \beta|1\rangle$ with $\alpha, \beta \in \mathbb{C}$ and such that $\alpha^2 + \beta^2 = 1$. For a certain subset of qubits we are going to introduce a graphical representation: start by defining $|0\rangle = \bigcirc$ and $|1\rangle = \bullet$. Then we represent $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ as $\{\bigcirc, \bullet\}$ and we think of it as a set with two observable outcomes, each one having the probability $\frac{1}{2}$. Similarly $\frac{1}{\sqrt{3}}|0\rangle + \sqrt{\frac{2}{3}}|1\rangle$ will be represented as $\{\bigcirc, \bullet, \bullet\}$. It should be obvious that only qubits with $\alpha^2, \beta^2 \in \mathbb{Q}^+$ admit this representation. Furthermore we could write $|0\rangle = \{\bigcirc\}$ but in such a case we choose to drop the pedantry and the braces. So we have:

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix} = \bigcirc \quad \text{and} \quad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix} = \bullet$$

A. One Qubit Gates, Phase and Superposition

Next we introduce the $\mathtt{X}$ gate defined by: $\mathtt{X}(\bigcirc) = \bullet$ and $\mathtt{X}(\bullet) = \bigcirc$. The reader can easily verify that this definition is a special case of the more traditional:

$$\mathtt{X}(\alpha|0\rangle + \beta|1\rangle) = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}\begin{pmatrix}\alpha\\\beta\end{pmatrix} = \begin{pmatrix}\beta\\\alpha\end{pmatrix}$$

The $\mathtt{X}$ gate is also known as the quantum $\mathtt{NOT}$ gate. It is a one-qubit gate. It is easy to check that $\mathtt{X}(\mathtt{X}(\bigcirc)) = \bigcirc$ and $\mathtt{X}(\mathtt{X}(\bullet)) = \bullet$ in other words $\mathtt{X}^2 = \mathbb{I}_2$ (the unit matrix of size 2). Here's another one-qubit gate, the Hadamard gate:

$$\mathtt{H} = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}$$

The Hadamard gate acts as follows:

$$\mathtt{H}|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = |+\rangle$$

We write this as follows in our notation:

$$\mathtt{H}(\bigcirc) = \Big\{\bigcirc, \bullet\Big\}$$

For the other input the behavior is:

$$\mathtt{H}|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |-\rangle$$

In our notation this becomes:

$$\mathtt{H}(\bullet) = \Big\{\bigcirc, \overline{\bullet}\Big\}$$

The negative sign placed on top of the qubit is called *phase*; we refer to a pair of curly braces as the *superposition* operator. If no intermediate measurements are involved the Hadamard gate shares this idempotence property with the $\mathtt{NOT}$ gate:

$$\mathtt{H}(\mathtt{H}(\bigcirc)) = \bigcirc \quad \text{and} \quad \mathtt{H}(\mathtt{H}(\bullet)) = \bullet$$

In fact $\mathtt{H}^2 = \mathbb{I}_2$ (these are unitary matrices) a property that can be easily checked via matrix multiplication. Now we establish rules of engagement for the phase and superposition operators.

Both operators are linear. For superposition we have:

$$\begin{aligned} \mathtt{H}(\mathtt{H}(\bigcirc)) =& \mathtt{H}(\Big\{\bigcirc, \bullet\Big\}) = \\ =& \Big\{\mathtt{H}(\bigcirc), \mathtt{H}(\bullet)\Big\} = \\ =& \Big\{\Big\{\bigcirc, \bullet\Big\}, \Big\{\bigcirc, \overline{\bullet}\Big\}\Big\} \end{aligned}$$

The operator represents sets of outcomes so:

$$\Big\{\Big\{\bigcirc, \bullet\Big\}, \Big\{\bigcirc, \overline{\bullet}\Big\}\Big\} = \Big\{\bigcirc, \bullet, \bigcirc, \overline{\bullet}\Big\}$$

Now $\Big\{\bigcirc, \bigcirc\Big\} = \Big\{\bigcirc\Big\} = \bigcirc$ follows from definition and is known as *constructive* interference. In this context the phase operator is responsible for a *destructive* type of interference:

$$\Big\{\bullet, \overline{\bullet}\Big\} = \emptyset \quad \text{so} \quad \Big\{\bigcirc, \bullet, \bigcirc, \overline{\bullet}\Big\} = \bigcirc$$

The phase operator acts in the same way the unary minus operator acts for multiplication[1]. Thus we have:

$$\begin{aligned} \mathtt{H}(\mathtt{H}(\bullet)) =& \mathtt{H}(\Big\{\bigcirc, \overline{\bullet}\Big\}) = \\ =& \Big\{\mathtt{H}(\bigcirc), \mathtt{H}(\overline{\bullet})\Big\} = \\ =& \Big\{\Big\{\bigcirc, \bullet\Big\}, \overline{\mathtt{H}(\bullet)}\Big\} = \\ =& \Big\{\Big\{\bigcirc, \bullet\Big\}, \overline{\Big\{\bigcirc, \overline{\bullet}\Big\}}\Big\} = \bullet \end{aligned}$$

The last step follows from $\overline{\Big\{\bigcirc, \overline{\bullet}\Big\}} = \Big\{\overline{\bigcirc}, \overline{\overline{\bullet}}\Big\} = \Big\{\overline{\bigcirc}, \bullet\Big\}$.

B. Two Qubit Gates and Entanglement

Now to study entanglement we need two-qubit gates. With two qubits the order matters:

$$|01\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \bigcirc\bullet \neq \bullet\bigcirc$$

It's customary to introduce the SWAP gate next:

$$\mathtt{SWAP}(|10\rangle) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = |01\rangle$$

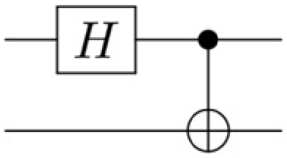
In our notation we have: $\mathtt{SWAP}(\bullet\bigcirc) = \bigcirc\bullet$

We now introduce the C-NOT gate, through an example:

$$\mathtt{C\text{-}NOT}(|10\rangle) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |11\rangle$$

In our notation that becomes: $\mathtt{C\text{-}NOT}(\bullet\bigcirc) = \bullet\bullet$

C-NOT distinguishes between its two arguments as follows: it first requires the control qubit, and then the target qubit. Let's look at a simple circuit now:



This is the circuit that creates the Bell states, as a function of its two inputs:

$$\mathtt{C\text{-}NOT}(\mathtt{H}(\bigcirc)\bigcirc) = |\Phi^+\rangle = \Big\{\bigcirc\bigcirc, \bullet\bullet\Big\}$$

Let's work this out:

$$\begin{aligned} \mathtt{C\text{-}NOT}(\mathtt{H}(\bigcirc)\bigcirc) =& \mathtt{C\text{-}NOT}(\Big\{\bigcirc, \bullet\Big\}\bigcirc) = \\ =& \mathtt{C\text{-}NOT}(\Big\{\bigcirc\bigcirc, \bullet\bigcirc\Big\}) = \\ =& \Big\{\mathtt{C\text{-}NOT}(\bigcirc\bigcirc), \mathtt{C\text{-}NOT}(\bullet\bigcirc)\Big\} = \\ =& \Big\{\bigcirc\bigcirc, \bullet\bullet\Big\} \end{aligned}$$

The reader should notice that the superposition operator is distributive (like in elementary algebra) with respect to (or within) the product of quantum states. Furthermore, as we said before, the product is not commutative.

Now, as an exercise, let's try to answer each of the following two questions: (a) Is this quantum state an entangled[2] state?

$$\Psi = \sqrt{\frac{1}{4}} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Answer: no, because

$$\Big\{\bigcirc\bigcirc, \bigcirc\bullet, \bullet\bigcirc, \bullet\bullet\Big\} = \Big\{\bigcirc, \bullet\Big\}\Big\{\bigcirc, \bullet\Big\}$$

(b) Is this quantum state an entangled state?

$$\Psi = \frac{|00\rangle + |01\rangle}{\sqrt{2}}$$

Answer: no, because

$$\Big\{\bigcirc\bigcirc, \bigcirc\bullet\Big\} = \Big\{\bigcirc\Big\}\Big\{\bigcirc, \bullet\Big\} = \bigcirc\Big\{\bigcirc, \bullet\Big\}$$

In both cases we invoke distributivity.

C. String Rewriting vs. Graph Rewriting

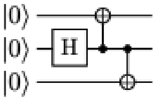
It should be clear that the system described in this paper is in fact a string-rewriting system [6]. Rewriting systems are at the foundation of Computer Science, they are, in fact, the very fabric of it (e.g., Turing machines and lambda calculus) so this is a very fortunate development. In recent years another rewriting system, the ZX-calculus ([1], [4]) has

[1] The product of two negatives is a positive because the inverse of the inverse of a positive number is that positive number back again. When applied to a set, though, the phase operator changes the sign (phase) for each outcome.

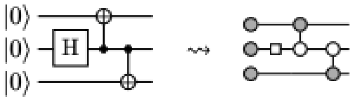
[2] As a reminder, an entangled state of a composite system is a state that cannot be written as a product state of the component systems.

gained in popularity and is routinely being used in industry for simplification of quantum circuits. Let's briefly compare and relate the two systems to each other before going any further.

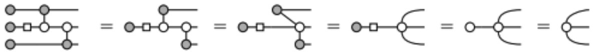
The system we're using here does not have an established name[3] (we refer to it as the "Quantum Abacus") and uses string rewriting rules to show what happens with the quantum state as it travels through a circuit. By contrast the ZX-calculus is a diagrammatic language that rewrites entire portions of the circuit (so it's a graph-rewriting technique) while preserving the equivalence of the circuit. One is a global technique; the other helps trace a quantum state through a circuit in "slow-motion". As an example consider the following circuit:



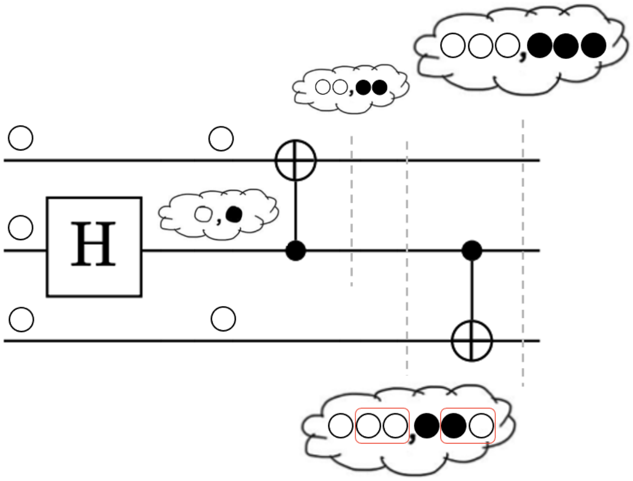
It can be written as a ZX-diagram (see [4]):



Which can then be simplified as follows:



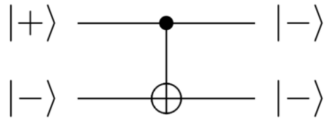
This proves (diagrammatically) that the circuit implements a GHZ state[4]. By comparison the same proof with the "Quantum Abacus" proceeds as follows:



Here superpositions are represented as "misty states" which is just a graphically somewhat richer representation of our superposition (set) operator .

II. Phase KickBack

Phase kickback is not an algorithm, but a technique (a useful concept, or trick) in quantum algorithm design. It provides a framework to understand many famous quantum algorithms, such as Shor's algorithm, the phase estimation algorithm, the Deutsch algorithm, Simon's algorithm, etc. The essence of it can be captured in this diagram:



The behavior of the `C-NOT` gate in this diagram is (at first) a bit counterintuitive: the control qubit changes while the target stays the same. By simple matrix multiplication one can verify the truth of this diagram. Let's prove it (with the abacus):

$$\texttt{C-NOT}(|+-\rangle) = |--\rangle$$

We start by reminding ourselves that

$$|+\rangle = \{\bigcirc, \bullet\} \quad \text{and} \quad |-\rangle = \{\bigcirc, \overline{\bullet}\}$$

We also decided to use $\Rsh$ instead of `C-NOT` to save space (the arrow reminds us that the control qubit is the first in the pair).

So now we calculate:

$$\begin{aligned}
\Rsh(|+-\rangle) &= \Rsh(\{\bigcirc,\bullet\}\{\bigcirc,\overline{\bullet}\}) \\
&= \Rsh(\{\bigcirc\bigcirc,\bigcirc\overline{\bullet},\bullet\bigcirc,\bullet\overline{\bullet}\}) \\
&= \{\Rsh(\bigcirc\bigcirc),\Rsh(\bigcirc\overline{\bullet}),\Rsh(\bullet\bigcirc),\Rsh(\bullet\overline{\bullet})\} \\
&= \{\bigcirc\bigcirc,\bigcirc\overline{\bullet},\bullet\bullet,\bullet\overline{\bigcirc}\} \\
&= \{\bigcirc\bigcirc,\bigcirc\overline{\bullet},\overline{\bullet}\overline{\bullet},\overline{\bullet}\bigcirc\} \\
&= \{\bigcirc\bigcirc,\bigcirc\overline{\bullet},\overline{\bullet}\bigcirc,\overline{\bullet}\overline{\bullet}\} \\
&= \{\bigcirc\{\bigcirc,\overline{\bullet}\},\overline{\bullet}\{\bigcirc,\overline{\bullet}\}\} \\
&= \{\bigcirc,\overline{\bullet}\}\{\bigcirc,\overline{\bullet}\} \\
&= |--\rangle
\end{aligned}$$

This concludes the proof.

III. Teleportation

Initially introduced in Bennett et al. (1993), quantum teleportation describes a protocol allowing to reconstruct an unknown quantum state $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$ at a new location by using a classical information channel and a pair of entangled states. So the first challenge is going to be to find a way to represent an arbitrary $|\Psi\rangle$ state in our "abacus" system. As we shall see this is not going to be very hard. Furthermore it will allow us to morph gradually into the traditional, mathematical representation. Let's start by assuming some simple, manageable form for the given quantum state, for

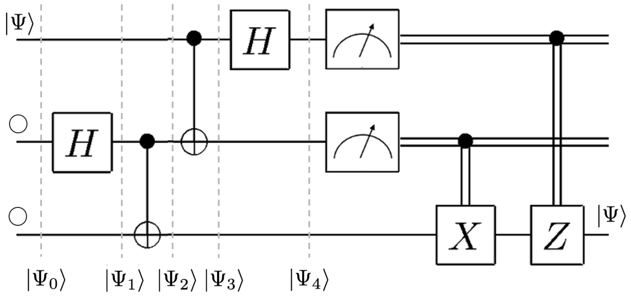
[3] Other than by reference to its inventor: Terry Rudolph's system.

[4] The Greenberger–Horne–Zeilinger (GHZ) state is an entangled quantum state for 3 qubits with this expression: $\frac{|000\rangle+|111\rangle}{\sqrt{2}}$ (so, $\{\bigcirc\bigcirc\bigcirc, \bullet\bullet\bullet\}$).

example let's assume $|\Psi\rangle = \frac{1}{\sqrt{3}}|0\rangle + \sqrt{\frac{2}{3}}|1\rangle$. Then we said we represent $|\Psi\rangle$ in our system via probabilities of measured outcomes; so, in this case $|\Psi\rangle = \{\text{○}, \text{●}, \text{●}\}$. It should be clear that $|\Psi\rangle = \{\text{○}, 2\,\text{●}\}$ conveys the same information via the new coefficient (i.e., 2) and the representation is a bit more compact, as well. In that case $\forall \alpha, \beta \in \mathbb{C}$ we have:

$$\alpha|0\rangle + \beta|1\rangle = \{\alpha\,\text{○}, \beta\,\text{●}\}$$

Quantum teleportation ([7], [9]) requires three qubits, where the first one holds the state to be teleported and the remaining ones are initialised to $|0\rangle$. The protocol consists of performing the following quantum circuit:



The word teleportation does fit well here as this phenomenon occurs instantaneously and is not affected by distance or separating barriers. The instantaneously teleported state cannot be used to achieve faster than light communication, as in order to be properly reconstructed requires classical information about measurement performed at the sender location, making it sensitive to limitations imposed by the speed of light. Let's prove the protocol by calculating the intermediary stages $|\Psi_0\rangle, \ldots, |\Psi_4\rangle$. We start with:

$$|\Psi_0\rangle = \Big\{\alpha\,\text{○○○}, \beta\,\text{●○○}\Big\}$$

Traditional calculation confirms this:

$$|\Psi_0\rangle = (\alpha|0\rangle + \beta|1\rangle) \otimes |00\rangle = \alpha|000\rangle + \beta|100\rangle$$

In the classroom this would be a good moment to talk about tensor products and relate the following:

$$\text{●○●} \equiv |1\rangle \otimes |0\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Please don't forget that $|0\rangle$ and $|1\rangle$ are in fact vectors. We will revisit this topic briefly at the end of this section. Furthermore we can continue to calculate and relate the results obtained via the "abacus" to those obtained via standard mathematical operations. As an example we can calculate:

$$\begin{aligned} \mathtt{H}(\{\alpha\,\text{○}, \beta\,\text{●}\}) =& \alpha\,\{\text{○}, \text{●}\} + \beta\,\{\text{○}, \overline{\text{●}}\} = \\ =& \{(\alpha+\beta)\text{○}, (\alpha-\beta)\text{●}\} \end{aligned}$$

This is clearly confirmed by the standard calculation:

$$\mathtt{H}(\alpha|0\rangle + \beta|1\rangle) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} \alpha+\beta \\ \alpha-\beta \end{pmatrix}$$

So now we can calculate:

$$|\Psi_1\rangle = \Big\{\alpha\,\text{○○○}, \alpha\,\text{○●○}, \beta\,\text{●○○}, \beta\,\text{●●○}\Big\}$$

Traditional calculation, again, confirms our result:

$$|\Psi_1\rangle = \frac{1}{\sqrt{2}}\Big(\alpha|000\rangle + \alpha|010\rangle + \beta|100\rangle + \beta|110\rangle\Big)$$

After the first `C-NOT` gate:

$$|\Psi_2\rangle = \Big\{\alpha\,\text{○○○}, \alpha\,\text{○●●}, \beta\,\text{●○○}, \beta\,\text{●●●}\Big\}$$

Traditional calculation yields:

$$|\Psi_2\rangle = \frac{1}{\sqrt{2}}\Big(\alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle\Big)$$

The second `C-NOT` acts on the first two qubits:

$$|\Psi_3\rangle = \Big\{\alpha\,\text{○○○}, \alpha\,\text{○●●}, \beta\,\text{●●○}, \beta\,\text{●○●}\Big\}$$

Using standard calculation techniques:

$$|\Psi_3\rangle = \frac{1}{\sqrt{2}}\Big(\alpha|000\rangle + \alpha|011\rangle + \beta|110\rangle + \beta|101\rangle\Big)$$

We now have only one stage left but it should be relatively clear that developments are now in lockstep. So, after the second Hadamard gate (acting on just the first qubit):

$$\begin{aligned} |\Psi_4\rangle = \Big\{&\alpha\{\text{○}, \text{●}\}\text{○○}, \alpha\{\text{○}, \text{●}\}\text{●●}, \\ &\beta\{\text{○}, \overline{\text{●}}\}\text{●○}, \beta\{\text{○}, \overline{\text{●}}\}\text{○●}\Big\} = \\ = \Big\{&\alpha\,\text{○○○}, \alpha\,\text{●○○}, \alpha\,\text{○●●}, \alpha\,\text{●●●}, \\ &\beta\,\text{○●○}, \beta\,\overline{\text{●}}\text{●○}, \beta\,\text{○○●}, \beta\,\overline{\text{●}}\text{○●}\Big\} \end{aligned}$$

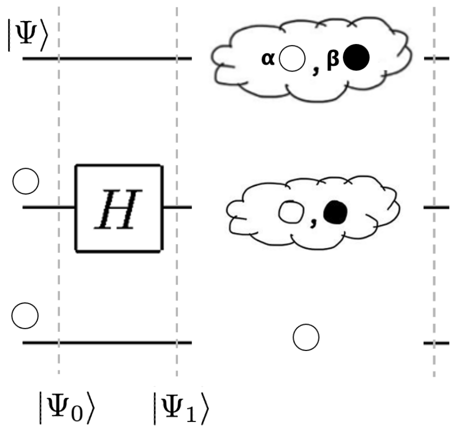
Traditional calculation meanwhile yields (same thing):

$$\begin{aligned} |\Psi_4\rangle = & \frac{1}{2}|00\rangle\Big(\alpha|0\rangle + \beta|1\rangle\Big) + \quad \mathtt{;\ nothing} \\ & + \frac{1}{2}|01\rangle\Big(\beta|0\rangle + \alpha|1\rangle\Big) + \quad \mathtt{;\ apply\ X} \\ & + \frac{1}{2}|10\rangle\Big(\alpha|0\rangle - \beta|1\rangle\Big) + \quad \mathtt{;\ apply\ Z} \\ & + \frac{1}{2}|11\rangle\Big(-\beta|0\rangle + \alpha|1\rangle\Big) \quad \mathtt{;\ X,\ then\ Z} \end{aligned}$$

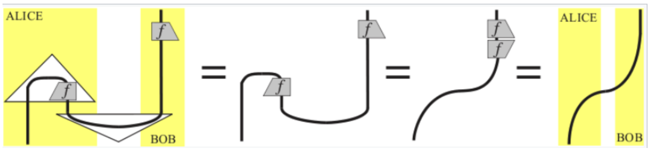
In this form it is visible what gates have to be applied to the last qubit to make it the input teleported state $\alpha|0\rangle + \beta|1\rangle$. The gates to be applied depend on the measurement of the first two qubits, as teleported state is still entangled with them. That is the motivation behind the idea of classical correction, which is the last stage in this protocol (and indicated via annotations in this last equation). Let's now revisit, as we promised, the topic of tensorial product in the context of our derivation:

$$\begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \otimes \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} \otimes \begin{pmatrix} z_0 \\ z_1 \end{pmatrix} = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \otimes \begin{pmatrix} y_0 \begin{pmatrix} z_0 \\ z_1 \end{pmatrix} \\ y_1 \begin{pmatrix} z_0 \\ z_1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \otimes \begin{pmatrix} y_0 z_0 \\ y_0 z_1 \\ y_1 z_0 \\ y_1 z_1 \end{pmatrix}$$

This is exactly what is happening in our “abacus” calculations, for example in the first stage, as we determine $|\Psi_1\rangle$:



Two final comments in this section. First, that a(nother) diagrammatical proof of teleportation would look like this:



This is Penrose notation [5] and the approach is similar to what we saw when we mentioned the ZX-calculus. Note also that there is no transfer of matter or energy involved. No particle has been physically moved (from Alice to Bob); only its state has been transferred. The term “teleportation”, coined by Bennett, Brassard, Crépeau, Jozsa, Peres and Wootters, reflects the indistinguishability of quantum mechanical particles.

IV. The Deutsch Algorithm

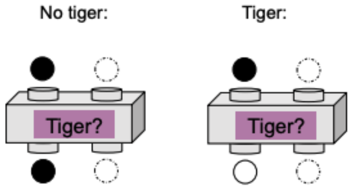
The Deutsch-Jozsa algorithm[5] was the first to show a separation between the quantum and classical difficulty of a problem. This algorithm demonstrates the significance of allowing quantum amplitudes to take both positive and negative values, as opposed to classical probabilities that are always non-negative.

We examine a variant of this algorithm designed as a game called ([3], [2]) “Money or Tiger”. As the authors explain ”[the game] does not require more than one student and relies on only pen and paper and the [“Quantum Abacus”] formalism[; i]t can [thus] be viewed as a preparatory step toward a proper linear-algebra treatment. [... I]t introduces the concept of a quantum algorithm and the advantages that [Quantum Mechanics] can bring to information processing. [...] It shows that a simple algorithm (combination of boxes) employing quantum gates can be used to solve a problem [faster than] what can be done using only classical information processing.” We emphasize that the quantum algorithm is twice as fast than the fastest possible classical solution.

The setup of the game is as follows: there are two doors, one labeled with a white circle, the other one with a black circle. There is a button on the wall that opens both doors simultaneously. It is not possible to open only one door. There is money behind at least one door. There may or may not be a tiger behind one of the doors. If there is no tiger, then you want to push the button and collect the money. However, if there is a tiger, then you do not want to push the button, and instead you leave without the money, happy enough that you are still alive. Also on the wall is a box labeled “Tiger?”.

You are allowed to query this box once (and only once) to check whether there is a tiger. The way the box works is as follows: the box has two input ports and two output ports. You always input a black marble in the left input, and in the right you insert a marble whose color matches the door you want to check. If you want to know whether there is a tiger behind the white door, then you insert a white marble, while to check if there’s a tiger behind the black door, you insert a black marble in the right input port. The door marble comes out the same color regardless of whether or not there is a tiger. However, the test marble changes color if a tiger is present.

These rules are summarized below:

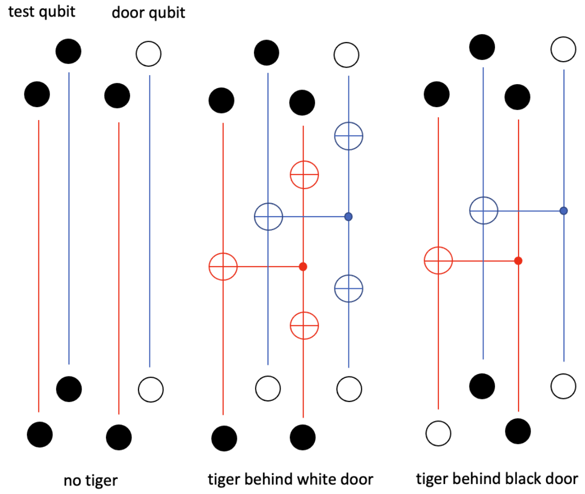


The “Tiger?” box is also called an *oracle*. If we only have access to classical information processing, then it is clear that the “Tiger?” box needs to be queried twice in order to be sure there is no tiger present. You would have to use it once for each one of the two doors. The point of this game is to show that Quantum Mechanics allows us to determine whether or not there is a tiger behind either one of the doors with absolute certainty *while only using the oracle box once*.

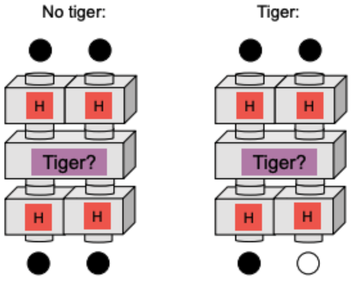
There are three cases to consider: (a) no tiger, (b) tiger behind white door and (c) tiger behind black door. We will design an oracle (and a quantum circuit) for each one and

[5] The Deutsch–Jozsa algorithm is a deterministic quantum algorithm proposed by David Deutsch and Richard Jozsa in 1992 with improvements by Richard Cleve, Artur Ekert, Chiara Macchiavello, and Michele Mosca in 1998. Although of little current practical use, it is one of the first examples of a quantum algorithm that is exponentially faster than any possible deterministic classical algorithm. The Deutsch–Jozsa problem is specifically designed to be easy for a quantum algorithm and hard for any deterministic classical algorithm. It is a black box problem that can be solved efficiently by a quantum computer with no error, whereas a deterministic classical computer would need an exponential number of queries to the black box to solve the problem. More formally, it yields an oracle relative to which EQP, the class of problems that can be solved exactly in polynomial time on a quantum computer, and P are different. Since the problem is easy to solve on a probabilistic classical computer, it does not yield an oracle separation with BPP, the class of problems that can be solved with bounded error in polynomial time on a probabilistic classical computer. Simon’s problem is an example of a problem that yields an oracle separation between BQP and BPP.

prove our claim using the "Quantum Abacus". Here are the three oracles and how they function:



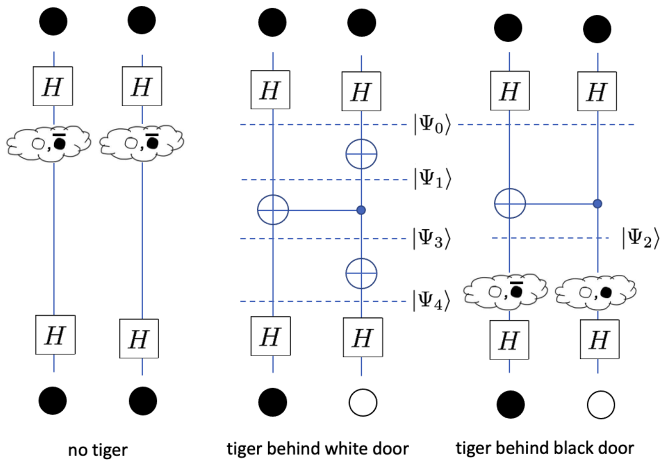
By adding additional gates above and below the oracle, it is possible to determine whether or not a tiger is present in one shot. This is shown below: if two black marbles are input into the circuit, then a white output signifies the presence of a tiger, regardless of which door the tiger is behind.



Unlike the classical case, where the oracle needs to be used twice, in the quantum case a single use of the tiger box suffices to identify the presence of a tiger. Note that if the box is used twice in the classical setting, we also find out which door the tiger is behind. In the quantum case, where the tiger box is only used once, we only find whether there is a tiger, but not which door it is behind. This is analogous to the Deutsch algorithm, where we find out using the quantum circuit whether a function is balanced or constant, but not which particular function it is. Barnes and Economou also note that "[as] the quantum case [...] only require[s] a single use of a box when the classical case requires a large number of uses [...] helps a student appreciate that the distinction between quantum and classical computing is about the number of algorithmic steps, and not about smaller and faster hardware or other similar misconceptions."

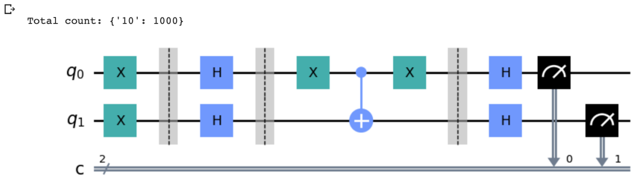
A. How the Deutsch Algorithm Works

For the non-classical case, involving quantum gates, we need to add Hadamard gates (as shown) before and after the oracle, in each of the three cases. Then, by changing the question we ask (we no longer have a test qubit and a door qubit, we now just drop two $|1\rangle$ qubits) we will be able to get the desired piece of information (is there a tiger behind the doors or not) in one shot. We now describe how that works.



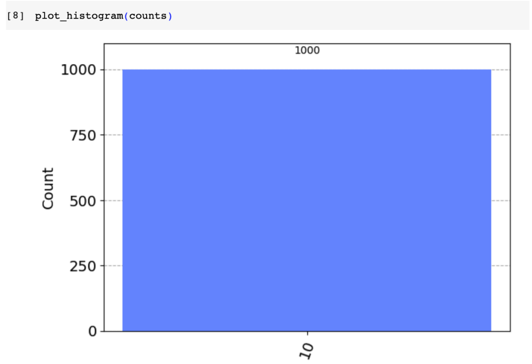
The circuit on the left is immediate: H(H(●)) = ● as we showed from the first section (page) or because H is unitary, so if there is no tiger the input ($|1\rangle|1\rangle$) is obtained unchanged in the output. If the tiger is behind a closed door we'll calculate shortly what happens via $|\Psi_0\rangle$, $|\Psi_1\rangle$, $|\Psi_2\rangle$, $|\Psi_3\rangle$ and $|\Psi_4\rangle$.

For when the tiger is behind the white door, as an additional means of checking that what we do here makes sense and is accurate we can even simulate with Qiskit[6] (in Colab):



```
Total count: {'10': 1000}
```

If this is not visible yet (above) here are the outputs (below):



```
[8] plot_histogram(counts)
```

[6] Qiskit also can calculate and produce (plot nicely in LaTeX) matrices for whole quantum circuits or parts thereof (so we can check the math).

Let's calculate the wave functions:

$$|\Psi_0\rangle = \left\{\bigcirc\bigcirc, \overline{\bullet}\bigcirc, \bigcirc\overline{\bullet}, \overline{\bullet}\,\overline{\bullet}\right\} \text{ and}$$

$$\begin{aligned}|\Psi_2\rangle &= \texttt{C-NOT}(|\Psi_0\rangle) = \\ &= \left\{\bigcirc\bigcirc, \overline{\bullet}\bullet, \bigcirc\overline{\bullet}, \overline{\bullet}\,\overline{\bigcirc}\right\} = \\ &= \left\{\bigcirc\left\{\bigcirc, \overline{\bullet}\right\}, \overline{\bullet}\left\{\bullet, \overline{\bigcirc}\right\}\right\} = \\ &= \left\{\bigcirc\left\{\bigcirc, \overline{\bullet}\right\}, \bullet\overline{\left\{\overline{\bigcirc}, \bullet\right\}}\right\} = \\ &= \left\{\bigcirc\left\{\bigcirc, \overline{\bullet}\right\}, \bullet\left\{\overline{\overline{\bigcirc}}, \overline{\bullet}\right\}\right\} = \\ &= \left\{\bigcirc, \bullet\right\}\left\{\bigcirc, \overline{\bullet}\right\}\end{aligned}$$

We are now left with the circuit in the middle (for which we also presented a Qiskit simulation). We have:

$$\Psi_1 = \left\{\bullet\bigcirc, \overline{\bigcirc}\bigcirc, \bullet\overline{\bullet}, \overline{\bigcirc}\,\overline{\bullet}\right\}$$

This follows from $|\Psi_0\rangle$ if we apply an `X` gate (as the circuit does) on the first qubit. Next, we have

$$|\Psi_3\rangle = \texttt{C-NOT}(|\Psi_1\rangle) == \left\{\bullet\bullet, \overline{\bigcirc}\bigcirc, \bullet\overline{\bigcirc}, \overline{\bigcirc}\,\overline{\bullet}\right\}$$

Note that traditional calculation matches this expression:

$$|\Psi_3\rangle = \frac{1}{2}\Big(|11\rangle - |00\rangle - |10\rangle + |01\rangle\Big)$$

However, if we implement this circuit in Qiskit the state vector at this point (i.e., for $|\Psi_3\rangle$) comes out as:

$$|\Psi_3\rangle = \frac{1}{2}\Big(|11\rangle - |00\rangle - |01\rangle + |10\rangle\Big)$$

We need to be mindful, always, when we check such calculations in Qiskit, because of the (widely known) change in how qubits are being ordered. With this we can calculate:

$$\begin{aligned}|\Psi_4\rangle &= \left\{\bigcirc\bullet, \overline{\bullet}\bigcirc, \bigcirc\overline{\bigcirc}, \overline{\bullet}\,\overline{\bullet}\right\} = \\ &= \left\{\bigcirc\left\{\bullet, \overline{\bigcirc}\right\}, \bullet\overline{\bigcirc}, \bullet\bullet\right\} = \\ &= \left\{\bigcirc\left\{\overline{\bigcirc}, \bullet\right\}, \bullet\left\{\overline{\bigcirc}, \bullet\right\}\right\} = \\ &= \left\{\bigcirc, \bullet\right\}\left\{\overline{\bigcirc}, \bullet\right\} = \\ &= \left\{\bigcirc, \bullet\right\}\overline{\left\{\bigcirc, \overline{\bullet}\right\}}\end{aligned}$$

And that finishes the proof because on the second wire the Hadamard gate reconstructs $|1\rangle$ (up to a phase which, however, does not affect the measurements) whereas a $|0\rangle$ emerges from the Hadamard gate on the first wire (right side in picture).

V. Bernstein-Vazirani

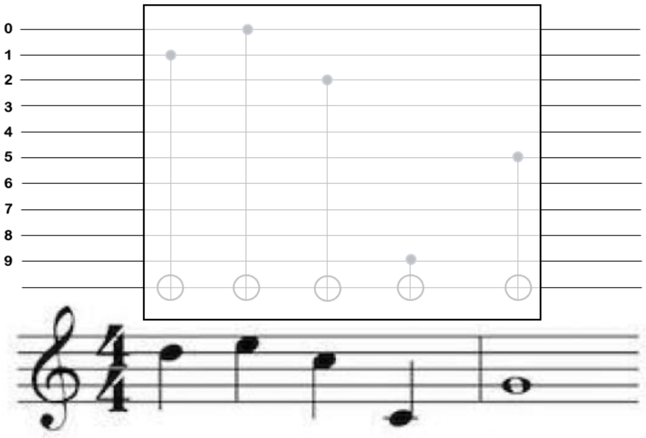
Despite the extraordinary power of today's computers, there are applications that are difficult for them to compute but seem to be easily "computed" by the quantum world: estimating the properties and behavior of quantum systems. While today's classical computers can simulate simple quantum systems, and often find useful approximate solutions for more complicated ones, for many such problems the amount of memory needed for the simulation grows exponentially with the size of the system simulated. In 1982, physicist Richard Feynman suggested that quantum mechanical phenomena could themselves be used to simulate a quantum system more efficiently than a naïve simulation on a classical computer.

In 1993, Bernstein and Vazirani showed that quantum computers could violate the extended Church-Turing thesis[7]. Quantum computation is the only model of computation to date to violate the extended Church-Turing thesis, and therefore only quantum computers are capable of exponential speedups over classical computers.

A. *How the Bernstein-Vazirani Algorithm Works*

We owe this description of the Bernstein-Vazirani algorithm to Diana Franklin. She refers to the kind of example that follows as "simultaneous computation with oracles." More people should take her two ([10], [11]) excellent EdX courses on Quantum Computation. Let's imagine a quantum circuit with $n+1$ inputs and such that any of the first n wires could control a `C-NOT` gate located on the remaining (bottom) wire.

Here's an example with $n = 10$.

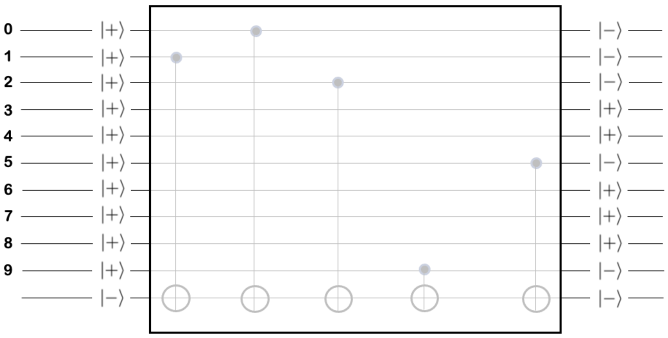


The theme from "Close Encounters of the Third Kind" is there to reinforce the pattern but also to allow me to say that the sequence (order) of the `C-NOT` gates is not relevant. The circuit itself is called an *oracle* and it hides a "secret" string of controls to the gates on the bottom wire. The task is to determine this string. The question is how fast can we determine the string (in this case `1110010001` that we could also write as $\{0, 1, 2, 5, 9\}$ to emphasize it's actually a set).

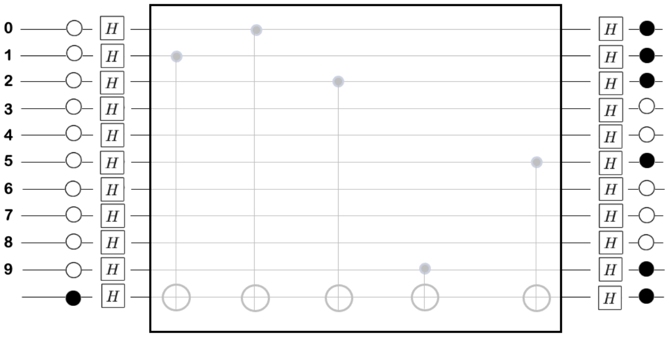
How fast can we determine this "characteristic" of the oracle? In the classical sense we need n tries, in each case feeding a $|1\rangle$ on a single line $0 \leq i \leq (n-1)$ and $|0\rangle$ on all other inputs, including the one at the bottom. A change in the output of the bottom wire will tell us that i is in the set. Can we do better? Yes, in the quantum case we need just one try.

[7] The extended Church-Turing thesis is a foundational principle of computer science that said that the performance of all computers was only polynomially faster than a probabilistic Turing machine. Bernstein and Vazirani's quantum algorithm offered an exponential speedup over any classical algorithm for a certain computational task called recursive Fourier sampling. Another example of a quantum algorithm demonstrating exponential speedup for a different computational problem was provided in 1994 by Dan Simon.

We will be using the knowledge we built when we studied the circuit from Phase Kickback. If we feed $|+\rangle$'s on the first n inputs and $|-\rangle$ on the bottom wire the changes in the outputs will only occur on the wires acting as control qubits for the gates at the bottom. The other inputs remain unchanged.



So the pattern has emerged, in one step, as the output. If we prefer to work in the computational basis we do this:



So one step is enough.

VI. Conclusion

We hope the paper makes a strong case for how much can be introduced in an eight weeks (half-semester) class using a formalism that as simple as elementary (i.e., middle school) pre-algebra. The topics chosen address the foundations (see also [8]). The "abacus" calculations are complete, accurate and lead naturally (we believe) into the more traditional mathematical approaches. A second eight weeks class could introduce that formalism (starting with a review of elementary probability, trigonometry and complex numbers) and using it to review and reinforce the topics presented here, then add chapters on Quantum Search (Grover) and Quantum Factoring (Shor's algorithm). We have been using this approach in the classroom for the last two years and the data we collected thus far, on how effective this approach is, looks very promising.

This is probably a good place to review the basic rules of the "abacus" as presented in this paper:

- the superposition operator describes a qubit as a set of possible outcomes, with their associated probabilities
- the phase operator acts like the unary minus sign in a multiplication (or product)
- when applied to a set of possible outcomes the phase operator changes the sign (phase) on each of the elements in that set (of possible outcomes)
- the tensor product between two qubits is the cartesian product between the superposition sets representing those qubits (and the order matters)
- if we allow (as we did when we explained teleportation) complex coefficients in the superposition operator's representation introduced in the first few lines of the paper then the full generality of representation for qubits is achieved.

Compared to the ZX-calculs the system presented in our paper resembles the "slow-motion" replays in televised sports. A slow-motion replay is an exponential process and nobody would ever argue that it would be useful, advisable or otherwise meaningful to watch an entire game in slow motion. Graph-rewriting (mentioned in passing twice, with examples) has significant advantages over the string rewriting techniques that we presented. However graph rewriting is in effect an orthogonal process to what we advocated here; it can't provide any of the insight this slow-motion "abacus" technique provides. Furthermore, a lot of the teaching (and learning) that happens when a student is first introduced to a complicated topic is, of necessity, of the slow-motion type.

Acknowledgment

The authors would like to thank their colleagues in the QED-C (especially the Workforce Development TAC) for numerous talks around how best these topics could be introduced to a noob (neophyte) audience. QED-C is a broad international group of stakeholders from industry, academia, national labs and professional organizations that aims to enable and grow the quantum industry and its associated supply chain. QED-C was established with support from NIST as part of the federal strategy for advancing QIST as per the National Quantum Initiative Act in 2018.

References

[1] Bob Coecke and Aleks Kissinger, *Picturing Quantum Processes—A First Course in Quantum Theory and Diagrammatic Reasoning*, Cambridge University Press, 2017.

[2] Sophia Economou, Terry Rudolph and Edwin Barnes, *Teaching quantum information science to high-school and early undergraduate students*, available at https://arxiv.org/abs/2005.07874, 2020

[3] Edwin Barnes, *Teaching QIS at the QISE Summer School organized by Virginia Tech* presentation at Spring 2022 CSAAPT Semi-Virtual Meeting available at https://indico.phys.vt.edu/event/48/contributions/996/

[4] John van de Wetering, *ZX-calculus for the working quantum computer scientist*, https://arxiv.org/abs/2012.13966v1, 2020

[5] Bob Coecke, "Quantum picturalism", *Contemporary Physics*, 2010.

[6] Terry Rudolph, *Q is for Quantum*, Terrence Rudolph 2017.

[7] Marek Narozniak, Simulating Quantum Teleportation, March 2020 available at https://mareknarozniak.com/

[8] Adrian German, Marcelo Pias, Qiao Xiang *On the Design and Implementation of a Quantum Architectures Knowledge Unit for a CS Curriculum* ACM, SIGCSE 2023, https://doi.org/10.1145/3545945.3569845

[9] Wikipedia article, https://en.wikipedia.org/wiki/Quantum_teleportation

[10] Diana Franklin, *Quantum Computing for Everyone (Part 1)*, EdX course at https://www.edx.org/course/quantum-computing, 2021.

[11] Diana Franklin, *Quantum Computing for Everyone (Part 2)*, 2021. www.edx.org/course/introduction-to-quantum-computing-for-everyone-2