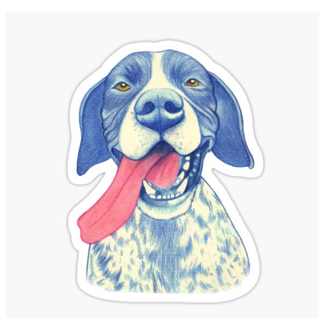
The Little Qubitzer

Being a conversation between:

DAN-ADRIAN GERMAN (IU)
MARCELO RITA PIAS (FURG)
and QIAO XIANG (XIAMEN)

dgerman@indiana.edu



```
Today I sniffed
Many dog butts - I celebrate
By kissing your face.
```

This is the booklet used for the SIGCSE 2024 Workshop 306
Fri 22 Mar 2024 19:00 - 22:00 at Meeting Room E146 - Workshop
Start by watching Terry's video here: `https://www.qisforquantum.org/`
Slides as written in real-time during the workshop can be found here:
`https://legacy.cs.indiana.edu/~dgerman/2024/sigcse/rctmpx-01.pdf`

SIGCSE 24
Technical Symposium

Wed 20 - Sat 23 March 2024 Portland, Oregon, United States

Most recent update on 07/05/2024 at 5:48pm GMT

[1] `https://legacy.cs.indiana.edu/~dgerman/`

Contents

Foreword

If you have tried teaching yourself (or others) the basics of Quantum Computing (or Quantum Information Science) using one of the many already available wonderful textbooks only to feel overwhelmed by the mathematical apparatus (or just its syntax) know that there is an alternative pedagogical approach that acts as a bridge to the standard quantum computation curriculum but in which the mathematics starts to feel supportive, organic and helpful, instead of oppressive.

We should stress that, in our view, there is nothing wrong with mathematics; mathematics by itself is not oppressive. We use mathematics to help describe things going on in the physical world around us. To a physicist, the math is an inextricable part of our understanding. Unfortunately, not everyone is good at math, and most have little, if any, training in physics.

A system devised by Terry Rudolph based on a rewriting system (which we call the "Quantum Abacus") can effectively be used to guide our students to the place where we would all like them to be, no less, but going through a stage where they feel that they "really understand" what our mathematics "means" in terms of stuff that goes on in the physical world.

Preface

In 2017 Terry Rudolph proposed a method of teaching quantum mechanics and quantum computing using only the simple rules of arithmetic to students as early as sixth grade. The method is incredibly effective and in a series of papers we showed how we use it to introduce superposition, phase, interference and entanglement with virtually no mathematical overhead.

Furthermore we showed that a complete eight week introductory course (for computer science sophomores) has been built around this approach with the following milestones: quantum gates and circuits, phase kickback, the Deutsch-Josza algorithm, Bernstein-Vazirani and the extended Church-Turing thesis, the GHZ game and quantum teleportation.

There is general consensus that the actual mathematics behind quantum computation is an inevitable and desirable destination for our students. But for those students that lack an adequate mathematical background (HS and younger students) one can reliably use Terry's method (i.e., computing with misty states, also referred to as The Quantum Abacus) to communicate a visual and entirely operational understanding of key quantum computing concepts without resorting to complex numbers or matrix multiplication.

Here[2] we present concrete evidence that the approach can create a genuine bridge to the actual mathematics behind quantum computation: we start with superdense coding and Grover's algorithm (to illustrate how effective the system is) then we identify an elementary break-even point when creating a three-qubit W-entangled state. Terry's (misty-state) formalism is based on a paper by Shih that Toffoli plus Hadamard gates are universal. When trying to create the W-entangled state we need to accommodate rotations and we must use controlled-Hadamard gates. And this is what allows for a break-even point: a Hadamard gate controlled by the output of another Hadamard gate breaks the ubiquitous symmetry in Terry's system, and from then on one has to carry around (i.e., specify) the actual probability amplitudes in misty states.This means that students can proceed to developing, in parallel, with (extended) misty states and Dirac notation. And after crossing that bridge we have an entirely conventional Quantum Computation course, but the intuition we acquired while computing with misty states remains with us.

[2] `https://legacy.cs.indiana.edu/~dgerman/2024/epj-paper/mar-10.pdf`

Chapter 1

Superposition

What is this: ◯	A (classical) bit value.
What is this: {◯}	A qubit value also known as $\lvert 0\rangle$ or $\begin{pmatrix}1\\0\end{pmatrix}$.
What's the difference?	The second one is a set of outcomes.
How many outcomes?	In this case just one.
What's this: ●	The other (classical) bit value.
What's this: {●}	The other qubit value.
In what basis?	Computational basis.
How else can we write it?	We can write it as $\lvert 1\rangle$ or $\begin{pmatrix}0\\1\end{pmatrix}$
Is that important?	No, at the moment: no.

What is this: {◯,●}	A set of two possible outcomes.
What's their likelihood?	They seem equally likely.
They are.	Good to know.
We will show that shortly.	Is that calculation important?
No, at the moment: no.	As an outcome, what is ◯
It's like heads (see below). 	This is also heads. 
So what is ● in this interpretation?	It's the (common) tails. 
So what do you have for {◯,●}	I'm thinking, something like this: 
Not bad...	I thought so, too.

What is this: {◯, ●, ●}	A set of two possible outcomes.
One of them seems more likely.	It is. But, by how much?
We could calculate the probabilities.	Great, let's do that later.
Is order important?	No, but an order is preferred.
What is this: {●, ●}	A set of just one outcome.
So is it: {●}	Yes, and we often write just ●
What is: {●, {◯, ●}}	Same as {●, ◯, ●} = {◯, ●, ●}
So, embedding sets of outcomes will act as the set union of those outcomes.	Is order important?
No, but an order is preferred.	I see: heads go first, then tails.
Let's now show how we calculate the probabilities of occurrence.	For that we need some notation.
Assume you have n heads and m tails.	Above, $n = 1$ and $m = 2$.
Then chance of heads is $\frac{n^2}{n^2+m^2} = \frac{1}{5}$.	Likewise chance of tails is $\frac{m^2}{n^2+m^2} = \frac{4}{5}$
The two add up to 1.	And {◯, ●, ●} = $\sqrt{\frac{1}{5}}\|0\rangle + \frac{2}{\sqrt{5}}\|1\rangle$
Is that important?	It's quite meaningful, but at the moment it's not a crucial concept.

Can this be simplified: {●, ○, ●, ●, ○, ●}	Yes. First, let's clean it up: {○, ○, ●, ●, ●, ●}
Here's another way this can be written: {○, ●, ●, ○, ●, ●}	Which can be written: {{○, ●, ●}, {○, ●, ●}}
Which should be equal to: {○, ●, ●}	And I can tell you why.
Why?	Recall the early formulas?
For probabilities?	Yes.
They were functions of n and m.	Replace n and m with kn and km.
The fractions get simplified by k^2.	Yes, since $k \in \mathbf{N}$ and $k \neq 0$.
Let me try one formula see how it goes.	Be my guest.
$\frac{(kn)^2}{(kn)^2+(km)^2} = \frac{k^2n^2}{k^2(n^2+m^2)} = \frac{n^2}{n^2+m^2}$	Probability amplitude: $\sqrt{\frac{n^2}{n^2+m^2}}$
For us, here, $n = k = 1$ and $m = 2$.	And the probability of ○ is the same.
A similar argument works for ●.	So simplifications are possible.
This will come in handy later.	I hope so.

Chapter 2

Multiple Qubits

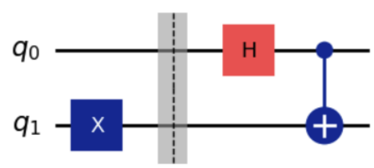
What is this: ◯●	A sequence of two (classical) bits.
The order is now very important.	Yes, ◯● $\neq$ ●◯
But {◯, ●} = {●, ◯}	Yes, those are sets (of outcomes).
With sets, the order is not important.	Although usually an order is preferred.
But with sequences ...	... the order is not negotiable.
What is this: {◯●}	A system of two qubits.
How else can we refer to it?	It's $\|01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$
What's {◯●, ●◯}	It's $\|\Psi^+\rangle = \frac{1}{\sqrt{2}}\|01\rangle + \frac{1}{\sqrt{2}}\|10\rangle$
Why are we pointing these out?	To show we're learning the real thing.

Anything special about $\|\Psi^+\rangle$?	Yes, it's one of the Bell states.
Here's how we obtain this state:	This is Qiskit code.

```
!pip install qiskit pylatexenc
```

```
from qiskit import QuantumCircuit
```

```
qc = QuantumCircuit(2)
qc.x(1)
qc.barrier()
qc.h(0)
qc.cx(0, 1)
qc.draw(output='mpl')
```




```
from qiskit.quantum_info import Statevector
sv = Statevector.from_instruction(qc)
sv.draw('latex')
```

$$\frac{\sqrt{2}}{2}|01\rangle + \frac{\sqrt{2}}{2}|10\rangle$$

In Google Colab.	That's right!
All you need is a gmail account.	Colab notebooks are very useful.
What are those boxes?	One- and two-qubit gates.
When are we talking about them?	Soon.
Can't wait.	Same here.

Chapter 3

Phase

How do we define quantum gates?	We start by how it works on $\bigcirc$ and $\bullet$.
Can you give me an example?	Let's take the `Z` gate: $\mathtt{Z}(\bigcirc) = \bigcirc$.
So for $\bigcirc$ the `Z` gate is the identity.	Yes but $\mathtt{Z}(\bullet) = \overline{\bullet}$
What's the bar on top of the bit value?	It's called phase and acts like a minus.
This section talks about phase.	How it acts on sets and sequences.
For sets it behaves like this: $\overline{\{s_1, s_2, ..., s_n\}} = \{\overline{s_1}, \overline{s_2}, ..., \overline{s_n}\}$	Every term of the set changes sign.
Like changing the sum of an addition.	Yes, that's what I see happening here.
For sequences of bits we have: $b_1...\overline{b_i}...b_n = \overline{b_1...b_i...b_n} \quad 1 \leq i \leq n$	We should also add: $\overline{\overline{x}} = x \ \ \forall x$
So the phase operator is its own inverse.	Yes. Time for some exercises.

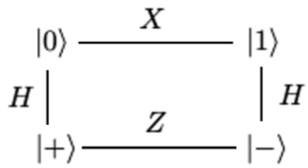
What is $\overline{\bullet}\,\overline{\circ}\,\bullet$	Same as $\overline{\overline{\bullet\circ\bullet}} = \bullet\circ\bullet$
What is this: $\{\circ, \bullet\}\bullet$	It's a sequence of two qubits.
The first qubit is a superposition.	The second one is $\{\bullet\}$
We usually write $\{\bullet\} = \bullet$	And we can calculate: $\{\circ, \bullet\}\bullet = \{\circ\bullet, \bullet\bullet\}$
Like a cartesian product.	Exactly.
So if you want to calculate $\overline{\{\circ, \bullet\}\bullet}$	... you can do it in two ways and you should still obtain the same result.
So in summary...	... and as a reminder, we write:
$-\circ\circ\bullet = \overline{\circ\circ\bullet} = \overline{\circ}\circ\bullet = \circ\overline{\circ}\bullet = \circ\circ\overline{\bullet}$	
That concludes our intro to phase.	Time to discuss one-qubit gates.
We'll introduce three such gates:	`X`, `Z` and `H`.
`Z` is a phase flip gate.	`X` is the `NOT` gate.
And `H` is the Hadamard gate.	Also known as the `PETE` box.
Later we'll introduce more gates.	Like `S` (which is inaccessible now).

Chapter 4

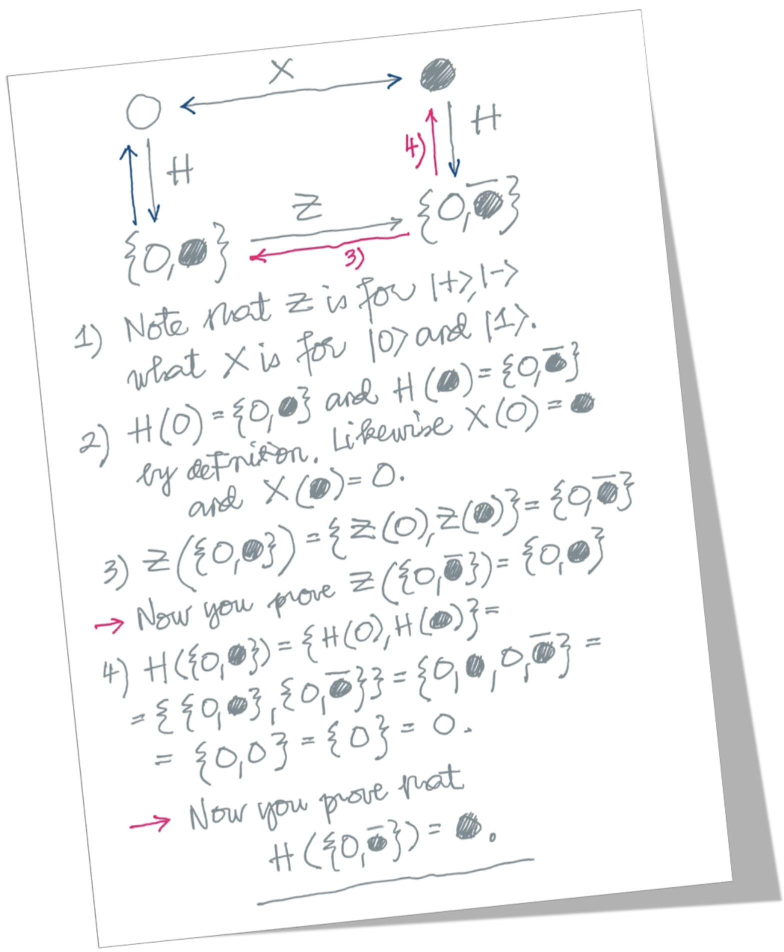
One-Qubit Gates

Fortunately, learning quantum computing (QC) is much easier than learning quantum mechanics (QM) because it deals with a very simple subset of QM.	I think your argument has three parts and starts with: a qubit—the foundation of QC—is the simplest non-trivial quantum system. Am I right?
Indeed. Second, you never have to solve the Schrödinger equation, or even learn what it is because the quantum systems that carry out quantum computations evolve in a controlled manner based on the quantum gates applied to them.	Finally, there is a model of quantum computation already, so the most difficult aspect of quantum mechanics—the art of applying it to real systems—is, in fact, absent.
We start defining gates by postulating their behavior on ◯ and ●. We have already done this for the `Z` gate.	We further have for the `NOT` gate, as expected: `X`(◯) = ● and `X`(●) = ◯.
We also have `H`(◯) = {◯, ●}	Incidentally {◯, ●} is known as $\|+\rangle$
Likewise `H`(●) = {◯, $\overline{●}$}	And {◯, $\overline{●}$} is also known as $\|-\rangle$
Having defined the base behavior of each gate we now need a general rule for unitary evolution.	Here it is: $f(\{s_1, ..., s_n\}) = \{f(s_1), ..., f(s_n)\}$

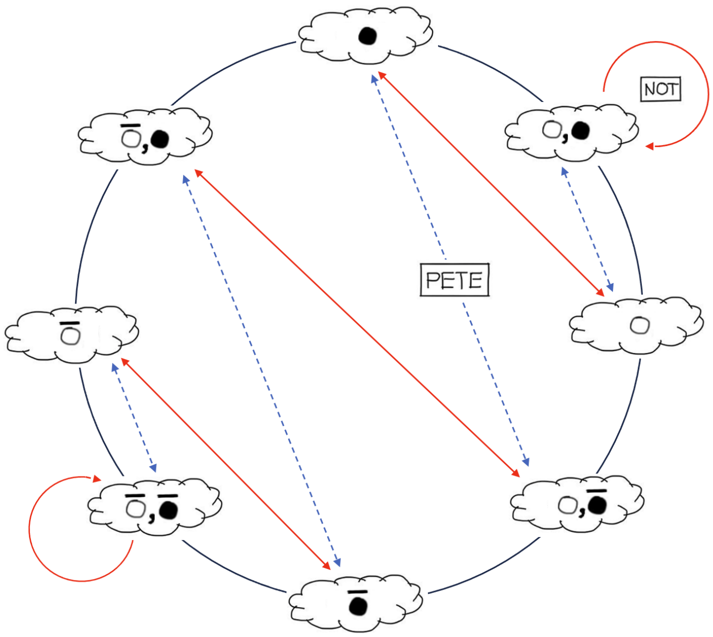
Here's an exercise. Prove that the following holds:



We'll work this out together. Here's a start:



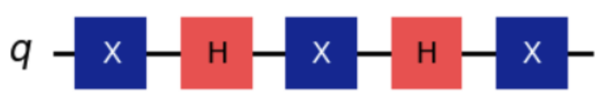
Another exercise. Prove that the following holds:



The unit circle state machine from Andrew Helwer's presentation[1] and slides[2].

[1] https://www.microsoft.com/en-us/research/video/quantum-computing-computer-scientists/
[2] https://ahelwer.ca/files/qc-for-cs.pdf

Then use it to demonstrate this:




```
from qiskit.quantum_info import Statevector
sv = Statevector.from_instruction(qc)
sv.draw('latex')
```

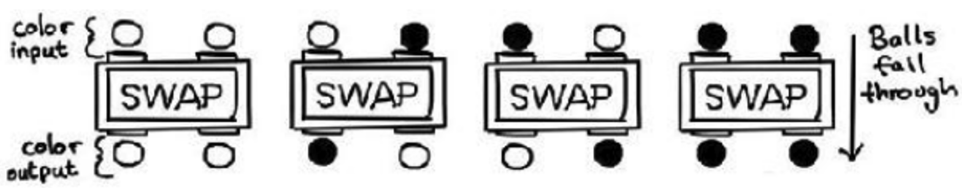
$$-|0\rangle$$

Chapter 5

Two-Qubit Gates

We introduce two two-qubit gates. One of them is `SWAP`:

An early example of a two-qubit gate from the book.



The gate does not swap the coins per se. Their states are being swapped.

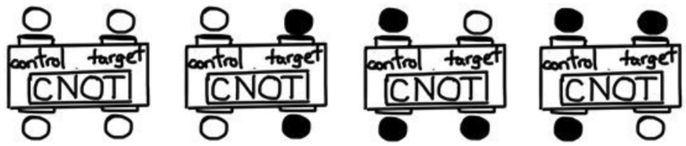
The coin on the left is from Germany. The one on the right is from Italy.

The other one is the `C-NOT`. Or “controlled-`NOT`.”

Its behavior is shown on the next page. The approach is similar.

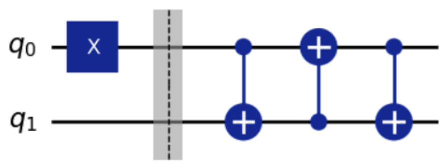
The definition involves four cases.

All possible inputs are considered:



Here's a circuit with three `C-NOT` gates.

Can this circuit be simplified?




```
sv = Statevector.from_instruction(qc)
sv.draw('latex')
```

$|10\rangle$

The answer is: yes.

But we won't address that here.

Here we introduce some notation.

Then use that to define the gate.

A two-qubit gate has two parameters (inputs). In the case of the `C-NOT` gate one of them is the control qubit.

As we can see from the circuit above (no matter how we look at it) sometimes the control qubit is the first qubit sometimes it is the second.

We introduce a little arrow on top of the `X` to indicate where the controlled qubit is (the arrow points to the target).

Then, we have:

$\overrightarrow{\texttt{X}}(●○) = ●●$ but $\overleftarrow{\texttt{X}}(●○) = ●○$

And the entire circuit could be described as follows:

$$\overrightarrow{\texttt{X}}(\overleftarrow{\texttt{X}}(\overrightarrow{\texttt{X}}(\texttt{X}(○)○))) = ○●$$

Make a note to prove this to be `SWAP`.	We need the extended system for that.
Absent that we need four cases.	One other thing is of interest here.
What is it?	That Qiskit orders qubits backwards.
Do they have a good argument?	Yes, and we should discuss it.
How do we read qubits.	Top down, with the circuit horizontal.
Show me an example.	In the picture we just saw: $q_0 q_1$.
How does Qiskit order qubits?	First rotate circuit 90° clockwise.
Then read qubits left to right.	Which gives us the order: $q_1 q_0$.
Yes.	So that's exactly backwards.
I have one other point to make.	About the Hadamard gate?
It is tempting to believe that `H`($\bigcirc$) is: 	But it isn't.
Unitary evolution is reversible.	`H`(`H`($\bigcirc$)) = $\bigcirc$

After chapter 12 come here and prove[1] that[2]

$$\overrightarrow{\mathtt{X}}(\overleftarrow{\mathtt{X}}(\overrightarrow{\mathtt{X}}(\{\alpha_1\bigcirc, \alpha_2\bullet\}\{\beta_1\bigcirc, \beta_2\bullet\}))) = \{\beta_1\bigcirc, \beta_2\bullet\}\{\alpha_1\bigcirc, \alpha_2\bullet\}$$

Right now that's not possible (other than case by case).

That's the difference between this formalism and the extended one.

[1] Do not use matrices. (What are those?) Can we? Should we? And if so, when?

[2] Which would prove that $\overrightarrow{\mathtt{X}}(\overleftarrow{\mathtt{X}}(\overrightarrow{\mathtt{X}}(q_0q_1))) = \mathtt{SWAP}(q_0q_1) = q_1q_0$ or that $\overrightarrow{\mathtt{X}}\,\overleftarrow{\mathtt{X}}\,\overrightarrow{\mathtt{X}} \equiv \mathtt{SWAP}$

Chapter 6

Entanglement

Is this quantum state[1] an entangled[2] state?

$$|\Psi_1\rangle = \sqrt{\frac{1}{4}}\begin{pmatrix}1\\1\\1\\1\end{pmatrix}$$

Answer: no, because

$$\left\{ ○○, ○●, ●○, ●● \right\} = \left\{ ○, ● \right\}\left\{ ○, ● \right\}$$

Is this quantum state an entangled state?

$$|\Psi_2\rangle = \frac{|00\rangle + |01\rangle}{\sqrt{2}}$$

Answer: no, because

$$\left\{ ○○, ○● \right\} = \left\{ ○ \right\}\left\{ ○, ● \right\} = ○\left\{ ○, ● \right\}$$

Consider the Bell states[3]:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|0\rangle_A \otimes |0\rangle_B + |1\rangle_A \otimes |1\rangle_B)$$

$$|\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$

$$|\Psi^\pm\rangle = \frac{|01\rangle \pm |10\rangle}{\sqrt{2}}$$

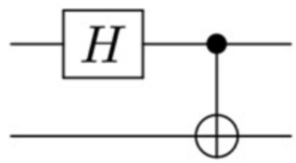
Note the many notations used aim to make you immune to such variations.

[1] Here we need to convert the given state to misty state formalism first.

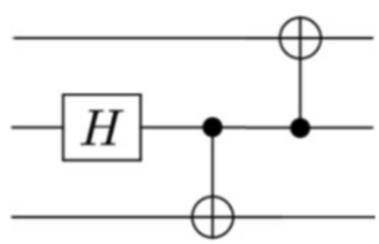
[2] As a reminder, an entangled state of a composite system is a state that cannot be written as a product state of the component systems.

[3] `https://quantumcomputinguk.org/tutorials/introduction-to-bell-states`

Calculate the state produced by this circuit:



Calculate the state produced by this circuit:

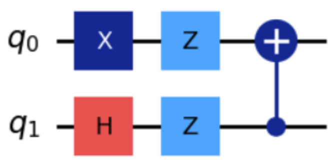


Which Bell state is created by this circuit:

```
!pip install qiskit pylatexenc
```

```
from qiskit import QuantumCircuit
```

```
qc = QuantumCircuit(2)
qc.h(1)
qc.x(0)
qc.z(1)
qc.z(0)
qc.cx(1, 0)
qc.draw(output='mpl')
```

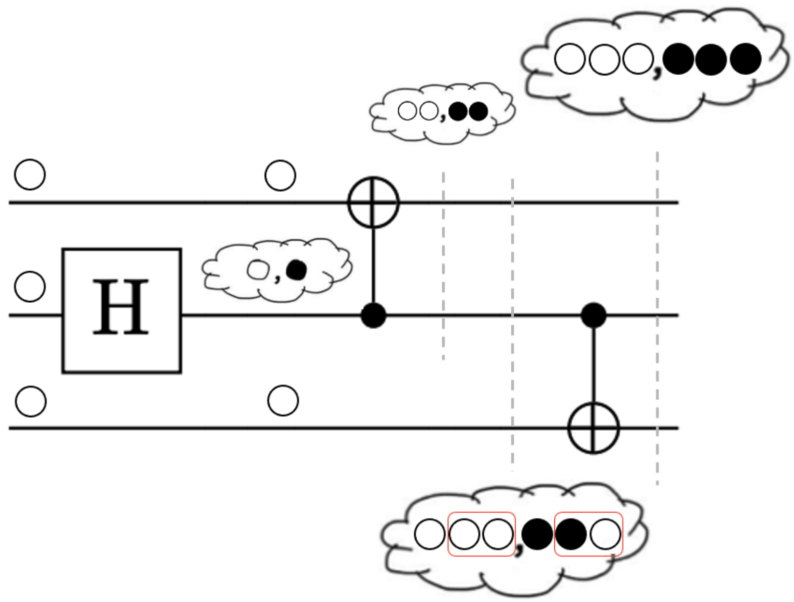



```
from qiskit.quantum_info import Statevector
sv = Statevector.from_instruction(qc)
sv.draw('latex')
```

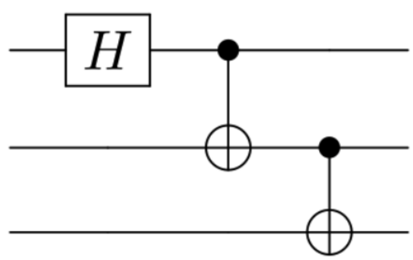
$$-\frac{\sqrt{2}}{2}|01\rangle + \frac{\sqrt{2}}{2}|10\rangle$$

Explain.

Explain what happens in this picture.



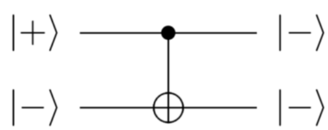
Calculate the state produced by this circuit:



Chapter 7

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[1] 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

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

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

[1]Do it.

So now we calculate:

$$
\begin{aligned}
\overrightarrow{\mathtt{X}}(|{+}{-}\rangle) &= \overrightarrow{\mathtt{X}}(\left\{\bigcirc, \bullet\right\}\left\{\bigcirc, \overline{\bullet}\right\}) \\
&= \overrightarrow{\mathtt{X}}(\left\{\bigcirc\bigcirc, \bigcirc\overline{\bullet}, \bullet\bigcirc, \bullet\overline{\bullet}\right\}) \\
&= \left\{\overrightarrow{\mathtt{X}}(\bigcirc\bigcirc), \overrightarrow{\mathtt{X}}(\bigcirc\overline{\bullet}), \overrightarrow{\mathtt{X}}(\bullet\bigcirc), \overrightarrow{\mathtt{X}}(\bullet\overline{\bullet})\right\} \\
&= \left\{\bigcirc\bigcirc, \bigcirc\overline{\bullet}, \bullet\bullet, \bullet\overline{\bigcirc}\right\} \\
&= \left\{\bigcirc\bigcirc, \bigcirc\overline{\bullet}, \overline{\bullet}\overline{\bullet}, \overline{\bullet}\bigcirc\right\} \\
&= \left\{\bigcirc\bigcirc, \bigcirc\overline{\bullet}, \overline{\bullet}\bigcirc, \overline{\bullet}\overline{\bullet}\right\} \\
&= \left\{\bigcirc\left\{\bigcirc, \overline{\bullet}\right\}, \overline{\bullet}\left\{\bigcirc, \overline{\bullet}\right\}\right\} \\
&= \left\{\bigcirc, \overline{\bullet}\right\}\left\{\bigcirc, \overline{\bullet}\right\} \\
&= |{-}{-}\rangle
\end{aligned}
$$

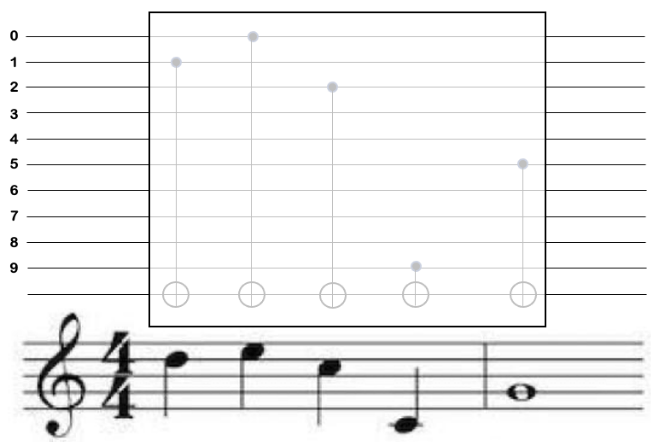
This concludes the proof.

Chapter 8

Bernstein-Vazirani

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?

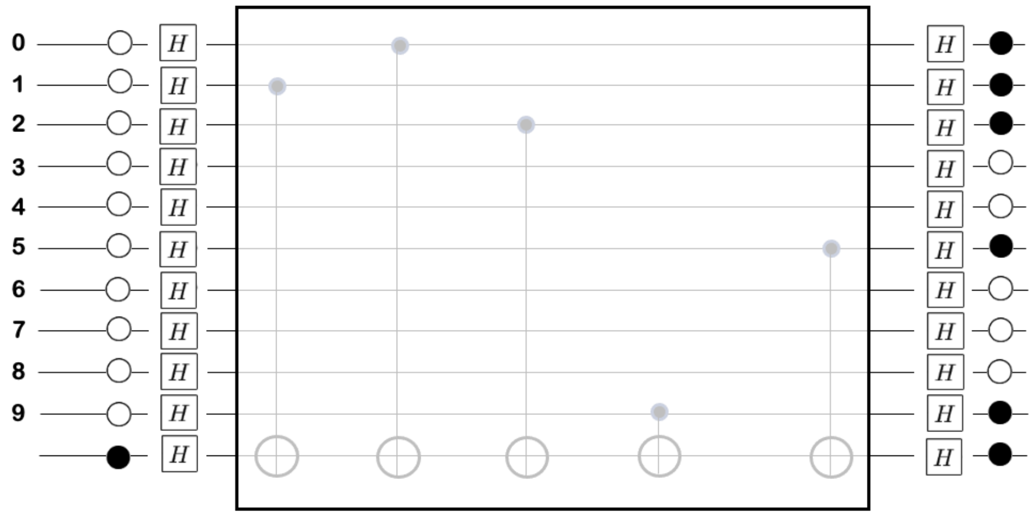
In the classical sense we need n tries, in each case feeding a $|1\rangle$ on a single line $0 \leq i \leq (n-1)$ and $|0\rangle$ on all other inputs, including the one at the bottom. A change in the output of the bottom wire will tell us that i is in the set. Can we do better? Yes, in the quantum case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

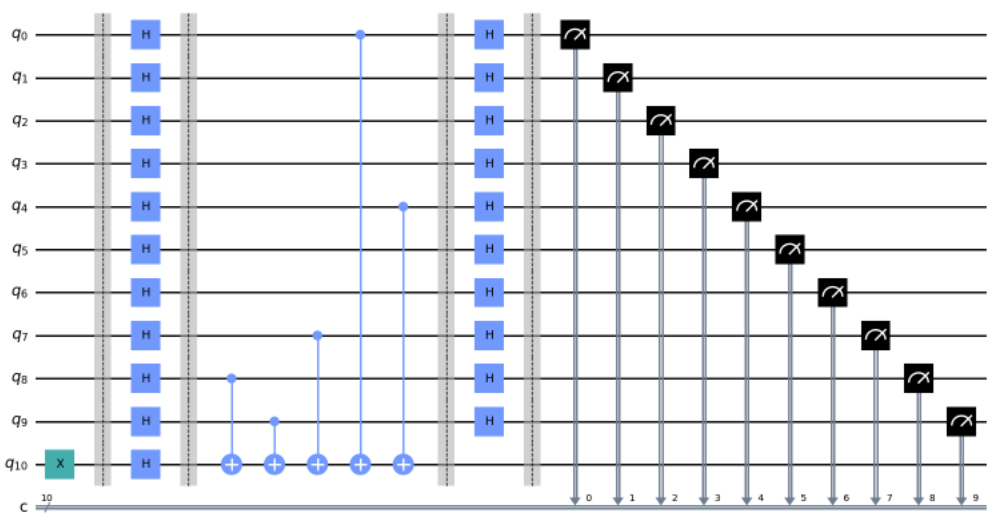
Show how you can find the secret string in one try.

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[1]. 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.

[1] 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.

Explain the relative significance of these pictures.





Same question for these pictures:

```
[13] !pip install qiskit qiskit-aer
```

```
[14] import qiskit
```

```
[15] from qiskit import *
```

```
[16] secretnumber = '1110010001'
     indices = [1, 0, 2, 9, 5]
     n = len(secretnumber)
     circuit = QuantumCircuit(n+1,n)
     circuit.x(n)
     circuit.barrier()
     circuit.h(range(n))
     circuit.h(n)
     circuit.barrier()
     for index in indices:
       circuit.cx(n - index - 1, n)
     circuit.barrier()
     circuit.h(range(n))
     circuit.barrier()
     circuit.measure(range(n),range(n))
```

```
[18] simulator = Aer.get_backend('qasm_simulator')
     result = execute(circuit, backend=simulator, shots=1).result()
     print(result.get_counts(circuit))

     {'1110010001': 1}
```

Chapter 9

Deutsch-Josza

The Deutsch-Jozsa algorithm[1] 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.

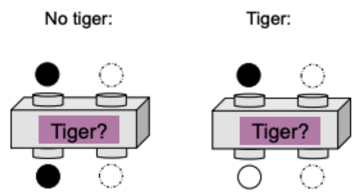
The setup of the game is as follows: there are two doors, one labeled with a white circle, the other one with a black circle. There is a button on the wall that opens both doors simultaneously. It is not possible to open only one door. There is money behind at least one door. There may or may not be 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

[1] 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.

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?”.

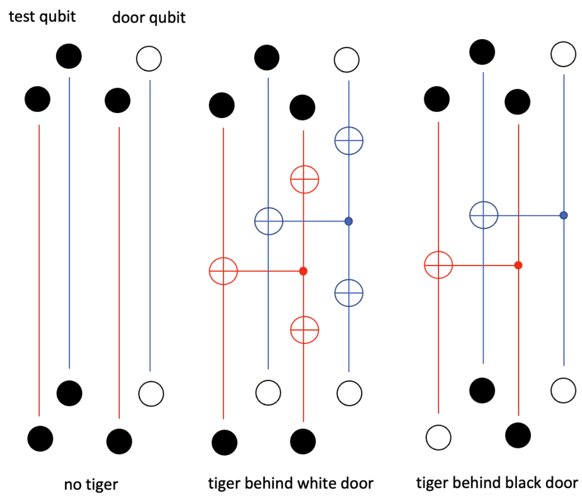
You are allowed to query this box once (and only once) to check whether there is a tiger. The way the box works is as follows: the box has two input ports and two output ports. You always input a black marble in the left input, and in the right you insert a marble whose color matches the door you want 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:

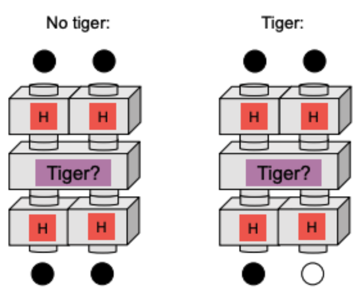


The “Tiger?” box is also called an *oracle*. If we only have access to classical information processing, then it is clear that the “Tiger?” box needs to be queried twice in order to be sure there is no tiger present. You would have to use it once for each one of the two doors. The point of this game 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 prove our claim using the “Quantum Abacus”. Here’s a CHALLENGE for you: can you sketch the three oracles below?

Here are the three oracles and how they function:



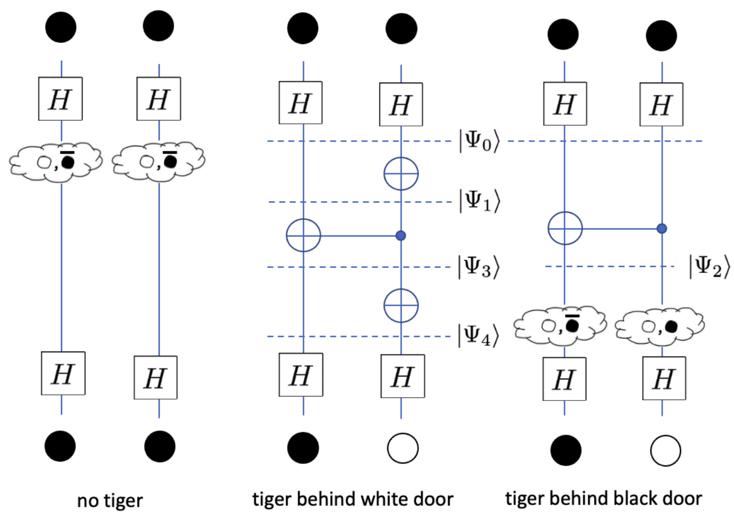
By adding additional gates above and below the oracle, it is possible to determine whether or not a tiger is present in one shot. This is shown below: if two black marbles are input into the circuit, then a white output signifies the presence of a tiger, regardless of which door the tiger is behind.



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 [... it] 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."

How the Deutsch Algorithm Works

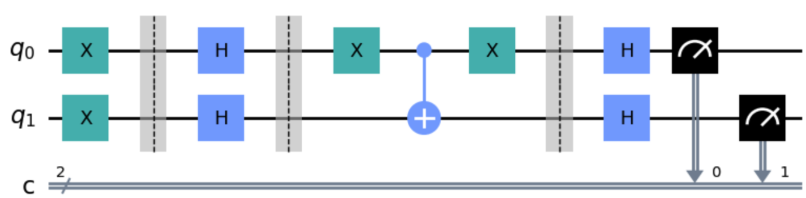
For the non-classical case, involving quantum gates, we need to add Hadamard gates (as shown) before and after the oracle, in each of the three cases. Then, by changing the question we ask (we no longer have a test qubit and a door qubit, we now just drop two $|1\rangle$ qubits) we will be able to get 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: $\mathtt{H}(\mathtt{H}(\bullet)) = \bullet$ as we showed from the beginning (page 10) 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[2] (in Colab):

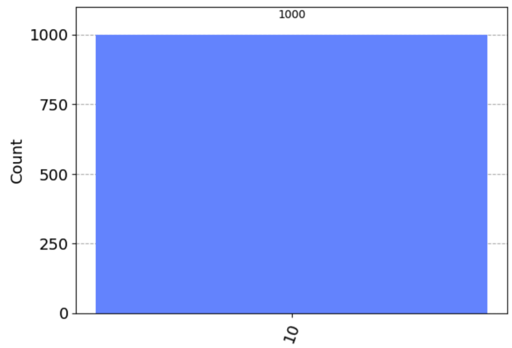
```
Total count: {'10': 1000}
```



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

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

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



Here's a CHALLENGE for you: can you please calculate $|\Psi_0\rangle$ and $|\Psi_2\rangle$?

Let's calculate the wave functions:

$$
\begin{aligned}
|\Psi_0\rangle &= \left\{\bigcirc\bigcirc, \overline{\CIRCLE}\bigcirc, \bigcirc\overline{\CIRCLE}, \overline{\CIRCLE}\,\overline{\CIRCLE}\right\} \text{ and} \\
|\Psi_2\rangle &= \overrightarrow{\mathtt{X}}(|\Psi_0\rangle) = \\
&= \left\{\bigcirc\bigcirc, \overline{\CIRCLE}\CIRCLE, \bigcirc\overline{\CIRCLE}, \overline{\CIRCLE}\,\overline{\bigcirc}\right\} = \\
&= \left\{\bigcirc\left\{\bigcirc, \overline{\CIRCLE}\right\}, \overline{\CIRCLE}\left\{\CIRCLE, \overline{\bigcirc}\right\}\right\} = \\
&= \left\{\bigcirc\left\{\bigcirc, \overline{\CIRCLE}\right\}, \CIRCLE\overline{\left\{\overline{\bigcirc}, \CIRCLE\right\}}\right\} = \\
&= \left\{\bigcirc\left\{\bigcirc, \overline{\CIRCLE}\right\}, \CIRCLE\left\{\overline{\overline{\bigcirc}}, \overline{\CIRCLE}\right\}\right\} = \\
&= \left\{\bigcirc, \CIRCLE\right\}\left\{\bigcirc, \overline{\CIRCLE}\right\}
\end{aligned}
$$

Here's another CHALLENGE for you: can you now calculate $|\Psi_1\rangle$ and $|\Psi_3\rangle$?

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

$$\Psi_1 = \Big\{\bullet\circ, \overline{\circ}\circ, \bullet\overline{\bullet}, \overline{\circ}\,\overline{\bullet}\Big\}$$

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

$$|\Psi_3\rangle = \overrightarrow{\mathtt{X}}(|\Psi_1\rangle) == \Big\{\bullet\bullet, \overline{\circ}\circ, \bullet\overline{\circ}, \overline{\circ}\,\overline{\bullet}\Big\}$$

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:

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

Quick CHALLENGE for you: please explain why.

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:

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

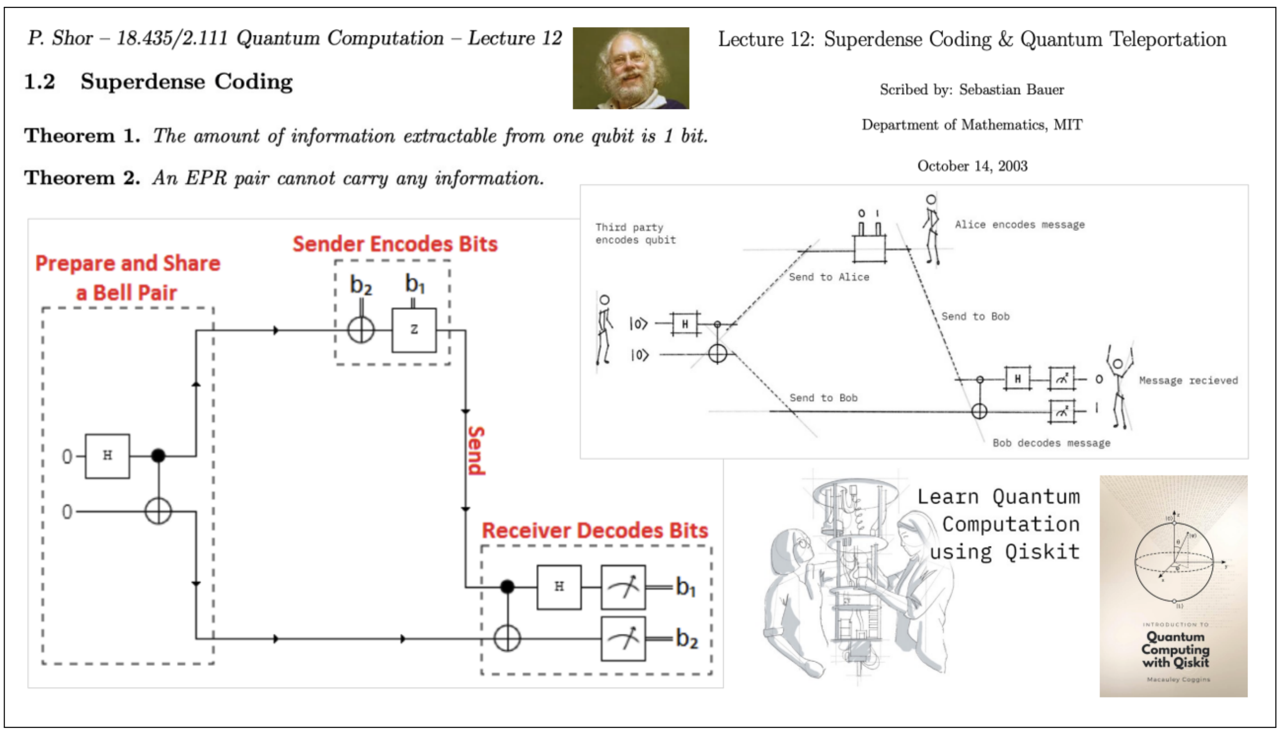
And that finishes the proof because on the second wire the Hadamard gate reconstructs $|1\rangle$ (up to a phase which, however, does not affect the measurements) whereas a $|0\rangle$ emerges from the Hadamard gate on the first[3] wire (right side in picture).

[3]Note how rotating the circuits clockwise 90° changes our perspective.

Chapter 10

Superdense Coding

We start with this slide as a reminder:

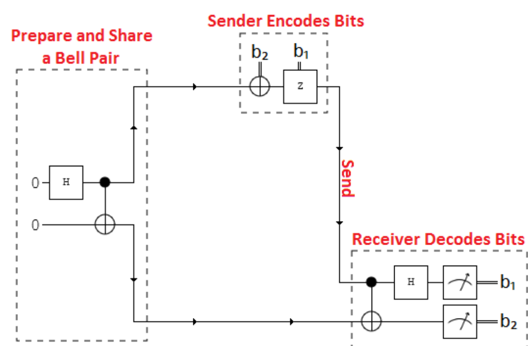


We will again show the (original) misty state formalism in action and before we start we need to state two theorems[1]. **Theorem 1**: The amount of information extractable from one qubit is 1 bit. **Theorem 2**: An EPR pair cannot carry any information. We are now ready for superdense coding.

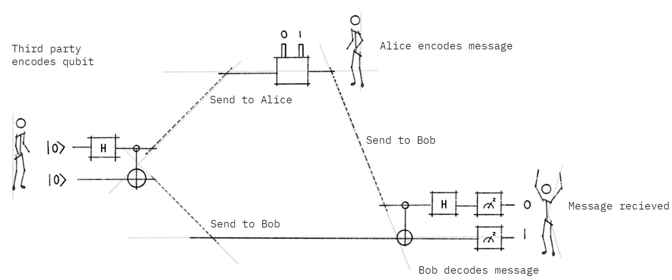
[1]See page 2 of the Peter Shor lecture notes at [26].

In superdense coding [27] the process starts with a third party, that we'll refer to as Charlie. Two qubits are prepared by Charlie in an entangled state. Charlie sends the first qubit to Alice and the second qubit to Bob. The goal of the protocol is for Alice to send two classical bits of information to Bob using her qubit. Based on the theorems stated before this section, that should not be possible. First, we said that a maximally entangled pair of qubits carries no information [29]. Furthermore the amount of information extractable from one qubit is 1 (one) bit. So how are we going to be able to send two bits of information if we put together these two resources? They just don't seem to add up (i.e., there seems to be a synergistic aspect at play here).

Here's the plan: Alice needs to apply a set of quantum gates to her qubit depending on the two bits of information that she wants to send. In [28] circuit diagram looks as follows:



A similar diagram can be found in the Qiskit textbook [27]:



Let's calculate the four cases and then compare one of them with Qiskit. Charlie starts with ◯◯ always. After the Hadamard gate the state is: H(◯)◯ = {◯,●}◯ which boils down to {◯◯,●◯} which is the input to the C-NOT gate. We then have $\overrightarrow{X}$({◯◯,●◯}) = {$\overrightarrow{X}$(◯◯), $\overrightarrow{X}$(●◯)} which is {◯◯,●●} namely the Bell state that we were expecting. Now Alice needs to take one of four courses of action based on the intended message she wants to send.

CHALLENGE for you: can you deduce the plan based on diagram(s) above?

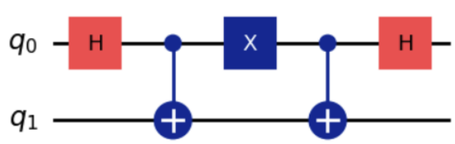
If the intended message is 00 Alice needs to apply the identity gate `I`, that is, she needs to just leave the qubit alone. You can see this in the first picture on the previous page where the message Alice wants to send (two bits) represents the controls of the `X` and `Z` gates. When the control is 0 such a gate does nothing. In that case the Bell state reaches Bob and after the `C-NOT` we have: $\overrightarrow{\texttt{X}}(\{◯◯, ●●\}) = \{\overrightarrow{\texttt{X}}(◯◯), \overrightarrow{\texttt{X}}(●●)\}$ which becomes $\{◯◯, ●◯\} = \{◯, ●\}◯$. Now the Hadamard gate acts as follows: $\texttt{H}(\{◯, ●\})◯ = ◯◯$. So that part was easy.

If the intended message is 01 Alice needs to apply an `X` gate. It's easy to check that the pictures are in fact consistent; but while the second picture shows a black box that Alice controls, the first picture clearly numbers the control bits and matches them with the outputs. The first picture also shows the contents of the black box. After the action of the `X` gate the quantum state is $\{\texttt{X}(◯)◯, \texttt{X}(●)●\} = \{●◯, ◯●\}$ and that's what Bob receives. After the `C-NOT` this becomes:

$$\overrightarrow{\texttt{X}}(\{●◯, ◯●\}) = \{●●, ◯●\} = \{●, ◯\}●$$

After the Hadamard we have: $\texttt{H}(\{●, ◯\})● = ◯●$.

```
qc = QuantumCircuit(2)
qc.h(0)
qc.cx(0, 1)
qc.x(0)
qc.cx(0, 1)
qc.h(0)
qc.draw(output='mpl')
```




```
sv = Statevector.from_instruction(qc)
sv.draw('latex')
```

$|10\rangle$

The picture above is a reminder of how Qiskit produces the answer: backwards (as $q_1 q_0$, and we have explained why).

Here's a new CHALLENGE for you: what if the intended message is 10?

If the intended message is 10 (i.e., $q_0 = 1$ and $q_1 = 0$, which in Qiskit convention would be reported as 01) Alice needs to apply a Z gate: $\{\mathtt{Z}(\bigcirc)\bigcirc, \mathtt{Z}(\CIRCLE)\CIRCLE\} = \{\bigcirc\bigcirc, \overline{\CIRCLE}\CIRCLE\}$. Bob receives this. The effect of the C-NOT gate is:

$$\overrightarrow{\mathtt{X}}(\{\bigcirc\bigcirc, \overline{\CIRCLE}\CIRCLE\}) = \{\bigcirc\bigcirc, \overline{\CIRCLE}\bigcirc\} = \{\bigcirc, \overline{\CIRCLE}\}\bigcirc$$

The Hadamard gate makes this: $\mathtt{H}(\{\bigcirc, \overline{\CIRCLE}\})\bigcirc = \CIRCLE\bigcirc$.

Here's a new CHALLENGE for you: what if[2] the intended message is 11?

Superdense coding[3] and teleportation are dual phenomena. Teleportation can be described as entanglement-assisted quantum information transfer over a classical channel; superdense coding can be described as entanglement-assisted classical information transfer over a quantum channel. In both cases entanglement plays a crucial role. We have addressed the topic in general [19] and with respect to very specific phenomena (e.g., the GHZ game, [20]) in the context of the misty states formalism and the quantum abacus in other papers [21]. Quantum particles seem to influence each other with superluminal speed over arbitrarily long distances [18]. Quantum algorithms make use of this property. Entanglement swapping, another important protocol, allows particles that never interacted in the past to become entangled. In that sense entanglement swapping is a sort of teleportation of entanglement.

---

[2]If the intended message is 11 Alice needs to apply an X gate and then a Z gate. After the X gate we already calculated the state to be: $\{\CIRCLE\bigcirc, \bigcirc\CIRCLE\}$. The effect of the Z gate on this state is: $\{\mathtt{Z}(\CIRCLE)\bigcirc, \mathtt{Z}(\bigcirc)\CIRCLE\} = \{\overline{\CIRCLE}\bigcirc, \bigcirc\CIRCLE\}$ and that's what Bob receives. After the C-NOT the state becomes: $\overrightarrow{\mathtt{X}}(\{\overline{\CIRCLE}\bigcirc, \bigcirc\CIRCLE\}) = \{\overrightarrow{\mathtt{X}}(\overline{\CIRCLE}\bigcirc), \overrightarrow{\mathtt{X}}(\bigcirc\CIRCLE)\} = \{\overline{\CIRCLE}\CIRCLE, \bigcirc\CIRCLE\}$ Now this further becomes: $\{\overline{\CIRCLE}, \bigcirc\}\CIRCLE = \{\bigcirc, \overline{\CIRCLE}\}\CIRCLE$ and after the Hadamard gate we have $\mathtt{H}(\{\bigcirc, \overline{\CIRCLE}\})\CIRCLE = \CIRCLE\CIRCLE$ so all checks out as originally announced. This concludes our description of superdense coding.

[3]Also, please, lookup Holevo's theorem.

Chapter 11

Grover Search Algorithm

Grover's algorithm [30], [31] can speed up an unstructured search problem quadratically, but its uses extend beyond that; it can serve as a general trick or subroutine to obtain quadratic run time improvements for a variety of other algorithms. This is called the amplitude amplification trick.

Suppose you are given a large list of N items. Among these items there is one item with a unique property that we wish to locate; we will call this one the winner w. Think of each item in the list as a box of a particular color. Say all items in the list are gray except the winner w, which is purple. To find the purple box—the marked item—using classical computation, one would have to check on average $\frac{N}{2}$ of these boxes, and in the worst case, all of them.

On a quantum computer, however, we can find the marked item in roughly $\sqrt{N}$ steps[1] with Grover's amplitude amplification trick. Grover's algorithm consists of three main algorithmic steps: state preparation, the oracle, and the diffusion operator. The state preparation is where we create the search space, which is all possible cases the answer could take. In the list example we mentioned above, the search space would be all the items of that list. The oracle is what marks the correct answer, or answers we are looking for, and the diffusion operator magnifies these answers so they can stand out and be measured at the end of the algorithm.

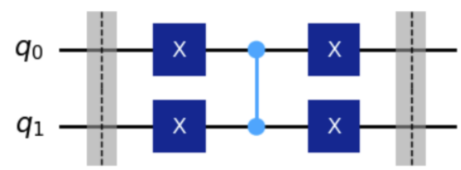
Here's a new CHALLENGE for you: you have four cards drawn from a deck of playing cards. One of the four cards is an ace, the other three are not. The cards are face down. How many[2] cards do you need to flip over, on average, to find the ace? Justify your answer.

[1] A quadratic speedup is indeed a substantial time-saver for finding marked items in long lists. Additionally, the algorithm does not use the list's internal structure, which makes it generic; this is why it immediately provides a quadratic quantum speed-up for many other classical problems.

[2] The answer is that, on average, $\frac{(6+3\cdot 18)}{4!} = 2.5$ cards need to be flipped.

The first step of Grover's algorithm is the initial state preparation. As we just mentioned, the search space is all possible values we need to search through to find the answer we want. Here our 'database' is comprised of all the possible computational basis states our qubits can be in. For example, if we have 2 qubits, our list is the states $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$ (i.e the states $|1\rangle$ to $|4\rangle$). So, in this case the size of our search space will be $N = 2^2 = 4$. For two qubits we can imagine there are four cards face-down on the table only one of which is an ace. To find the ace we would have to turn face up on average more than two cards. Grover's algorithm finds the card in just one iteration, always[3].

The second and most important step of Grover's algorithm is the oracle. Oracles add a negative phase to the solution states so they can standout from the rest and be measured[4]. For our two-qubit example we have four oracles. We show them below and start with the one for $|00\rangle$:




```
sv = Statevector.from_instruction(oracle_00)
sv.draw('latex')
```

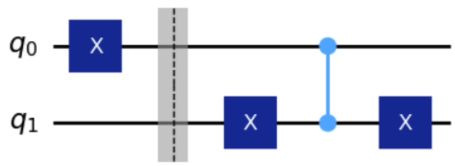
$-|00\rangle$

Here's a CHALLENGE for you: prove that this is the oracle for $|00\rangle$.

[3]Another game associated with the two qubit Grover's algorithm is 'money or tigers' by Ed Barnes from Virginia Tech: imagine you have four doors, and behind one of them there's a large sum of money, while behind each one of the other three there is a hungry tiger. Grover's algorithm shows you which door to open to get to the money in one iteration, thus eliminating the risk of running, in the process, into any of the tigers.

[4]What makes Grover's algorithm so powerful is how easy it is to convert a problem to an oracle of this form. There are many computational problems in which it is difficult to find a solution, but relatively easy to verify a solution.

We know that the controlled-Z (`C-Z`) gate adds a phase only when both qubits are ●. That should be enough to realize why this is indeed the oracle for ◯◯: any other state would fail to produce the needed prerequisite for an added phase. It is now easy to determine the other three oracles:

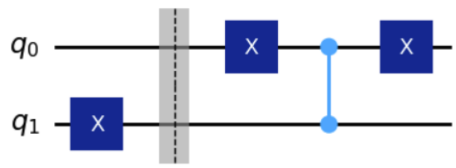



```
sv = Statevector.from_instruction(oracle_10)
sv.draw('latex')
```

$-|01\rangle$

First off we note again, here, that the name of the quantum oracle (circuit) is chosen with the traditional numbering convention of qubits in mind (i.e., q_0 first, then q_1) whereas the reporting is done using Qiskit numbering convention, that is, q_1 is listed first, then q_0. Then the state that is identified by the oracle, in Qiskit, as the solution, is consistent with ●◯.

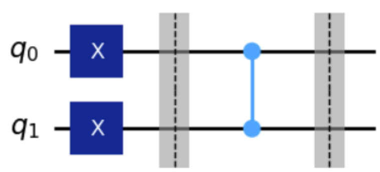
Here's a CHALLENGE for you: convince yourself that is the oracle for ●◯.




```
sv = Statevector.from_instruction(oracle_01)
sv.draw('latex')
```

$-|10\rangle$

Meanwhile the oracle for $\bigcirc\bullet = |01\rangle$ or, in Qiskit notation $|10\rangle$, is exactly symmetric. With input $\bigcirc\bullet$ to the oracle the state after the controlled-Z gate is $\texttt{C-Z}(\texttt{X}(\bigcirc)\bullet) = \overline{\bullet\bullet}$; all other inputs do not acquire a phase. Then what comes out of the oracle in this case is $\overline{\texttt{X}(\bullet)\bullet} = \overline{\bigcirc\bullet}$ and Qiskit is reporting that as $-|10\rangle$. All other inputs are reconstructed unchanged, and are not marked as solution(s). Finally, the simplest oracle is the one for $|11\rangle$:

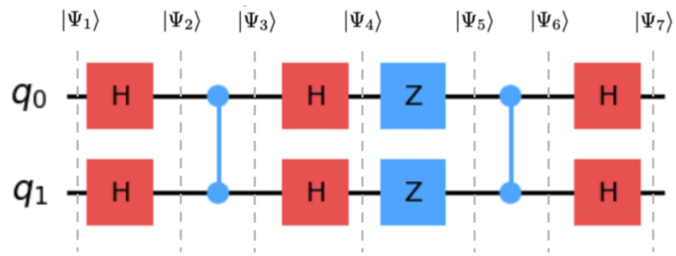



```
sv = Statevector.from_instruction(oracle_11)
sv.draw('latex')
```

$-|11\rangle$

CHALLENGE for you: convince yourself that this is the oracle for $|11\rangle$.

We will now present the circuit for Grover's search algorithm when the input consists of two qubits. We put the oracle for $|11\rangle$ in and remind the



reader that the oracle simply recognizes (or validates) the right answer—it does not attempt to construct it in any way. It is through the procedure called amplitude amplification that this quantum algorithm significantly enhances the probability of guessing the right answer w. This procedure stretches out (amplifies) the amplitude of the marked item, which shrinks the other items' amplitude, so that measuring the final state will return the right item with near-certainty. We will trace the algorithm step by step and naturally we start with $|\Psi_1\rangle =$ ◯◯ $= |00\rangle$.

CHALLENGE for you: please calculate $|\Psi_2\rangle$ and $|\Psi_3\rangle$.

Then,

$$
\begin{aligned}
|\Psi_2\rangle &= H(\bigcirc)H(\bigcirc) \\
&= \{\bigcirc, \bullet\}\{\bigcirc, \bullet\} \\
&= \{\bigcirc\bigcirc, \bullet\bigcirc, \bigcirc\bullet, \bullet\bullet\} \\
&= \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle
\end{aligned}
$$

Next stage is after the controlled-Z (`C-Z`) gate:

$$
\begin{aligned}
|\Psi_3\rangle &= \{\texttt{C-Z}(\bigcirc\bigcirc), \texttt{C-Z}(\bullet\bigcirc), \texttt{C-Z}(\bigcirc\bullet), \texttt{C-Z}(\bullet\bullet)\} \\
&= \{\bigcirc\bigcirc, \bullet\bigcirc, \bigcirc\bullet, \overline{\bullet\bullet}\} \\
&= \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle
\end{aligned}
$$

CHALLENGE for you: please calculate $|\Psi_4\rangle$.

And here comes a long (but instructive) calculation:

$$
\begin{aligned}
|\Psi_4\rangle &= \{\mathtt{H}(\bigcirc)\mathtt{H}(\bigcirc), \mathtt{H}(\bullet)\mathtt{H}(\bigcirc), \mathtt{H}(\bigcirc)\mathtt{H}(\bullet), \overline{\mathtt{H}(\bullet)\mathtt{H}(\bullet)}\} \\
&= \{\{\bigcirc, \bullet\}\{\bigcirc, \bullet\}, \\
&\quad \{\bigcirc, \overline{\bullet}\}\{\bigcirc, \bullet\}, \\
&\quad \{\bigcirc, \bullet\}\{\bigcirc, \overline{\bullet}\}, \\
&\quad \overline{\{\bigcirc, \overline{\bullet}\}}\{\bigcirc, \overline{\bullet}\}\} \\
&= \{\ \bigcirc\bigcirc, \bigcirc\bullet, \bullet\bigcirc, \bullet\bullet, \\
&\quad \bigcirc\bigcirc, \bigcirc\bullet, \overline{\bullet}\bigcirc, \overline{\bullet}\bullet, \\
&\quad \bigcirc\bigcirc, \bigcirc\overline{\bullet}, \bullet\bigcirc, \bullet\overline{\bullet}, \\
&\quad \{\overline{\bigcirc}, \bullet\}\{\bigcirc, \overline{\bullet}\}\} \\
&= \{\ \bigcirc\bigcirc, \bigcirc\bullet, \\
&\quad \bigcirc\bigcirc, \\
&\quad \bigcirc\bigcirc, \qquad \bullet\bigcirc, \bullet\overline{\bullet}, \\
&\quad \overline{\bigcirc}\bigcirc, \overline{\bigcirc\bullet}, \bullet\bigcirc, \bullet\overline{\bullet}\} \\
&= \{\ \bigcirc\bigcirc, \bigcirc\bullet, \bullet\bigcirc, \bullet\overline{\bullet}, \\
&\quad \bigcirc\bigcirc, \bigcirc\bullet, \bullet\bigcirc, \bullet\overline{\bullet}\} \\
&= \{\ \bigcirc\bigcirc, \bigcirc\bullet, \bullet\bigcirc, \overline{\bullet\bullet}\} \\
&= \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle
\end{aligned}
$$

As one can see some pairs of terms cancel[5] each other. I have removed them (please note that the canceling pairs are conveniently placed one term above the other) but also kept their place in the original equation for easier tracking.

New CHALLENGE for you: please calculate $|\Psi_5\rangle$ and $|\Psi_6\rangle$.

[5]For the last simplification we recall the calculation of probability from the first chapter as follows: if $\bigcirc$ appears n times in the misty state and $\bullet$ appears m times in the misty state then their probabilities are $\frac{n^2}{n^2+m^2}$ and $\frac{m^2}{n^2+m^2}$ respectively. It follows that if states occur an equal multiple of times (e.g., nk and mk with $k \in \mathbb{N}$) then the probabilities are unchanged (because k^2 appears everywhere and consequently it simplifies).

Now $|\Psi_5\rangle$ reflects the action of the Z gates on $|\Psi_4\rangle$:

$$\begin{aligned}
|\Psi_5\rangle &= \{\mathtt{Z}(\bigcirc)\mathtt{Z}(\bigcirc), \mathtt{Z}(\bullet)\mathtt{Z}(\bigcirc), \mathtt{Z}(\bigcirc)\mathtt{Z}(\bullet), \overline{\mathtt{Z}(\bullet)\mathtt{Z}(\bullet)}\} \\
&= \{\bigcirc\bigcirc, \overline{\bullet\bigcirc}, \overline{\bigcirc\bullet}, \overline{\overline{\bullet}\,\overline{\bullet}}\} \\
&= \{\bigcirc\bigcirc, \overline{\bullet\bigcirc}, \overline{\bigcirc\bullet}, \overline{\bullet\bullet}\} \\
&= \frac{1}{2}|00\rangle - \frac{1}{2}|01\rangle - \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle
\end{aligned}$$

The calculation of $|\Psi_6\rangle$ is similar to the one for $|\Psi_3\rangle$:

$$\begin{aligned}
|\Psi_6\rangle &= \{\mathtt{C\text{-}Z}(\bigcirc\bigcirc), \overline{\mathtt{C\text{-}Z}(\bullet\bigcirc)}, \overline{\mathtt{C\text{-}Z}(\bigcirc\bullet)}, \overline{\mathtt{C\text{-}Z}(\bullet\bullet)}\} \\
&= \{\bigcirc\bigcirc, \overline{\bullet\bigcirc}, \overline{\bigcirc\bullet}, \overline{\overline{\bullet\bullet}}\} \\
&= \{\bigcirc\bigcirc, \overline{\bullet\bigcirc}, \overline{\bigcirc\bullet}, \bullet\bullet\} \\
&= \frac{1}{2}|00\rangle - \frac{1}{2}|01\rangle - \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle
\end{aligned}$$

New CHALLENGE for you: please calculate $|\Psi_7\rangle$.

The calculation for $|\Psi_7\rangle$ also matches the type of steps we have seen at $|\Psi_4\rangle$ just that the result is, convincingly, different:

$$
\begin{aligned}
|\Psi_7\rangle &= \{\mathtt{H}(\bigcirc)\mathtt{H}(\bigcirc), \overline{\mathtt{H}(\bullet)\mathtt{H}(\bigcirc)}, \overline{\mathtt{H}(\bigcirc)\mathtt{H}(\bullet)}, \mathtt{H}(\bullet)\mathtt{H}(\bullet)\} \\
&= \{\{\bigcirc, \bullet\}\{\bigcirc, \bullet\}, \\
&\quad \{\bigcirc, \overline{\bullet}\}\overline{\{\bigcirc, \bullet\}}, \\
&\quad \overline{\{\bigcirc, \bullet\}}\{\bigcirc, \overline{\bullet}\}, \\
&\quad \{\bigcirc, \overline{\bullet}\}\{\bigcirc, \overline{\bullet}\}\} \\
&= \{ \bigcirc\bigcirc, \bigcirc\bullet, \bullet\bigcirc, \bullet\bullet, \\
&\quad \{\bigcirc, \overline{\bullet}\}\{\overline{\bigcirc}, \overline{\bullet}\}, \\
&\quad \{\overline{\bigcirc}, \overline{\bullet}\}\{\bigcirc, \overline{\bullet}\}, \\
&\quad \bigcirc\bigcirc, \bigcirc\overline{\bullet}, \overline{\bullet}\,\bigcirc, \overline{\bullet}\,\overline{\bullet}\} \\
&= \{ \bigcirc\bigcirc, \qquad\qquad \bullet\bullet, \\
&\quad \bigcirc\overline{\bigcirc}, \bigcirc\overline{\bullet}, \overline{\bullet}\,\overline{\bigcirc}, \overline{\bullet}\,\overline{\bullet}, \\
&\quad \overline{\bigcirc}\bigcirc, \overline{\bigcirc}\,\bullet, \overline{\bullet}\bigcirc, \overline{\bullet}\,\overline{\bullet}, \\
&\quad \bigcirc\bigcirc, \qquad\qquad \bullet\bullet\} \\
&= \{ \qquad\qquad\qquad \bullet\bullet, \\
&\quad \qquad\qquad\qquad \bullet\bullet, \\
&\quad \qquad\qquad\qquad \bullet\bullet, \\
&\quad \qquad\qquad\qquad \bullet\bullet\} \\
&= \{\bullet\bullet\} = |11\rangle
\end{aligned}
$$

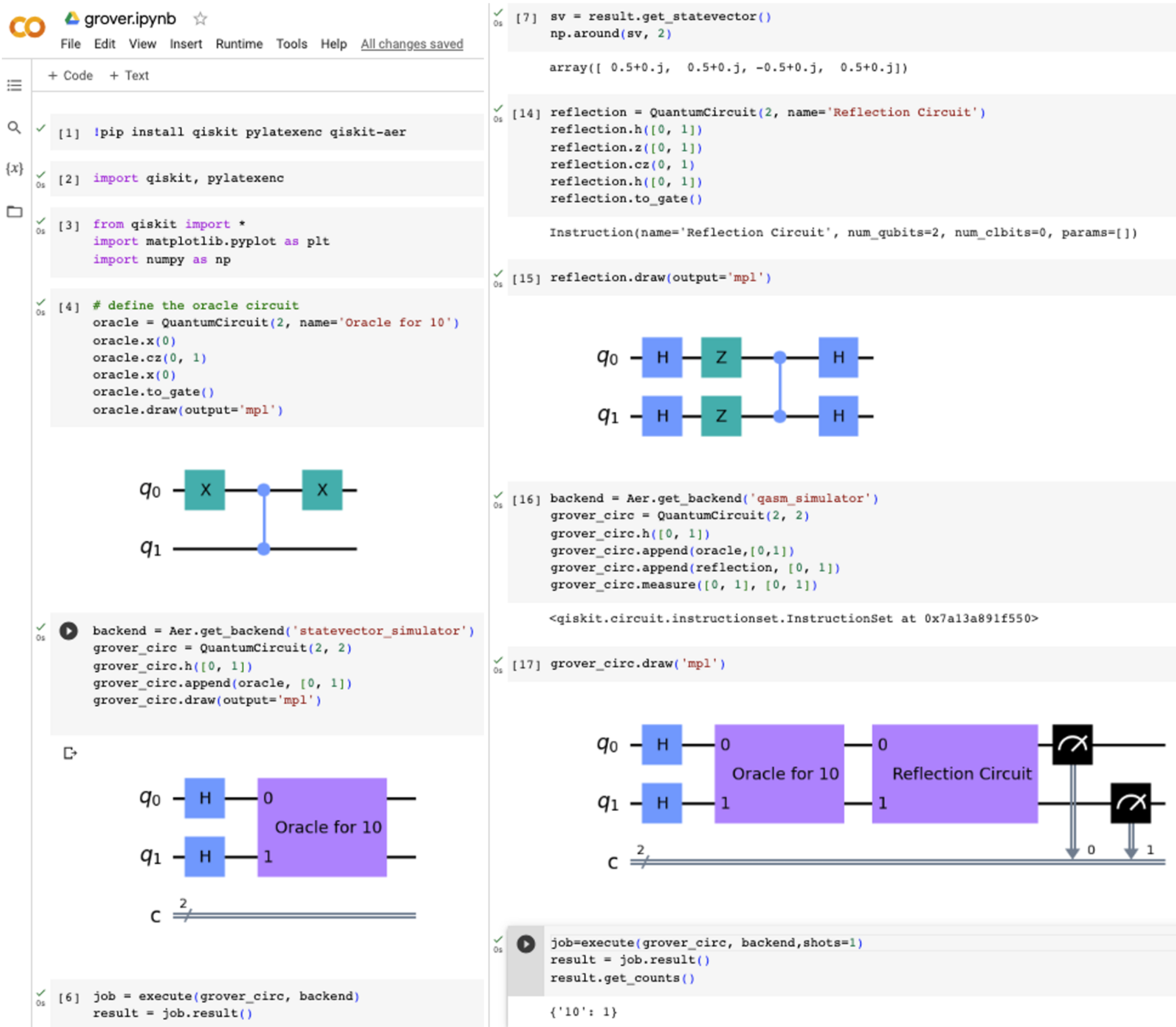
This time the pairs of items that cancel each other are not next to each other but they're still in the same vertical column.

Thus, Grover's algorithm produces the answer in one step[6]. Using a quantum oracle that is able to identify (not construct) the correct answer we know which of the four cards face down on the table is the w card[7] in just one step.

HOMEWORK assignment: trace Grover for each of the other three cases.

[6] Calculations for the other three cases proceed in a similar manner.

[7] Or, in the Eddie Barnes game, which door to open to get to the money, in just one step, avoiding altogether the other three doors that lead to a hungry tiger.

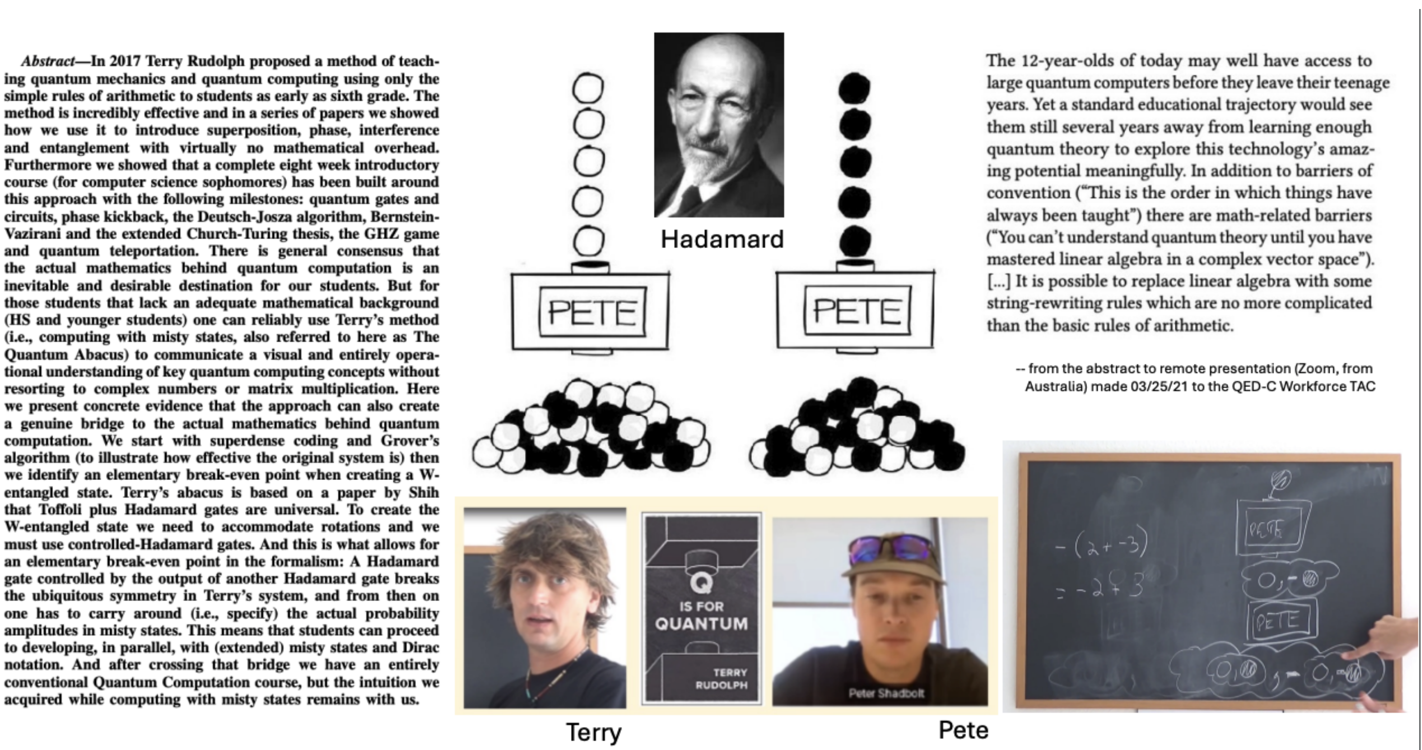


Quick CHALLENGE for you: what is this picture trying to convey?

Chapter 12

Breaking Even

We again resort to a slide:



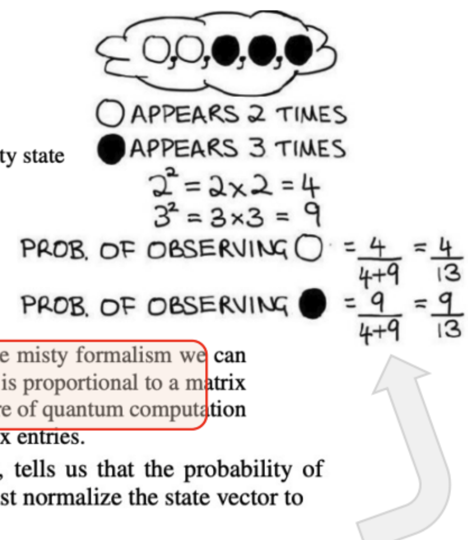
Its purpose it to reset the conversation and to (re-)introduce the main characters. In this chapters we explain what the misty state formalism is and how it could be made to meet the conventional algebra normally used with Dirac notation, for the benefit of the student(s).

It should be clear that the formalism proposed by Terry Rudolph is deliberately very simple, so it can be accessible to middle school students. Here we make the claim that it is possible to gradually increase the expressive power of the formalism to the point where it meets conventional math.

Some of the things we will emphasize below are from the book's website[1]. Of significant relevance are (a) the note on recovering the standard quantum mathematical formalism and (b) from the dialogue with John Horgan in the FAQ[2] the part(s) about "the genesis of the whole misty-method" and how to deal with imaginary and complex numbers in the mist. (These parts are close together. Everything is interesting on the associated website, though, so we encourage you take a close look to what's available there.)

Converting between the misty and regular quantum formalisms

Terry Rudolph



These are some *very* brief notes to help a student learning regular quantum theory to connect the misty state formalism with the standard Hilbert space approach, as there are a couple of subtleties.

Unitary evolution in the quantum formalism just gives the "boxes". For example the PETE box is just the Hadamard single qubit (ball) gate

$$H = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} /\sqrt{2}.$$

There is one subtlety to be careful about in going between the mist and the quantum state - in the misty formalism we can only incorporate boxes whose representation in the quantum formalism is via a unitary matrix which is proportional to a matrix of integer entries. For example $H \times \sqrt{2}$ contains only integers. It is a remarkable and nontrivial feature of quantum computation that such unitaries can be universal, ie used to simulate all unitaries, even ones with irrational complex entries.

I avoided normalizing vectors by introducing the "squaring rule" which, in this example, tells us that the probability of observing the ball to be white is $2^2/(2^2+1^2) = 4/5$. In standard quantum theory we would just normalize the state vector to

$$\begin{bmatrix} 2 \\ -1 \end{bmatrix} /\sqrt{5},$$

Here $\{\bigcirc, \bigcirc, \bullet, \bullet, \bullet\} = \frac{2}{\sqrt{13}}|0\rangle + \frac{3}{\sqrt{13}}|1\rangle$, clearly.
A misty state is just an un-normalized quantum state.

and the probabilities are then the squares of the vector entries.

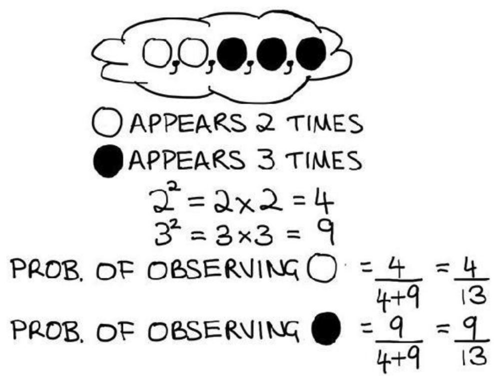
The misty state formalism is just a different way of representing a subset of the quantum formalism which is actually universal for quantum computing (a fact first proven by Shih, references are at the end of the book). So in fact all quantum calculations could be done in this formalism to arbitrary accuracy. You may wonder where the complex amplitudes you get taught about are - they can be dealt with by having an extra ball in the mist. Effectively configurations with this extra ball white are "real parts of the quantum amplitudes" and those with the extra ball white are "imaginary parts of the quantum amplitudes".

With this we're going to start describing how we break even. And we start by reminding ourselves that a classical bit is much like the side of a coin that sits on the table: it is either head or tails (i.e., 0 or 1) no matter how many times you look at it. A quantum bit, on the other hand, is more like a coin that's spinning on the table: the only thing you can hope to know is a (or, the) set of possible outcomes and their associated probabilities.

[1] `https://www.qisforquantum.org/supplementary-material`
[2] `https://www.qisforquantum.org/faqs`

A qubit is written as $\alpha|0\rangle + \beta|1\rangle$ with $\alpha, \beta \in \mathbb{C}$ and such that $|\alpha|^2 + |\beta|^2 = 1$. This linear superposition specifies the expected outcomes (i.e., $|0\rangle$ and $|1\rangle$) the probability amplitudes (α and β) and states that the two probabilities should add up to 1.

In "Q is for Quantum" ([13], [32]) Terry Rudolph "teaches [quantum mechanics] to an audience presumed only to know basic arithmetic." The book uses a simple (but very effective) string-rewriting technique that starts $|0\rangle =$ ◯ and $|1\rangle =$ ⬤ and represents[3] superposition as "misty states":



Here {◯, ◯, ⬤, ⬤, ⬤} $= \frac{2}{\sqrt{13}}|0\rangle + \frac{3}{\sqrt{13}}|1\rangle$, clearly.

A misty state is just an un-normalized quantum state. Note that a state like {◯, ◯, ⬤, ⬤, ⬤} is not written in the book as {2 × ◯, 3 × ⬤} or {2 · ◯, 3 · ⬤} or even {2◯, 3⬤} by choice. We refer to the misty state formalism from [13] as the "Quantum Abacus". The quantum abacus is simply a different way of representing a subset of the quantum formalism which is actually universal for quantum computing, a fact first proven by Shih [33]. Syntactically, within the abacus, misty states are shown always only in white and black and no numbers are used (as coefficients) anywhere in the formalism. Since misty states always contain a whole number of black and white balls, it follows that $\frac{1}{\sqrt{3}}|0\rangle + \frac{\sqrt{2}}{\sqrt{3}}|1\rangle$ does not admit[4] a representation as a misty state, something that will prove very important shortly.

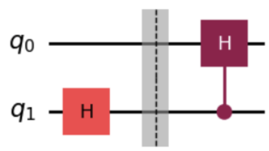
Please also see the comment in the previous slide that says that only a certain kind of boxes can be incorporated in the misty formalism. The book uses mostly Hadamard (`PETE`) boxes as well as controlled-`NOT`'s and so a student could easily experiment with Qiskit and come up with various circuits and try to apply the misty state formalism to them. Students are very inquisitive, in fact, in a little bit we're going to assume that they might find (on their own) and start watching John Watrous' series of lectures[5] on Quantum Information

[3] This diagram appears on p. 83 in the book.
[4] Because no two perfect squares in **N** add up to 3
[5] `https://www.youtube.com/playlist?list=PLOFEBzvs-VvqKKMXX4vbi4EB1uaErFMSO`

Science on the Qiskit channel on YouTube—and we will see what interesting and unexpected consequences that might have[6] with respect to our efforts here.

Here's an example constructed with just Hadamard gates:

```
be = QuantumCircuit(2)
be.h(1)
be.barrier()
be.ch(1, 0)
be.draw(output='mpl')
```




```
sv = Statevector.from_instruction(be)
sv.draw('latex')
```

$\frac{\sqrt{2}}{2}|00\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle$

Let's see if we can calculate the same with our abacus.

Quick CHALLENGE for you: can you lead us in the calculation?

We have to appreciate the fact that so far there have been no numbers (coefficients, for probability amplitudes) in the misty states formalism[7].

The initial state is $q_0 q_1 = $ ◯◯. After the first Hadamard gate it becomes $|\Psi_1\rangle = $ ◯{◯, ●}. That's the state at the barrier. We then have to calculate the effect of $\overleftarrow{\texttt{H}}$ (◯{◯, ●}) = { $\overleftarrow{\texttt{H}}$ (◯◯), $\overleftarrow{\texttt{H}}$ (◯●)} so we write:

$|\Psi_2\rangle$ = {◯◯, {◯, ●}●}
= {◯◯, {◯●, ●●}}
= {◯◯, ◯●, ●●}

$$= \frac{1}{\sqrt{3}}|00\rangle + \frac{1}{\sqrt{3}}|10\rangle + \frac{1}{\sqrt{3}}|11\rangle$$

New CHALLENGE for you: is the above calculation correct?

[6] https://www.youtube.com/watch?v=DfZZS8Spe7U&t=2845s

[7] That's what makes it accessible to students as early as middle school.

We switched to Qiskit ordering at the end but it's clear that the calculation is not accurate: the probability amplitudes don't match. Let's do a little research:

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \end{pmatrix} = \{\bigcirc\bigcirc, \bigcirc\bullet\}$$

This is the effect of the first Hadamard gate acting on q_1 when on the other wire we have the identity gate (the combined matrix is their tensor product). Now we have to use the matrix representation of the controlled Hadamard gate:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{2} \\ 0 \\ \frac{1}{2} \end{pmatrix}$$

Now the resulting vector is equivalent to:

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|11\rangle$$

In Qiskit ordering this is:

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle$$

So now we have the answer: as Terry warned us, in the MSF "we can only [use] boxes whose representation in the quantum formalism is via a unitary matrix which is proportional to a matrix of integer entries." Clearly this is not true of the controlled Hadamard matrix and that's the reason for which our calculations fail. What can we do?

Quick CHALLENGE for you: propose a course of action.

This is the point where the MSF and the conventional formalism need to break even. We propose we extend the MSF by allowing coefficients representing the probability amplitudes. This will bring us closer to the Dirac algebraic notation but at this point we have so much that we have been able to understand with just the pure MSF. The upgrade does not feel gratuitous, in fact it seems to be earned. Here's how the calculation proceeds now: $|\Psi_1\rangle = ◯\{\frac{1}{\sqrt{2}}◯, \frac{1}{\sqrt{2}}●\}$ and then calculate $\overleftarrow{\mathtt{H}}(◯\{\frac{1}{\sqrt{2}}◯, \frac{1}{\sqrt{2}}●\})$ like we did before. The difference is that now we have the probability amplitudes with us. So we have the following sequence of steps:

$$
\begin{aligned}
\overleftarrow{\mathtt{H}}(◯\mathtt{H}(◯)) &= \overleftarrow{\mathtt{H}}(◯\{\frac{1}{\sqrt{2}}◯, \frac{1}{\sqrt{2}}●\}) \\
&= \{\frac{1}{\sqrt{2}}\overleftarrow{\mathtt{H}}(◯◯), \frac{1}{\sqrt{2}}\overleftarrow{\mathtt{H}}(◯●)\} \\
&= \{\frac{1}{\sqrt{2}}◯◯, \frac{1}{\sqrt{2}}\{\frac{1}{\sqrt{2}}◯, \frac{1}{\sqrt{2}}●\}●\} \\
&= \{\frac{1}{\sqrt{2}}◯◯, \frac{1}{2}◯●, \frac{1}{2}●●\} \\
&= \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle
\end{aligned}
$$

On the last line above we switched to Qiskit ordering.

New CHALLENGE for you: look up the `S` gate (and its associated matrix). Using Qiskit, matrix multiplication, or whatever method you prefer try to answer this question: Is it true that $\mathtt{S}^2 = \mathtt{Z}$? If so, what is the $\sqrt{\mathtt{Z}}$?

12.1 $\sqrt{\mathtt{NOT}}$ and $\sqrt{\mathtt{Z}}$

Now that we have extended the MSF with coefficients we can introduce (and prove) $\mathtt{S} = \sqrt{\mathtt{Z}}$ and $\sqrt{\mathtt{X}} = \mathtt{HSH}$. First we have $\mathtt{S}(\bigcirc) = \bigcirc$ and $\mathtt{S}(\bullet) = i\bullet$ where $i = \sqrt{-1}$. From this it's clear that $\mathtt{S}^2 = \mathtt{Z}$ so $\mathtt{S} = \sqrt{\mathtt{Z}}$ because $i^2 = -1$.

Likewise $\mathtt{HSH} \cdot \mathtt{HSH} = \mathtt{HS}^2\mathtt{H} = \mathtt{HZH} = \mathtt{X}$ which we proved early in this paper so $\mathtt{HSH} = \sqrt{\mathtt{X}} = \sqrt{\mathtt{NOT}}$ checks out.

Quick CHALLENGE for you: why is $\mathtt{HSH} \cdot \mathtt{HSH} = \mathtt{HS} \cdot \mathtt{SH}$. What is $\mathtt{H}^2$. How is that relevant here? What do we call that property, in general?

These are also great opportunities to introduce students to matrices and properties of matrix multiplication as well as the notion of inverse and/or unitary matrix. When we can derive a result in more than one way we feel more confident about its correctness.

New kind of CHALLENGE for you:

Calculate $\mathtt{H}(\{\alpha\bigcirc, \beta\bullet\})$.

After this you can go back to page 16 and solve that exercise.

Chapter 13

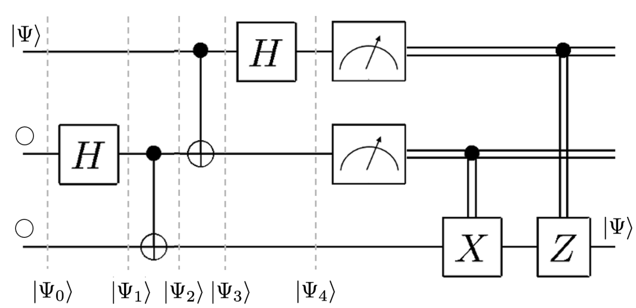
Teleportation

We now have all the tools to address teleportation (inaccessible in the regular, original misty state formalism (MSF)). 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 step is going to be to find a way to represent an arbitrary $|\Psi\rangle$ state in our "abacus" system. But following our argument thus far this is no longer a challenge (since we are now using the extended MSF). That will allow us to morph gradually into the traditional, mathematical representation.

In that case $\forall \alpha, \beta \in \mathbb{C}$ we may also have[1]:

$$\alpha|0\rangle + \beta|1\rangle = \{\alpha \bigcirc, \beta \bullet\}$$

Quantum teleportation ([16], [22]) 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 instan-

[1]This is the extension to the misty state(s) formalism (MSF).

taneously[2] and is not affected by distance or separating barriers. Let's prove the protocol by calculating intermediary stages $|\Psi_0\rangle, \ldots, |\Psi_4\rangle$. We start with:

$$|\Psi_0\rangle = \Big\{\alpha \bigcirc\bigcirc\bigcirc, \beta \bullet\bigcirc\bigcirc\Big\}$$

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:

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

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:

$$\begin{aligned} \mathtt{H}(\{\alpha \bigcirc, \beta \bullet\}) =& \alpha \{\bigcirc, \bullet\} + \beta \{\bigcirc, \overline{\bullet}\} = \\ =& \{(\alpha + \beta)\bigcirc, (\alpha - \beta)\bullet\} \end{aligned}$$

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:

$$|\Psi_1\rangle = \Big\{\alpha \bigcirc\bigcirc\bigcirc, \alpha \bigcirc\bullet\bigcirc, \beta \bullet\bigcirc\bigcirc, \beta \bullet\bullet\bigcirc\Big\}$$

Traditional calculation, again, confirms our result:

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

Quick CHALLENGE for you: calculate $|\Psi_2\rangle$, $|\Psi_3\rangle$, $|\Psi_4\rangle$.

[2] 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.

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):

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

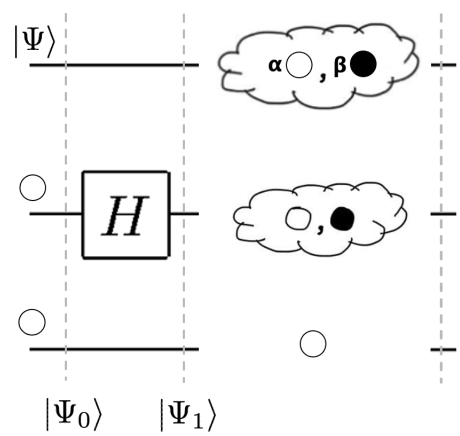
Quick CHALLENGE for you: what does this all mean?

ANSWER In this form it is visible what gates have to be applied[3] to the last qubit to make it the input teleported state $\alpha|0\rangle + \beta|1\rangle$.

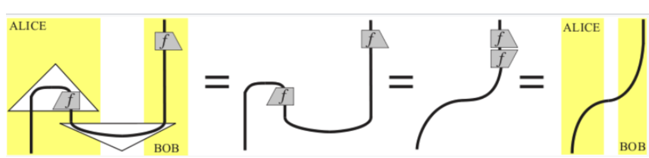
Let's now revisit, as we promised, the topic of tensorial product in the context of our derivation. Please finish this calculation:

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

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 [12] 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.

[3]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).

Chapter 14

Arbitrary Rotations

It's time to introduce another gate that does not have a representation in the MSF (but readily has one in the extended MSF). At QSEEC 2023 in Seattle we were asked how we define arbitrary rotations in the MSF.

The answer is: we define them as primitives in the extended MSF. We were also asked how we define arbitrary qubits, but by now we have already answered that question[1]. So let's consider a specific rotation gate that will be useful a bit later. The first axiom is:

$$R_y(\theta_3)(\bigcirc) = \{\frac{1}{\sqrt{3}}\bigcirc, \sqrt{\frac{2}{3}}\CIRCLE\}$$

This is precisely the quantum state that we said, in the beginning of the paper, that it did not have a representation in the MSF. The other axiom is:

$$R_y(\theta_3)(\CIRCLE) = \{-\frac{2}{\sqrt{3}}\bigcirc, \sqrt{\frac{1}{3}}\CIRCLE\}$$

From this we can already calculate in general how this gate acts on a generic superposition of $|0\rangle = \bigcirc$ and $|1\rangle = \CIRCLE$. The reason this gate does not exist in the MSF will become clear below when we look at its matrix.

First off $\theta_3 = 2\arccos\frac{1}{\sqrt{3}}$ and so the matrix is:

$$\begin{pmatrix} \cos\frac{\theta_3}{2} & -\sin\frac{\theta_3}{2} \\ \sin\frac{\theta_3}{2} & \cos\frac{\theta_3}{2} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{3}} & -\sqrt{\frac{2}{3}} \\ \sqrt{\frac{2}{3}} & \frac{1}{\sqrt{3}} \end{pmatrix}$$

[1] e.g., $\texttt{X}(\{\alpha\bigcirc, \beta\CIRCLE\}) = \{\beta\bigcirc, \alpha\CIRCLE\}\ \forall\alpha, \beta \in \mathbb{C}$

Quick CHALLENGE for you: assume $R = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is a valid unitary matrix (perhaps it even represents a rotation). What is $R(\bigcirc)$ and what is $R(\CIRCLE)$?

Chapter 15

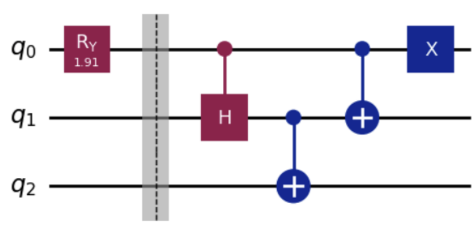
W-Entangled States

We acknowledge two sources here: one is the John Watrous video we referenced in the footnote on page 52, the other one is an online Wolfram Alpha reference[1].

We can now create W-entangled states:

```
import numpy as np
import math
theta = 2 * np.arccos(1/math.sqrt(3))
```

```
w = QuantumCircuit(3)
w.ry(theta, 0)
w.draw(output='mpl')
w.barrier()
w.ch(0, 1)
w.cx(1, 2)
w.cx(0, 1)
w.x(0)
w.draw('mpl')
```




```
sv = Statevector.from_instruction(w)
sv.draw('latex')
```

$$\frac{\sqrt{3}}{3}|001\rangle + \frac{\sqrt{3}}{3}|010\rangle + \frac{\sqrt{3}}{3}|100\rangle$$

Quick CHALLENGE for you: what is the state after the rotation?

[1]https://demonstrations.wolfram.com/ThreeQubitWStatesOnAQuantumComputer/

Let's calculate: the initial state is still ◯◯◯.

After the rotation we have $\{\frac{1}{\sqrt{3}}◯, \sqrt{\frac{2}{3}}●\}◯◯ = \{\frac{1}{\sqrt{3}}◯◯◯, \sqrt{\frac{2}{3}}●◯◯\}$.

New CHALLENGE for you: what's the state after the controlled-Hadamard gate?

When the controlled Hadamard kicks in we have:

$$
\begin{aligned}
&\{\frac{1}{\sqrt{3}}\overrightarrow{\texttt{H}}(\bigcirc\bigcirc)\bigcirc, \sqrt{\frac{2}{3}}\overrightarrow{\texttt{H}}(\text{●}\bigcirc)\bigcirc\} = \\
&\{\frac{1}{\sqrt{3}}\bigcirc\bigcirc\bigcirc, \sqrt{\frac{2}{3}}\text{●}\{\frac{1}{\sqrt{2}}\bigcirc, \frac{1}{\sqrt{2}}\text{●}\}\bigcirc\} = \\
&\{\frac{1}{\sqrt{3}}\bigcirc\bigcirc\bigcirc, \frac{1}{\sqrt{3}}\text{●}\bigcirc\bigcirc, \frac{1}{\sqrt{3}}\text{●}\text{●}\bigcirc\} = \\
&\frac{1}{\sqrt{3}}|000\rangle + \frac{1}{\sqrt{3}}|001\rangle + \frac{1}{\sqrt{3}}|011\rangle
\end{aligned}
$$

In the last line we switched to Qiskit ordering of qubits.

New CHALLENGE for you: what is the state after the first `C-NOT`?

After the first `C-NOT` we have:

$$
\begin{aligned}
&\{\frac{1}{\sqrt{3}}\bigcirc\overrightarrow{\mathtt{X}}(\bigcirc\bigcirc), \frac{1}{\sqrt{3}}\CIRCLE\overrightarrow{\mathtt{X}}(\bigcirc\bigcirc), \frac{1}{\sqrt{3}}\CIRCLE\overrightarrow{\mathtt{X}}(\CIRCLE\bigcirc)\} = \\
&\{\frac{1}{\sqrt{3}}\bigcirc\bigcirc\bigcirc, \frac{1}{\sqrt{3}}\CIRCLE\bigcirc\bigcirc, \frac{1}{\sqrt{3}}\CIRCLE\CIRCLE\CIRCLE\} = \\
&\frac{1}{\sqrt{3}}|000\rangle + \frac{1}{\sqrt{3}}|001\rangle + \frac{1}{\sqrt{3}}|111\rangle
\end{aligned}
$$

Again, we switched to Qiskit ordering at the very end.

New CHALLENGE for you: what is the state after the second `C-NOT`?

The calculation after the second `C-NOT` proceeds similarly:

$$
\begin{aligned}
&\{\frac{1}{\sqrt{3}}\overrightarrow{\texttt{X}}(\bigcirc\bigcirc)\bigcirc, \frac{1}{\sqrt{3}}\overrightarrow{\texttt{X}}(\bullet\bigcirc)\bigcirc, \frac{1}{\sqrt{3}}\overrightarrow{\texttt{X}}(\bullet\bullet)\bullet\} = \\
&\{\frac{1}{\sqrt{3}}\bigcirc\bigcirc\bigcirc, \frac{1}{\sqrt{3}}\bullet\bullet\bigcirc, \frac{1}{\sqrt{3}}\bullet\bigcirc\bullet\} = \\
&\frac{1}{\sqrt{3}}|000\rangle + \frac{1}{\sqrt{3}}|011\rangle + \frac{1}{\sqrt{3}}|101\rangle
\end{aligned}
$$

New CHALLENGE for you: what is the state at the end?

Finally after the `X` gate we have:

$$\{\frac{1}{\sqrt{3}}\text{●○○}, \frac{1}{\sqrt{3}}\text{○●○}, \frac{1}{\sqrt{3}}\text{○○●}\} =$$
$$\frac{1}{\sqrt{3}}|001\rangle + \frac{1}{\sqrt{3}}|010\rangle + \frac{1}{\sqrt{3}}|100\rangle$$

As before the Dirac notation is with Qiskit ordering.

Everything checks out.

Chapter 16

The GHZ Game

In this chapter we extend (slightly) the formalism from "Q is for Quantum" to discuss the GHZ game using misty states. Both the CHSH game and the game discussed in Part II (Q-Entanglement) of the book require a large number of repetitions and two players. If players can share Bell states they win with a probability (0.85) that is just above a certain (important) threshold: any local strategy can only win this game 75% of the time. The GHZ game does require three players (instead of two, and an entangled quantum state for three qubits) but in the GHZ game there is a strategy for the players to win every single time, whereas without entanglement the best they can do is to win three out of four games at most (so, probability 0.75).

Introduction

We assume the reader is familiar with the first two parts of "Q is for Quantum" by Terry Rudolph [13]. The Greenberger–Horne–Zeilinger (GHZ) state is a tripartite entangled quantum state (so, three entangled qubits) with the following expression: $\frac{|000\rangle+|111\rangle}{\sqrt{2}}$ (so, $\{\bigcirc\bigcirc\bigcirc, \bullet\bullet\bullet\}$). The first circuit on page 20 (taken from [19]) implements a GHZ state and is annotated with misty states to show why (and how) it works. Part II of the book [13] describes a game (due to Hardy [15]) that exhibits a special case of a version of quantum nonlocality. It is more common when discussing nonlocal quantum correlations to consider a game due to Clauser, Horne, Shimony and Holt (CHSH). The Greenberger–Horne–Zeilinger (GHZ) game is another interesting example of quantum pseudo-telepathy. Classically, the GHZ game has 75% winning probability. However, with a quantum strategy, the players will always win with a winning probability that equals 1 (one).

The Game

Following [14] we will summarize the GHZ game as follows: there are three players, Alice, Bob and Carol playing against a referee. The referee draws a triplet (x, y, z) randomly from the four options shown in the table below and listed here for your convenience: $(0,0,0),(1,1,0),(1,0,1)$ and $(0,1,1)$. Alice, Bob and Carol each respond with an answer a, b and c also in the form of 0 or 1. The players can formulate a strategy prior to the start of the game. However, no communication is allowed during the game itself.

Winning condition of GHZ game

x	y	z	$a+b+c$
0	0	0	0 mod 2
1	1	0	1 mod 2
1	0	1	1 mod 2
0	1	1	1 mod 2

Players win if the sum of their answers $a+b+c$ is even when $x = y = z = 0$ and odd in the other three cases. It can be shown [14] that there is no classical strategy that satisfies all four winning conditions simultaneously.

Classically, the best winning strategy is for Alice, Bob and Carol to always produce an odd sum (e.g., Alice always outputs a 1 while Bob and Carol always output 0). With such a strategy the players win 75% of the time.

However, if they are allowed to share a tripartite entangled state (known as the GHZ state) $|\psi\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$ they can win all the time (i.e., with probability 1). The strategy they need to adopt is as follows: any player that receives a 0 must make a measurement in the X basis; any player receiving a 1 must make a measurement in the Y basis. If the measurement comes out a $|+\rangle$ or a $|+i\rangle$ the player responds with a 0. Otherwise the player outputs a 1.

The X basis is made of the eigenstates of the Pauli X operator:

$$|+\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big) \text{ and } |-\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big)$$

The Y basis is made of the eigenstates of the Pauli Y operator:

$$|+i\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle + i|1\rangle\big) \text{ and } |-i\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle - i|1\rangle\big)$$

The next section shows the calculation. In the spirit of [13] we assume the reader sees i here for the first time. In the spirit of [19] we provide, here and there, milestones expressed in the traditional algebraic formalism and equivalents of the calculations done using misty states, for reinforcement and as a reality check.

16.1 The Calculations

We follow, by and large, the approach from Part II in "Q is for Quantum": we take the GHZ state that the three players share and show what the measurements associated with the quantum strategy turn it into. We then verify that players win in all combinations of outputs states from the resulting "misty state". We start with some notation. The computational (bit) basis consists of: $|0\rangle$ and $|1\rangle$. We write:

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

A qubit is a linear combination of the basis vectors so:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \text{ with } \alpha, \beta \in \mathbb{C} \text{ and } |\alpha|^2 + |\beta|^2 = 1$$

We choose to represent a misty state using curly brace(s) as the superposition operator:

$$|\psi\rangle = \{\alpha\bigcirc, \beta\bullet\}$$

We can then write:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big) = |+\rangle = \{\bigcirc, \bullet\} \tag{16.1}$$

Likewise we have

$$|\Phi^-\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big) = |-\rangle = \{\bigcirc, \overline{\bullet}\} \tag{16.2}$$

Here the overline denotes the phase.

Here's a statement:

$$|0\rangle = \frac{1}{\sqrt{2}}\big(|+\rangle + |-\rangle\big) \tag{16.3}$$

Is it true? And if so, how do we represent this as a misty state?

We need new notation, so we write:

$$|+\rangle = \textcolor{yellow}{\bullet} \text{ and } |-\rangle = \textcolor{red}{\bullet}$$

With this we can rewrite (16.1) as follows:

$$\textcolor{yellow}{\bullet} = \{\bigcirc, \bullet\}$$

Then (16.2) becomes:

$$\textcolor{red}{\bullet} = \{\bigcirc, \overline{\bullet}\}$$

New CHALLENGE for you: try proving (16.3) and $|1\rangle = \frac{1}{\sqrt{2}}\big(|+\rangle - |-\rangle\big)$.

Now we can answer (16.3) in the affirmative:

$$|0\rangle = \frac{1}{\sqrt{2}}\big(|+\rangle + |-\rangle\big) = \{{\color{yellow}\bullet}, {\color{red}\bullet}\} = \{\{\bigcirc, \bullet\}, \{\bigcirc, \overline{\bullet}\}\} = \{\bigcirc, \bullet, \bigcirc, \overline{\bullet}\} = \{\bigcirc, \bigcirc\} = \{\bigcirc\} = \bigcirc$$

Likewise we can now state and prove the following:

$$|1\rangle = \frac{1}{\sqrt{2}}\big(|+\rangle - |-\rangle\big) = \{{\color{yellow}\bullet}, \overline{{\color{red}\bullet}}\} = \{\{\bigcirc, \bullet\}, \overline{\{\bigcirc, \overline{\bullet}\}}\} = \{\bigcirc, \bullet, \overline{\bigcirc}, \overline{\overline{\bullet}}\} = \{\bigcirc, \bullet, \overline{\bigcirc}, \bullet\} = \bullet$$

We can now easily measure our inputs in the X basis.

We develop something similar for the Y basis. We start with:

$$|+i\rangle = {\color{lime}\bullet} = \{\bigcirc, i\bullet\} = \frac{1}{\sqrt{2}}\big(|0\rangle + i|1\rangle\big)$$

All of this is just notation, including the fact that $i^2 = -1$.

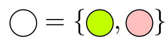
Likewise we have:

$$|-i\rangle = {\color{pink}\bullet} = \{\bigcirc, i\overline{\bullet}\} = \frac{1}{\sqrt{2}}\big(|0\rangle - i|1\rangle\big)$$

The new colors involved in this page are, in order: yellow, red, lime and pink.

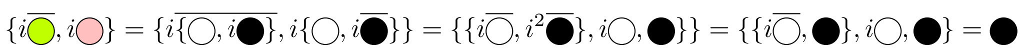
New CHALLENGE for you: try proving $\bigcirc = \{{\color{lime}\bullet}, {\color{pink}\bullet}\}$ and $\bullet = \{i\overline{{\color{lime}\bullet}}, i{\color{pink}\bullet}\}$

We claim:



This is immediate from the two definitions above. We also claim:

Let's prove this last one:

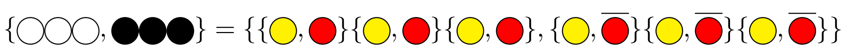


Recall that $i^2 = -1$ and the overline is just a unary minus placed above a circle for logistics reasons. Furthermore, the superposition operator inside a superposition operator works just like a set union operator.

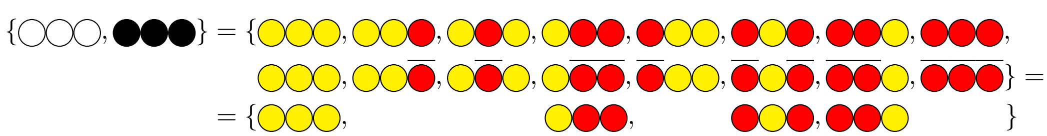
16.2 Playing the Game

There are two cases[1].

In the first case the players all receive inputs 0. In that case they all have to measure in the X basis. The tripartite entangled state becomes:



This then becomes:



If you go back to the section in which we formulated the quantum strategy you will find out that we said the following: if the players measure and obtain a red[2] (or pink) outcome they need to output a 1 otherwise (for yellow or lime) they output a 0. And we see above that triplets with an odd number of red outcomes cancel each other due to distructive interference.

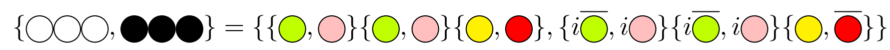
What is left is a set of triplets with an even number of red outcomes. The players then will output a sum $a+b+c$ that is even so in this case they always win (regardless of their actual results obtained when they measure, all possible outcomes are listed above).

New CHALLENGE for you: what's the other case?

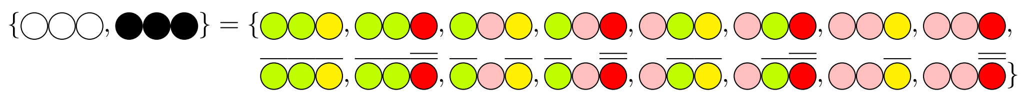
[1] We also want to state somewhere that these calculations bring to memory the rotational invariance of Bell states property. Not sure where we should say this so we dedicated this footnote to it, lest we forget to mention it at all.

[2] Now that we have the colors.

In the other three cases two of the players receive a 1 and the third one receives a 0 from the referee. The players that receive a 1 need to measure in the Y basis, the third one needs to measure in the X basis. We will calculate what happens in only one of the three cases[3], e.g. $(1, 1, 0)$. So Alice and Bob need to measure in the Y basis and Carol in the X basis. At the end we need to look into the resulting set of outcomes and replace red and pink with 1 and replace yellow and lime with 0 and calculate $a + b + c$ for each outcome to determine if the players win or not. The tripartite entangled state becomes:

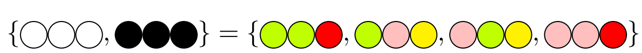


Performing the same calculation steps as before we have:



In calculating the second line of outcome triplets above we have used the fact that i only shows with the second power, so it transforms in a unary minus. We placed that as an extra phase on Carol's outcome (third in each triplet).

You see then that half of the resulting triplets cancel each other and we are left with:



New CHALLENGE for you: are you convinced? Why? Please explain.

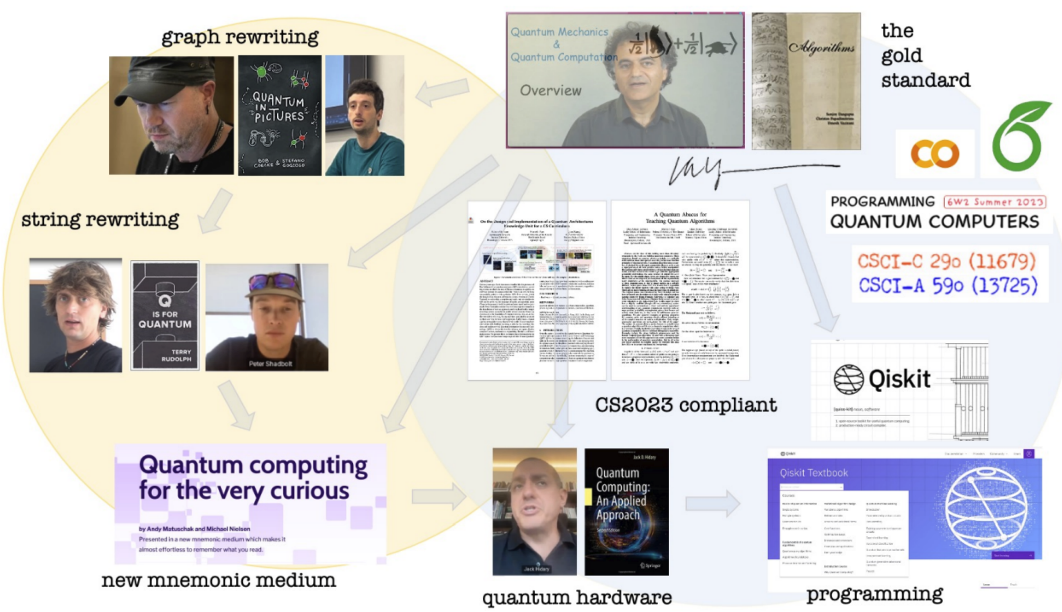
[3]And claim the same result in the other two, by symmetry.

Chapter 17

Commencement

Starting anew.

We recommend this[1] and this[2] (for now).



This[3] is also an excellent playlist (we call it the gold standard above).

Finally this[4] is a comprehensive (free) introductory book by a great author[5].

[1] https://legacy.cs.indiana.edu/~dgerman/2020/boot-camp/mathematics_qm_v21-martin-laforest.pdf

[2] https://legacy.cs.indiana.edu/~dgerman/2020/boot-camp/qc-high-2e-with-cover.pdf

[3] https://www.youtube.com/playlist?list=PL74Rel4IAsETUwZS_Se_P-fSEyEVQwni7

[4] https://www.thomaswong.net/introduction-to-classical-and-quantum-computing-1e4p.pdf

[5] https://www.thomaswong.net/

Updates on 07/14:

- Mermin, Miller (exams, selected, all solutions), Mike and Ike, Aaronson, Wallace, Regan and Lipton.

BRILLIANT Building a quantum-safe organization (QED-C).

Math primer from IQC at the U of Waterloo (Martin LaForest). And this is a special link (exercise).

Feynman (in NZ, in 1979). Dave Bacon (couple of years ago, also t=9960s). Jim Freericks (EdX)

Code from the PsiQuantum book.

Last CHALLENGE for you: what are some books you knew from before?

Bibliography

[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] David McMahon, *Quantum Computing Explained*, Wiley-IEEE Computer Society Press; 1st edition (November 19, 2007)

[5] Phillip Kaye, Raymond Laflamme, Michele Mosca *An Introduction to Quantum Computing*, Oxford University Press; 1st ed. (Jan 18, 2007)

[6] N. David Mermin, *Quantum Computer Science: An Introduction (1st Edition)*, Cambridge University Press; 1st edition (September 17, 2007)

[7] Michael A. Nielsen, Isaac L. Chuang *Quantum Computation and Quantum Information: 10th Anniversary Edition*, Cambridge University Press; Anniversary edition (January 31, 2011)

[8] Martin LaForêt, *Mathematics of Quantum Mechanics* `https://uwaterloo.ca/institute-for-quantum-computing/download-qcsys-math-primer`

[9] Eleanor G Rieffel, Wolfgang H Polak *Quantum Computing: A Gentle Introduction (Scientific and Engineering Computation)*, The MIT Press; Illustrated edition (August 29, 2014)

[10] Chris Berhardt, *Quantum Computing for Everyone*, The MIT Press. Reprint edition (September 8, 2020)

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

[12] Bob Coecke, “Quantum picturalism”, *Contemporary Physics*, 2010.

[13] Terry Rudolph (2017) *Q is for Quantum.* Terence G. Rudolph 7/19/2017.

[14] https://en.wikipedia.org/wiki/Quantum_pseudo-telepathy

[15] https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.71.1665

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

[17] 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

[18] Bettina Just, *Quantum Computing Compact: Spooky Action at a Distance and Teleportation Easy to Understand*, Springer Nature 2022.

[19] Adrian German, Marcelo Pias, Qiao Xiang and Sreesha Kuruvadi, *A Quantum Abacus for Teaching Quantum Algorithms*, available at https://legacy.cs.indiana.edu/~dgerman/2023/qseec/proposal.pdf. Paper presented at IEEE Frontiers in Education (FIE) 2023, October 2023, College Station, TX.

[20] Adrian German and Alex Alani, *The GHZ Game with Misty States*, August 2023, preprint available at https://legacy.cs.indiana.edu/~dgerman/2023/qseec/ghz-game-with-misty-states.pdf

[21] Alex Alani, Adrian German and Terry Rudolph, *The GHZ Game in Black and White*, August 2023, preprint available at https://legacy.cs.indiana.edu/~dgerman/2023/qseec/ghz-game-in-black-and-white.pdf

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

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

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

[25] Abe Asfaw, Programming in IBM Qiskit (Season 1, Ep. 6: Bernstein-Vazirani) https://www.youtube.com/watch?v=sqJIpHYl7oo&t=6s

[26] Peter Shor, MIT Open Courseware, Lecture 12: Superdense Coding, lecture by Peter Shor, scribed by Sebastian Bauer. 10/14/2003. Available at https://ocw.mit.edu/courses/18-435j-quantum-computation-fall-2003/4f97e39d2509d075121823e6b999455c_qc_lec12.pdf

[27] Old Qiskit Textbook, Superdense Coding Python Notebook, available at https://github.com/Qiskit/textbook/blob/main/notebooks/ch-algorithms/superdense-coding.ipynb

[28] Macauley Coggins, *Introduction to Quantum Computing with Qiskit*, available at `https://quantumcomputinguk.org/tutorials/superdense` 2019

[29] John Preskill, Online Notes for PHYS 219 and CSCI 219 Quantum Computation. This is chapter 4: Quantum Entanglement available at `http://theory.caltech.edu/~preskill/ph229/notes/chap4.pdf`

[30] Old Qiskit Textbook, Grover's Algorithm Python Notebook, available at `https://github.com/Qiskit/textbook/blob/main/notebooks/ch-algorithms/grover.ipynb` (GitHub repository)

[31] Dayton Ellwanger, *A Quantum Computing Pamphlet*, custom published, Monee, IL, 02/15/2020. Concise presentation of Grover (linear algebra).

[32] Terry Rudolph, `https://www.qisforquantum.org/` (book website)

[33] Yaoyun Shi, *Both Toffoli and Controlled-NOT need little help to do universal quantum computation* `https://arxiv.org/abs/quant-ph/0205115`