Jumpstarting Quantum Computing: A Guide for Teachers and Learners

Adrian German and John Phillips
Indiana University Bloomington

January 17, 2025 at around 12:51am GMT

Abstract

This is a collection of milestones to share with potential co-authors for a paper (whose structure is thus communicated/proposed) for submission to ACM InRoads on "Quantum Computing Education: A Curricular Perspective." The submission to InRoads is a recommendation from the CS2023 Curricular Guidelines Task Force & Report.

We represent classical bits 0 and 1 as $\bigcirc$ and $\bullet$.

As qubits $\bigcirc = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $\bullet = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

An elementary quantum operation is analogous to an elementary gate in a classical circuit. One of the most important examples is the Hadamard gate, denoted by H, which operates on a single qubit as follows:

$$\mathrm{H}(\bigcirc) = \{\bigcirc, \bullet\} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle$$

$$\mathrm{H}(\bullet) = \{\bigcirc, \overline{\bullet}\} = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$$

Normally H is represented by the unitary matrix $\mathrm{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. The definitions above effectively extract the two columns of this matrix. The misty state formalism introduced by Terry Rudolph eliminates the need to use matrices and Dirac algebra while accurately describing quantum phenomena.

The Hadamard gate is a genuinely quantum gate, since it creates a superposition of states. Two other gates that we will use (X and Z) are more classical in nature, for example: X(◯) = ● and X(●) = ◯. The X gate is also known as the NOT gate. Meanwhile Z(◯) = ◯ and Z(●) = $\overline{\text{●}}$. The Pauli Z gate is also known as the phase-flip gate. Three very well-known properties of the gates shown thus far can be expressed as follows: Z(H(◯)) = H(X(◯)) while H(H(◯)) = ◯ and H(H(●)) = ●. To prove them we need additional rules, such as:

$$f(\{s_1, s_2, \ldots s_n\}) = \{f(s_1), f(s_2), \ldots f(s_n)\}$$

The property above is called *linearity* and holds for any quantum gate f applied to a quantum state that is a superposition of states $s_1, s_2, \ldots s_n$. As an example the phase operator is linear:

$$\overline{\{s_1, s_2, \ldots, s_n\}} = \{\overline{s_1}, \overline{s_2}, \ldots, \overline{s_n}\} \text{ and } \overline{f(s)} = f(\overline{s})$$

Two or more superpositions of states can be combined via set union if they're at the same depth level and have the same number of states:

$$\{\{s_1, s_2, \ldots, s_n\}, \{t_1, t_2, \ldots, t_n\}\} = \{s_1, s_2, \ldots, s_n, t_1, t_2, \ldots, t_n\}$$

Same depth level means that $\{s_1, \{s_2, s_3\}\}$ cannot be reduced.

To illustrate how our rules work we write the following:

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

We see that $\overline{\overline{s}} = s$ and that $\{s_1, \overline{s_2}, s_1, s_2\} = \{s_1, s_1\}$ as expected.

We also see that $\{s_1, s_1\} = s_1$ in other words a superposition of two (or more) identical states can be reduced to that state as a certain (i.e., sure) measurement outcome.

A misty state is just an unnormalized quantum state.

For example $\{\bigcirc, \bigcirc, \bullet\}$ can be represented by the vector $\begin{pmatrix} 2 \\ 1 \end{pmatrix}$.

After normalization $\{\bigcirc, \bigcirc, \bullet\} = \frac{1}{\sqrt{5}} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = \frac{2}{\sqrt{5}}|0\rangle + \frac{1}{\sqrt{5}}|1\rangle$.

Because misty states don't need to be normalized their formalism is entirely pure, that is, devoid of numbers (coefficients). There is however one subtlety to be careful about in going between the mist and the quantum state—in the misty formalism we can only incorporate boxes (gates) whose representation in the traditional (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. We will examine how the misty formalism can be extended but, for now, we need to point out that it is a remarkable and nontrivial feature of quantum computation that such unitaries can be universal, i.e. used to simulate all unitaries, even ones with irrational complex entries.

Given an arbitrary misty state, how can one determine the probability of each outcome in the state upon measurement? The "squaring rule" is the only rule with numbers in the original (pure) misty state formalism and says:

$$\{\underbrace{\bigcirc\cdots\bigcirc}_{n},\overbrace{\bullet\cdots\bullet}^{m}\} = \frac{n}{\sqrt{n^2+m^2}}|0\rangle + \frac{m}{\sqrt{n^2+m^2}}|1\rangle$$

Now consider the following quantum snack: {🍒, {🍎, 🍉}}.
Can you calculate the probability of getting each piece of fruit?

Since misty states always contain a whole number of black and white balls the following state does not admit a representation in the (original, pure) misty state formalism:

$$\frac{1}{\sqrt{3}}|0\rangle + \sqrt{\frac{2}{3}}|1\rangle$$

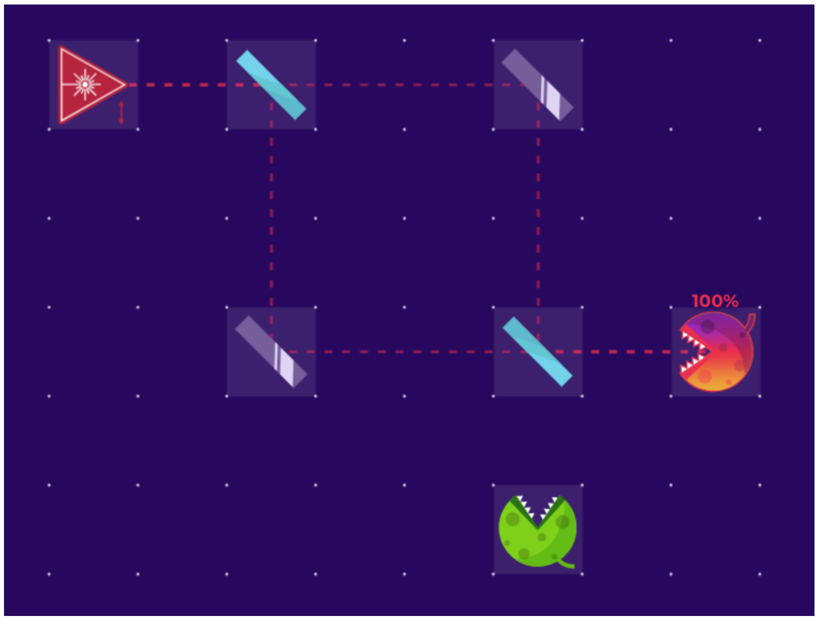
This follows from the "squaring rule" and because 3 cannot be expressed as a sum of two perfect squares.

Since every quantum gate is represented by a unitary matrix we can introduce the notion of eigenvector. In simple terms an eigenvector is a state that is a fixed point (up to a coefficient) for the quantum operator under consideration. One can easily prove that:

$$\mathrm{X}(\{\bigcirc, \bullet\}) = \{\bigcirc, \bullet\} \text{ and } \mathrm{X}(\{\bigcirc, \overline{\bullet}\}) = \overline{\{\bigcirc, \overline{\bullet}\}}$$

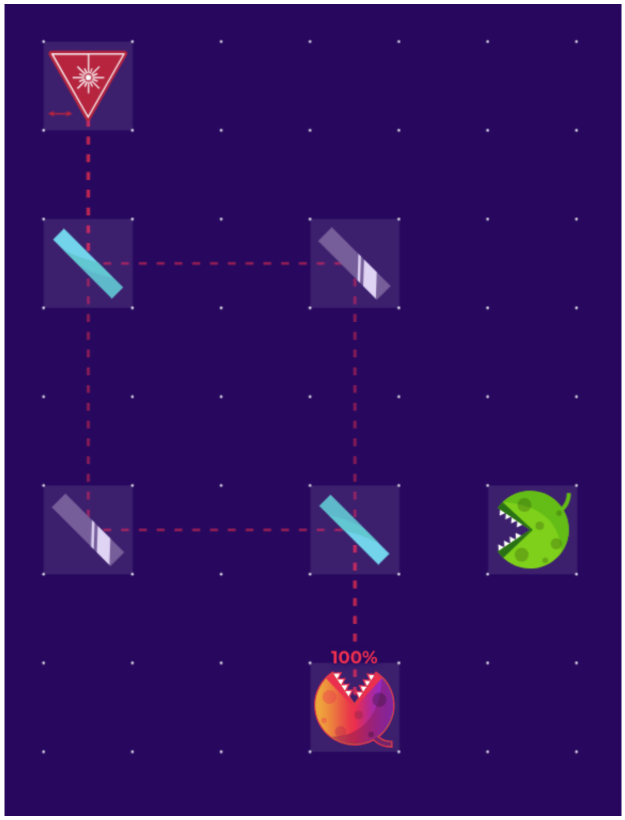
This way we'd prove using misty states that $|+\rangle$ and $|-\rangle$ are in fact the eigenvectors of $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ and that the corresponding eigenvalues are ± 1.

Now using the quantum flytrap environment, let's perform the following experiment using individual photons:



If we associate the horizontal direction with $|0\rangle$ and the vertical direction with $|1\rangle$ each beam splitter acts as a Hadamard gate and the mirrors are acting as NOT gates. Thus the experiment is equivalent to $\mathrm{H}(\mathrm{H}(\bigcirc)) = \bigcirc$. One can trace it using the misty state formalism on the interferometer's branches.

Likewise this is the proof that H(H(●)) = ●.



What happens if we introduce a delay on one of the branches?

(We'll come back to this question later.)

Another basic gate is the controlled-NOT, or CNOT. It operates upon two qubits, with the first acting as a control qubit and the second as the target qubit (for us CNOT $\equiv \overrightarrow{\mathrm{X}}$ and we will explain the arrow shortly: it basically points from the control qubit to the target qubit) The CNOT gate flips the second bit if and only if the first bit is $\CIRCLE$. So we have:

$$\overrightarrow{\mathrm{X}}(\Circle\Circle) = \Circle\Circle$$
$$\overrightarrow{\mathrm{X}}(\Circle\CIRCLE) = \Circle\CIRCLE$$
$$\overrightarrow{\mathrm{X}}(\CIRCLE\Circle) = \CIRCLE\CIRCLE$$
$$\overrightarrow{\mathrm{X}}(\CIRCLE\CIRCLE) = \CIRCLE\Circle$$

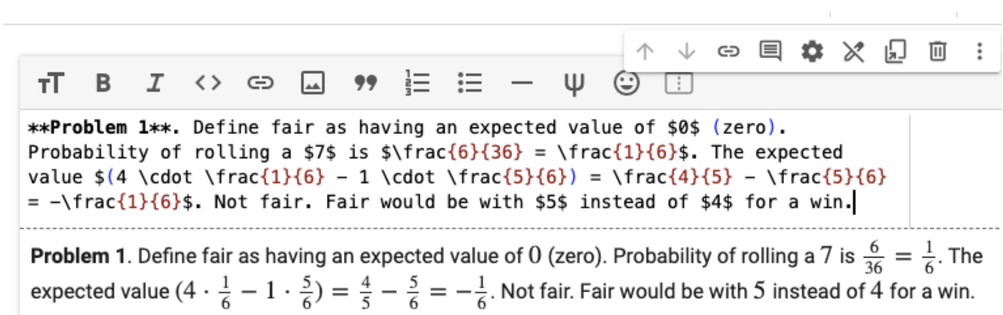
The only prerequisites for an introductory class to quantum computing using the misty states formalism (as shown here) should be:

- access to "Q is for Quantum" and to Google Colab,
- basic knowledge of how to write simple programs in Python.

To define "simple" consider the following problem:

> Craps is a game played with two dice. You decide to play. Each bet is one chip. The goal is to roll a seven (with two dice). The house pays 4 chips plus your original chip if you win. Is this fair? (If not, define fair).

We could be asked to solve this problem in Google Colab in three different ways. The first approach could be writing down the actual answer (mathematical argument) in LaTeX as follows:



```
**Problem 1**. Define fair as having an expected value of $0$ (zero).
Probability of rolling a $7$ is $\frac{6}{36} = \frac{1}{6}$. The expected
value $(4 \cdot \frac{1}{6} - 1 \cdot \frac{5}{6}) = \frac{4}{5} - \frac{5}{6}
= -\frac{1}{6}$. Not fair. Fair would be with $5$ instead of $4$ for a win.
```

Problem 1. Define fair as having an expected value of 0 (zero). Probability of rolling a 7 is $\frac{6}{36} = \frac{1}{6}$. The expected value $(4 \cdot \frac{1}{6} - 1 \cdot \frac{5}{6}) = \frac{4}{5} - \frac{5}{6} = -\frac{1}{6}$. Not fair. Fair would be with 5 instead of 4 for a win.

Second approach would be to simply model (simulate) it in Python:

```
[3] import random

    times = 1000000
    money = 0

    for i in range(times):
      d1 = random.randint(1, 6)
      d2 = random.randint(1, 6)
      sum = d1 + d2
      if sum == 7:
        money += 4
      else:
       money-= 1

    print(money/times)
```

```
-0.167965
```

Of course, in this case, we'd get a slightly different answer every time we run the program but careful choice of parameters would ensure that the experiment converges by the time our result is printed so the answer is close to the (expected) theoretical result.

Finally, a third approach would be to write a program to help us determine the probability of getting a 7 (assuming it's otherwise difficult to calculate). This is a middle ground approach between the two shown thus far, and since calculating the probability of rolling a 7 with two dice is easy let's illustrate this with a different problem:

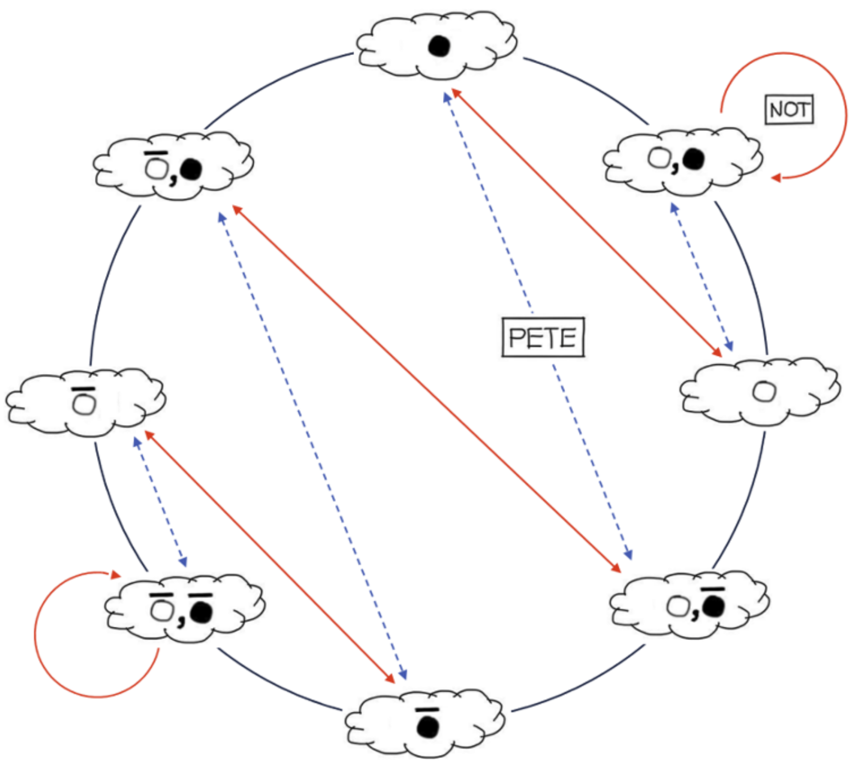
> "You roll three dice. What is the probability that the sum of the three dice is not a prime number?"

Here's the Python code:

```
count = 0

for d1 in range(1, 7):
  for d2 in range(1, 7):
    for d3 in range(1, 7):
      sum = d1 + d2 + d3
      if sum not in [3, 5, 7, 11, 13, 17]:
        count += 1
      else:
        pass

print(count/(6 * 6 * 6))
```

```
0.6620370370370371
```

Now following Andrew Helwer we can put together a unit circle state machine that can help us determine transitions between (misty) states for one-qubit quantum circuits constructed with H and X gates:



The X gate has two eigenvectors, and both are visible in this diagram. One of them is $\{\bigcirc, \bullet\}$ as discussed earlier. The other one (has already been revealed and) is not $\{\overline{\bigcirc}, \overline{\bullet}\} = -|+\rangle$ since $\mathrm{X}(\{\overline{\bigcirc}, \overline{\bullet}\}) = \{\overline{\bigcirc}, \overline{\bullet}\}$ simply means:

$$\mathrm{X}(-|+\rangle) = \mathrm{X}(\overline{\{\bigcirc, \bullet\}}) = \overline{\mathrm{X}(\{\bigcirc, \bullet\})} = \overline{\{\bigcirc, \bullet\}} = -|+\rangle$$

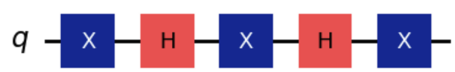
which simply states that $\mathrm{X}(\{\bigcirc, \bullet\}) = \{\bigcirc, \bullet\}$ so it refers to the same exact eigenstate. What then is the second eigenvector of X as seen in this state diagram?

We can now create a simple circuit in Qiskit and test our predictive (theoretical) powers against the quantum circuit simulator:

```
[4] !pip install qiskit pylatexenc
```

```
[5] from qiskit import QuantumCircuit
```

```
[6] qc = QuantumCircuit(1)
    qc.x(0)
    qc.h(0)
    qc.x(0)
    qc.h(0)
    qc.x(0)
    qc.draw(output='mpl')
```

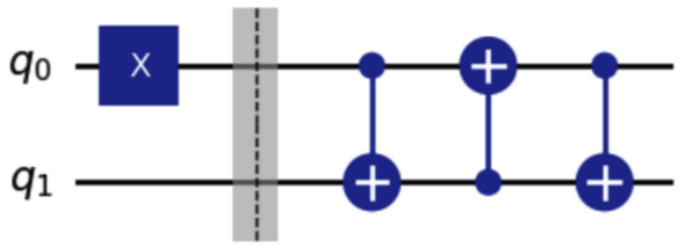



```
[7] from qiskit.quantum_info import Statevector
    a = Statevector.from_instruction(qc)
    a.draw('latex')
```

$-|0\rangle$

Starting from ◯ we can trace the transitions on the unit circle state diagram to confirm the result. We have just shown how using the misty state formalism we built a useful tool that is general (i.e., it is, in fact, mist agnostic).

Now consider the following two-qubit circuit:




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

$$|10\rangle$$

It implements: $\overrightarrow{\mathrm{X}}(\overleftarrow{\mathrm{X}}(\overrightarrow{\mathrm{X}}(\mathrm{X}(\bigcirc)\bigcirc))) \equiv \mathrm{SWAP}(\bullet\bigcirc) = \bigcirc\bullet$. This illustrates the meaning (and necessity) of the over arrows (as they point from the control qubit to the target). This also reminds us that Qiskit reports the output backwards (i.e., from left to right once the entire picture has been rotated clockwise with 90°).

The superposition operator is a set operator; the order of states (outcomes) inside does not matter, although an order might be preferred. As a reminder the phase operator acts as follows:

$$-\{s_1, s_2, \ldots, s_n\} = \overline{\{s_1, s_2, \ldots, s_n\}} = \{\overline{s_1}, \overline{s_2}, \ldots, \overline{s_n}\}$$

A multi-qubit state is written as a tensor product and the order now matters, i.e. ◯● $\neq$ ●◯. The good news is that the notation simplifies even further to the point where it now closely matches Dirac notation, e.g. ◯● $= |01\rangle$, ●●◯ $= |110\rangle$, etc.

Phase acts on a multi-qubit state like in a multiplication:

$$-(s_1 s_2 \ldots s_n) = \overline{s_1} s_2 \ldots s_n = s_1 \overline{s_2} \ldots s_n = s_1 s_2 \ldots \overline{s_n} = \overline{s_1 s_2 \ldots s_n}$$

Other properties (either clear by now, or that can be easily proved):

$$\{s_1 s_2, s_3 s_2\} = \{s_1, s_3\} s_2$$

$$s_3 \{s_1, \overline{s_2}\} = \overline{s_3} \overline{\{s_1, \overline{s_2}\}}$$

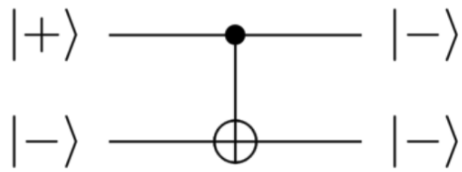
$$\overline{s_1}\ \overline{s_2} \ldots \overline{s_{2n}} = s_1 s_2 \ldots s_{2n}$$

$$\overline{s_1}\ \overline{s_2} \ldots \overline{s_{2n+1}} = \overline{s_1 s_2 \ldots s_{2n+1}}$$

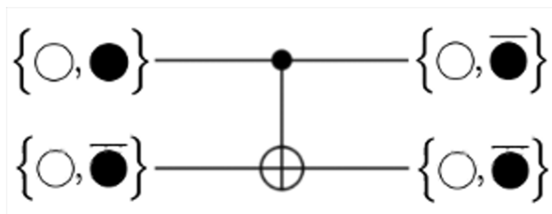
$$\{s_1, \overline{s_2}, s_3, s_2\} = \{s_1, s_3\}$$

In a superposition, states in antiphase cancel each other like in a sum.

So now let's prove this property (also known as phase kickback):



In our misty state formalism it can be restated as follows:



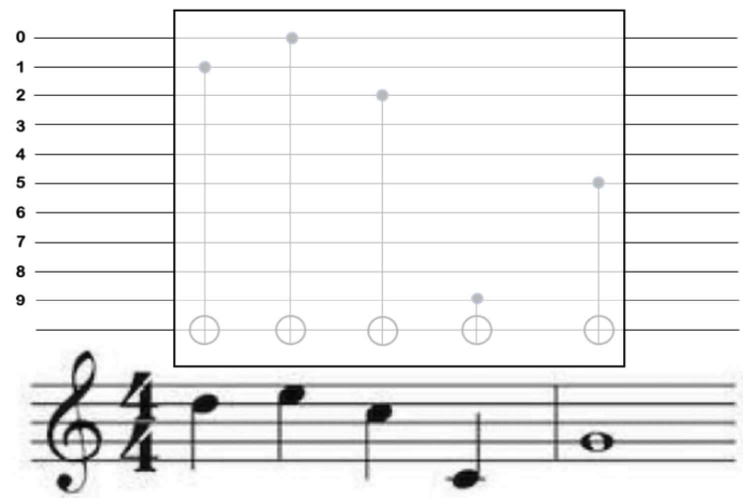
Here's how the proof proceeds:

$$
\begin{aligned}
\overrightarrow{\mathrm{X}}(\{\bigcirc,\bullet\}\{\bigcirc,\overline{\bullet}\}) &= \overrightarrow{\mathrm{X}}(\{\bigcirc\{\bigcirc,\overline{\bullet}\},\bullet\{\bigcirc,\overline{\bullet}\}\}) = \\
&= \{\overrightarrow{\mathrm{X}}(\bigcirc\{\bigcirc,\overline{\bullet}\}),\overrightarrow{\mathrm{X}}(\bullet\{\bigcirc,\overline{\bullet}\})\} = \\
&= \{\bigcirc\{\bigcirc,\overline{\bullet}\},\bullet\{\bullet,\overline{\bigcirc}\}\} = \\
&= \{\bigcirc\{\bigcirc,\overline{\bullet}\},\overline{\bullet}\overline{\{\bullet,\overline{\bigcirc}\}}\} = \\
&= \{\bigcirc\{\bigcirc,\overline{\bullet}\},\overline{\bullet}\{\overline{\bullet},\overline{\overline{\bigcirc}}\}\} = \\
&= \{\bigcirc\{\bigcirc,\overline{\bullet}\},\overline{\bullet}\{\bigcirc,\overline{\bullet}\}\} = \\
&= \{\{\bigcirc,\overline{\bullet}\}\{\bigcirc,\overline{\bullet}\}\}
\end{aligned}
$$

We can now discuss the Bernstein-Vazirani algorithm:

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.

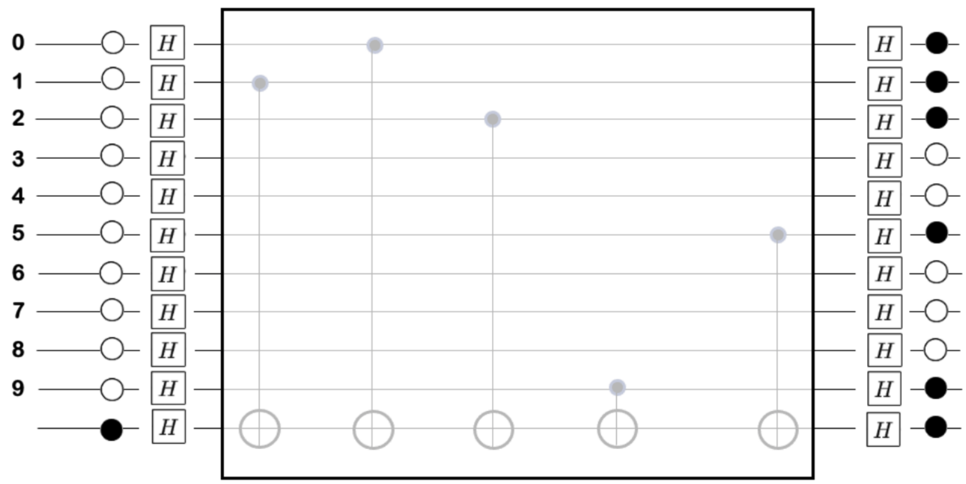
Here's an example with $n = 10$.



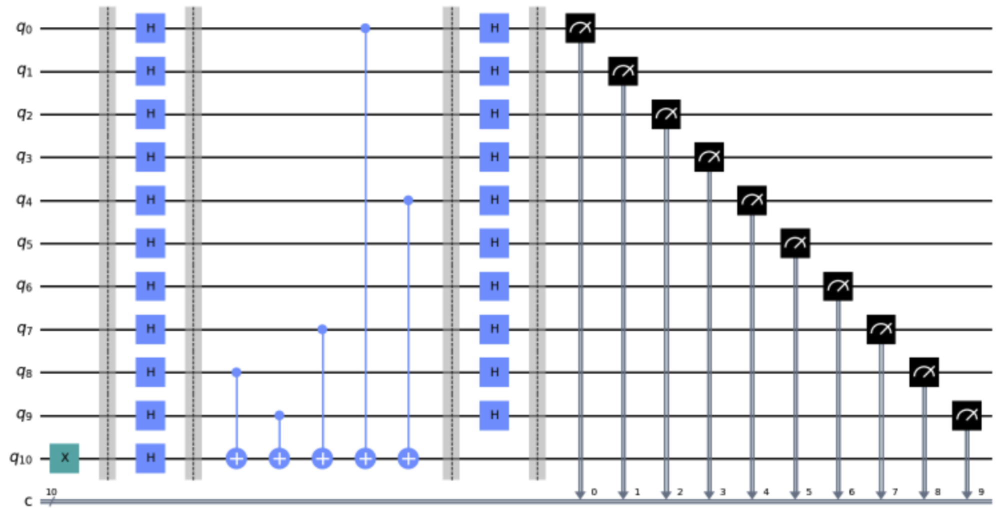
The theme from "Close Encounters of the Third Kind" is there to reinforce the pattern but also to allow me to say that the sequence (order) of the `C-NOT` gates is not relevant. The circuit itself is called an *oracle* and it hides a "secret" string of controls to the gates on the bottom wire. The task 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.

Notice that if the gates in the black box had been reversed there would have been no challenge. As they are, though, we would need to convert the inputs to superpositions (using Hadamard gates) so we can then apply the phase kickback phenomenon that we just proved a bit earlier:



Here's the implementation in Qiskit. Taking into account the order in which Qiskit prefers to report the output the implementation of the black box below is an exact reflection of our initial diagram:



Reminder that these milestones are just pictures. In the text (narration) we'd mention that "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." (from Quantum Computing: Progress and Prospects, National Academies of Sciences, 2019).

The code for the Qiskit circuit we just presented:

```
[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))
```

And here's how we obtain the answer (in one step):

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

     {'1110010001': 1}
```

So far, we have established that one reason to study quantum computation is that it is the only model of computation that violates the extended Church-Turing thesis. It does not violate (see also Tommy Wong's book) the original Church-Turing thesis: what is impossible to compute remains impossible to compute, just that some things could be computed faster by quantum computers. The other reason is that this model of computation makes use of physical concepts for which we (as humans, at our scale) have no intuition. One of them is entanglement, capable of superluminal correlations (Bettina Just).

In [2]:

```
import qiskit
```

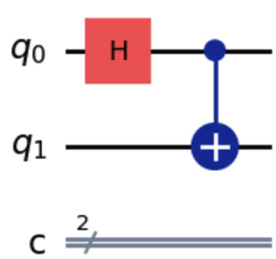
In [3]:

```
from qiskit import *
```

In [12]:

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

Out[12]:



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

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

Is this quantum state an entangled state?

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

Answer: no, because

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

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`

In [13]:

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

Out[13]:

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

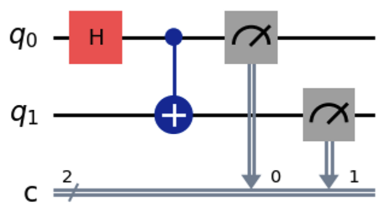
Here's how we create (and measure) an entangled state in Qiskit:

In [ ]:
```
!pip install qiskit-aer
```

In [15]:
```
circuit.measure(range(2), range(2))

from qiskit_aer import AerSimulator
aersim = AerSimulator()
circuit = transpile(circuit, aersim)
```

In [16]:
```
circuit.draw(output='mpl')
```

Out[16]:

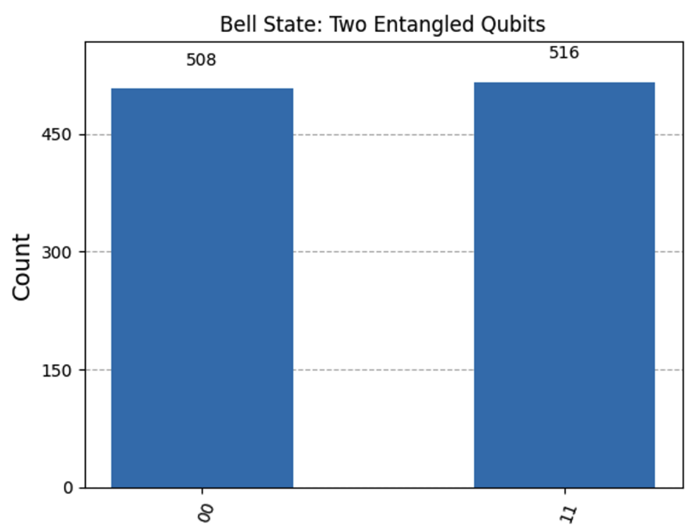


In [18]:
```
result = aersim.run(circuit).result()
counts = result.get_counts(circuit)
print(result.get_counts(circuit))
```

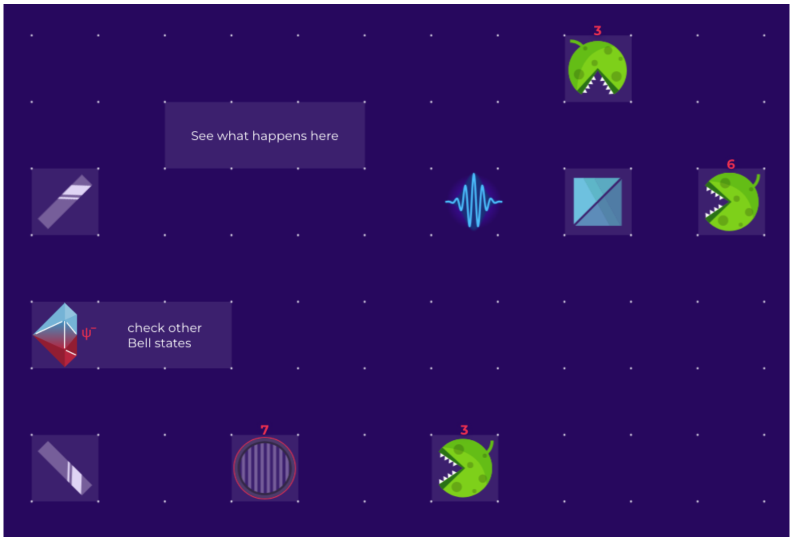
```
{'11': 516, '00': 508}
```

In [19]:
```
from qiskit.visualization import plot_histogram
plot_histogram(counts, title='Bell State: Two Entangled Qubits')
```

Out[19]:

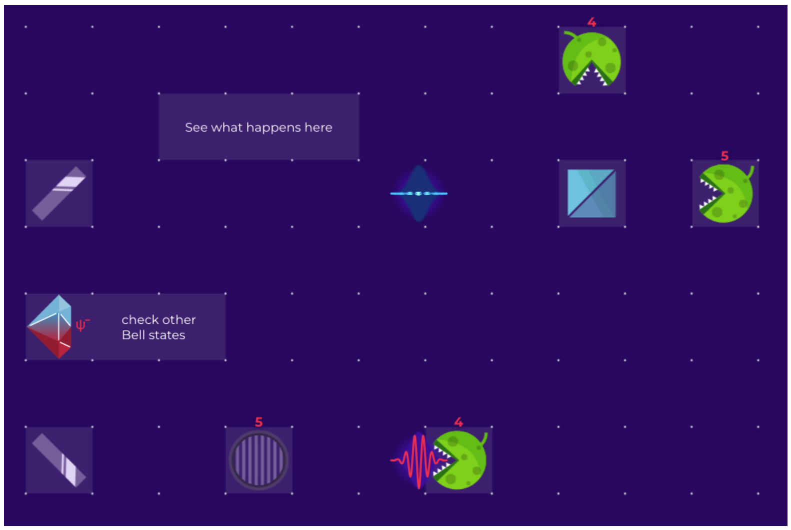


One can set up and perform experiments in the Quantum Flytrap to better understand quantum entanglement:



In the experiment above a measurement on the lower branch resulted in the photon being absorbed (as the 7-th such photon on that branch). That instantaneously collapsed the state of the other photon which will now enter the detector on the right (and change its counter from 6 to 7—in this environment no photon is ever lost).

The alternative situation is when the photon on the bottom branch does not get absorbed, instead passes through:



The state of the other photon immediately collapses to its other state.

The photon on that branch will be collected in the detector at the top, whose counter will change from 4 to 5—something that will happen with the detector at the very bottom, as well.

Since we brought up the fact that every gate can be represented by a matrix we can now reveal that Qiskit can provide that information for individual gates as well as whole circuits.

In [ ]:
```
!pip install qiskit
```

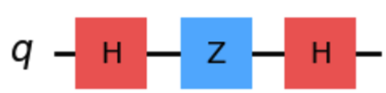
In [ ]:
```
!pip install pylatexenc
```

In [5]:
```
import qiskit, pylatexenc
from qiskit import QuantumCircuit
from qiskit.quantum_info import Operator
```

In [6]:
```
def circuit():
  q = qiskit.QuantumCircuit(1)
  q.h(0)
  q.z(0)
  q.h(0)
  return q
```

In [10]:
```
circuit().draw(output='mpl')
```

Out[10]:



In [11]:
```
Operator.from_circuit(circuit()).draw(output='latex')
```

Out[11]:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Two circuits are equivalent when they have the same (composite) matrix. This can lead into the ZX calculus and Quantum in Pictures.

More circuits created on the fly via abstractions (procedures) designed specifically for that purpose, with associated matrices:

In [13]:

```
def d():
  q = qiskit.QuantumCircuit(1)
  q.x(0)
  q.z(0)
  return q

d().draw(output='mpl')
```

Out[13]:

In [16]:

```
Operator.from_circuit(d()).draw(output='latex')
```

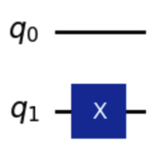
Out[16]:

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

In [27]:

```
def example():
  q = qiskit.QuantumCircuit(2)
  q.x(1)
  return q

example().draw(output='mpl')
```

Out[27]:



In [28]:

```
Operator.from_circuit(example()).draw(output='latex')
```

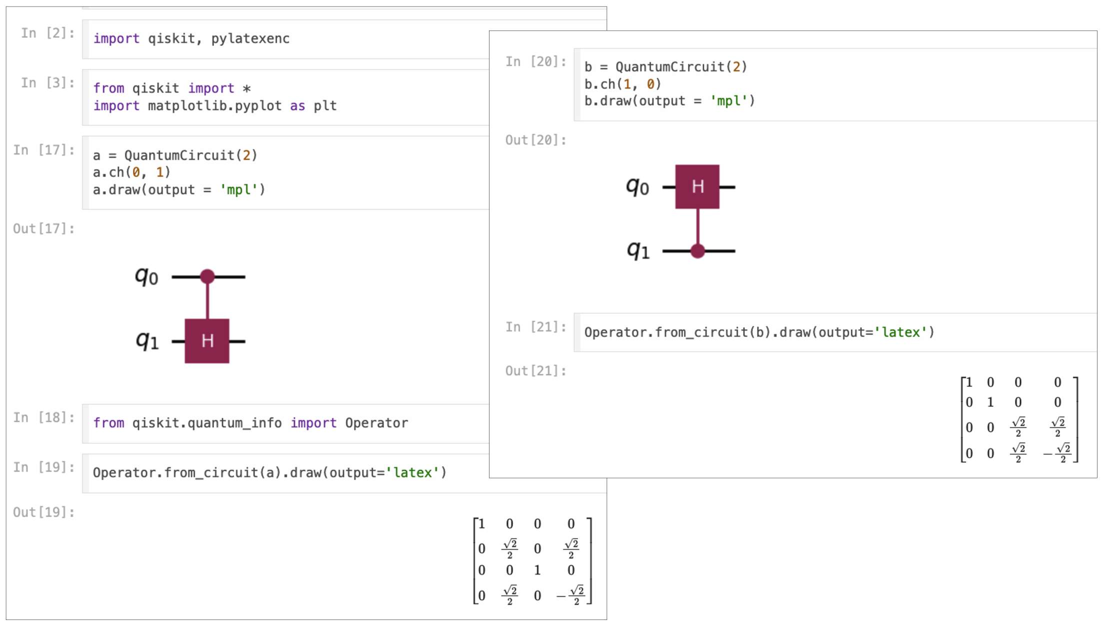
Out[28]:

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Second example introduces tensor product with the identity matrix.

Now we can bring Burd's book into the course too.

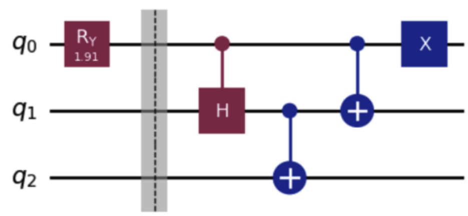
The controlled Hadamard is a gate that does not fit into the original (pure) misty state formalism (and now we also know why):



Here's a circuit that creates a three qubit W-entangled state:

```
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')
```

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

Recall that our goal is to become fluent in the original misty state formalism so we can then extend it (if it needs to be extended, and in any way that that may be necessary) so we can then graduate to the regular (conventional) mathematical apparatus.

Take a look at the first gate in the preceding circuit. It's time to remind you that we already talked about it (albeit indirectly) a long time ago at the beginning of this document.

VIII. $R_y(\theta_3)$

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[12]. So let's consider a specific rotation gate that will be useful a bit later. The first axiom is:

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

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:

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

From this we can already calculate in general how this gate acts on a generic superposition of $|0\rangle = \bigcirc$ and $|1\rangle = \bullet$. The reason this gate does not exist in the MSF will become clear below. First off $\theta_3 = 2\arccos\frac{1}{\sqrt{3}}$ and so the matrix is:

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

[12]e.g., $\mathtt{X}(\{\alpha\bigcirc, \beta\bullet\}) = \{\beta\bigcirc, \alpha\bullet\}\forall\alpha,\beta \in \mathbb{C}$

So, now, how do we process this gate in our formalism (in class)?

One solution is to add coefficients and we can then calculate:

Let's calculate: the initial state is still $\bigcirc\bigcirc\bigcirc$. After the rotation we have $\{\frac{1}{\sqrt{3}}\bigcirc, \sqrt{\frac{2}{3}}\bullet\} = \{\frac{1}{\sqrt{3}}\bigcirc\bigcirc\bigcirc, \frac{1}{\sqrt{3}}\bullet\bullet\bullet\}$.

When the controlled Hadamard kicks in we have:

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

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

The coefficients are the probability amplitudes:

After the first `C-NOT` we have:

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

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

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

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

Finally after the `X` gate we have:

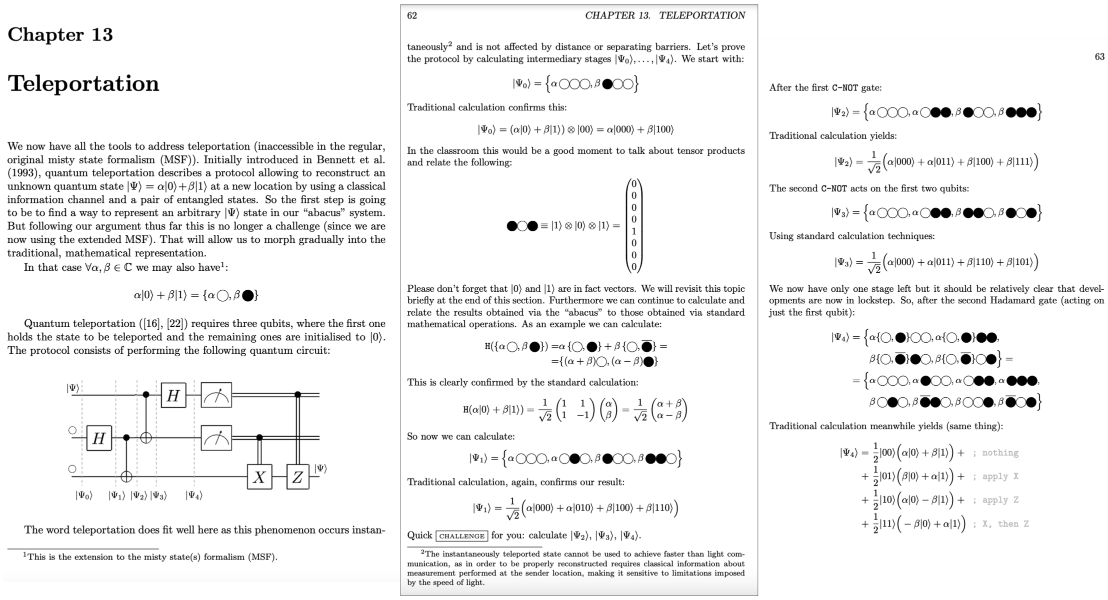
$$
\begin{aligned}
&\{\frac{1}{\sqrt{3}}\bullet\bigcirc\bigcirc, \frac{1}{\sqrt{3}}\bigcirc\bullet\bigcirc, \frac{1}{\sqrt{3}}\bigcirc\bigcirc\bullet\} = \\
&\frac{1}{\sqrt{3}}|001\rangle + \frac{1}{\sqrt{3}}|010\rangle + \frac{1}{\sqrt{3}}|100\rangle
\end{aligned}
$$

As before the Dirac notation is with Qiskit ordering.

Everything checks out.

Transition to Dirac notation is now accessible/possible.

We can now discuss teleportation:



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:

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

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:

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

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 $\boxed{\text{CHALLENGE}}$ 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:

$$|\Psi_2\rangle = \left\{\alpha \bigcirc\bigcirc\bigcirc, \alpha \bigcirc\bullet\bullet, \beta \bullet\bigcirc\bigcirc, \beta \bullet\bullet\bullet\right\}$$

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:

$$|\Psi_3\rangle = \left\{\alpha \bigcirc\bigcirc\bigcirc, \alpha \bigcirc\bullet\bullet, \beta \bullet\bullet\bigcirc, \beta \bullet\bigcirc\bullet\right\}$$

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):

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

Traditional calculation meanwhile yields (same thing):

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

As an aside, by now in the course the following topics will have been covered: Bernstein-Vazirani (as shown earlier in this document), Deutsch-Josza, Grover search, superdense coding and now teleportation. We also have a module based on Mathematica demonstrating the CHSH game (as put together by John McNally from Wolfram Research) and we have also discussed W-entangled states and referred to John Watrous' online IBM course (videos). The last section in the course discusses the GHZ game (using three different approaches).

Here's how we implement teleportation in Qiskit:

burd.ipynb

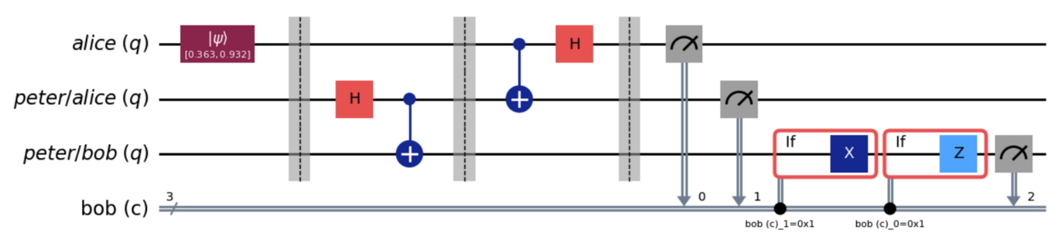
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[7]
```python
generate_amplitudes()
```
```
(0.8402911506075744, 0.5421353910330875)
```

[8]
```python
def add_gates(circ, alpha, beta):
    circ.initialize([alpha, beta], 0)
    circ.barrier()
    circ.h(1)
    circ.cx(1, 2)
    circ.barrier()
    circ.cx(0, 1)
    circ.h(0)
    circ.barrier()
    circ.measure(0, 0)
    circ.measure(1, 1)
    with circ.if_test((1, 1)):
        circ.x(2)
    with circ.if_test((0, 1)):
        circ.z(2)
    circ.measure(2, 2)
    return circ
```

[9]
```python
alpha, beta = generate_amplitudes()
circ = create_registers()
circ = add_gates(circ, alpha, beta)
# display(circ.draw('latex', cregbundle=False))
circ.draw(output='mpl')
```

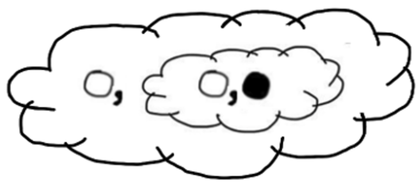


[10]
```python
from qiskit_aer import AerSimulator
device = AerSimulator() # Aer.get_backend("qasm_simulator")

shots = 1000
job = device.run(circ, shots=shots)
print(job.job_id())

result = job.result()
counts = result.get_counts(circ)
counts_m = marginal_counts(counts, [2])
number_of_0s = counts_m.get('0')
number_of_1s = counts_m.get('1')
alpha = np.sqrt(number_of_0s / shots)
beta = np.sqrt(number_of_1s / shots)
print("|\u03C8\u27E9 ({:.4f}, {:.4f})".format(alpha, beta))
```
```
3bbabe3b-cd0d-4066-84a6-3b8035b699e4
|ψ⟩ (0.3768, 0.9263)
```

This is the most comprehensive implementation out there. Note that probability amplitudes of the teleported state ($\alpha, \beta \in \mathbb{R}$) are real.

In the original (pure) misty state formalism two or more separate mists can be combined when (and only when) they (a) are at the same level and (b) have the same cardinality. This leads us to states that are irreducible. For example:

$$\{\bigcirc, \{\bigcirc, \bullet\}\} = \cos\frac{\pi}{8}|0\rangle + \sin\frac{\pi}{8}|1\rangle = 0.923879532511...|0\rangle + 0.382683432365...|1\rangle$$

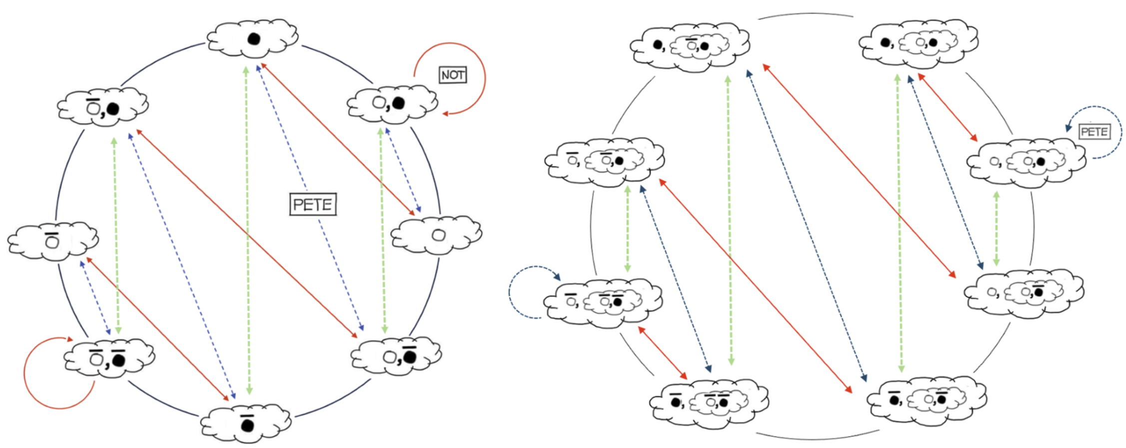


$$\{\bigcirc, \{\bigcirc, \bullet\}\} \approx \{\bigcirc, \bigcirc, \bullet\} = \frac{2}{\sqrt{5}}|0\rangle + \frac{1}{\sqrt{5}}|1\rangle = 0.89|0\rangle + 0.45|1\rangle$$

$$\{\bigcirc, \{\bigcirc, \bullet\}\} \approx \{\{\bigcirc, \bigcirc\}, \{\bigcirc, \bullet\}\} = \{\bigcirc, \bigcirc, \bigcirc, \bullet\} = \frac{3}{\sqrt{10}}|0\rangle + \frac{1}{\sqrt{10}}|1\rangle = 0.95|0\rangle + 0.32|1\rangle$$

The picture above shows three things: (a) there is a straightforward conversion from a misty state to an algebraic expression; (b) that the original, irreducible state can be approximated reasonably well within the original (pure) misty state formalism and (c) that the state we have chosen (smallest irreducible state with mist inside mist) is in fact one of the eigenvectors of the Hadamard gate.

Below we see the first two levels of a construction (so, the circle on the right should in fact be on top of the other) that reaches irreducible states immediately but is able to get (in the limit) arbitrarily close to (and thus approximate) any vector in the space. The construction resembles the Stern-Brocot tree that use exclusively rational numbers to approximate all irrationals. Two important consequences to this: first, we can get arbitrarily close to any state on the unit circle. The construction converges to a countably infinite set of states that effectively approximates the (uncountably) infinite set of states with real probability amplitudes. Second, the string representation of the quantum states involved in this expansion grows exponentially.



The construction is clearly recursive (like the recursive definition of a line via midpoints when endpoints are reasonably far apart). Transitions shown are for the Hadamard (dotted blue lines), X (red lines) and Z (dotted green lines) gates. One of the states shown here is an eigenvector of the Hadamard (PETE) gate. What is the other?

As we may have mentioned before this is an intermediary extension of the original (pure) misty state formalism that we have been able to crystallize only recently and have yet to appreciate fully, namely that arbitrary embeddings of superposition states (see diagrams on previous two pages) lead to irreducible states, of which these are the simplest cases: $\{\bigcirc, \{\bigcirc, \bullet\}\}$ and $\{\bigcirc, \overline{\{\bigcirc, \bullet\}}\}$.

Notice that these two states are, in fact, the two eigenvectors of the Hadamard (PETE) gate. Even though these states stretch a little the syntax (and semantics) of the original misty state formalism it is easy to prove what we said: first, $\mathrm{H}(\{\bigcirc, \{\bigcirc, \bullet\}\}) = \{\bigcirc, \{\bigcirc, \bullet\}\}$, as can be seen in the Figure on page 36 (and below) and then:

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

Notice also that $\{\bigcirc, \{\bigcirc, \bullet\}\} = \frac{\sqrt{2+\sqrt{2}}}{2}|0\rangle + \frac{\sqrt{2-\sqrt{2}}}{2}|1\rangle$ has probability amplitudes that cannot result directly from calculations associated with those mentioned in the "squaring rule" (pp. 83-84 in the book). They are the result of the embedded mist. Furthermore, the state appears to be (and is) irreducible. This has several meaningful consequences, but first let's see how we motivate the transition to this new syntax from the initial formalism.

In other words, now that we know what the second eigenvector of the Hadamard gate is—what good is that?

Terry proposes[1] that we should start by getting students used to the simplified notation $\{a\bigcirc, b\newmoon\}$ where $a, b \in \mathbb{N}$ are integers representing the number of copies. Although not strictly necessary one could also emphasize that ultimately when we calculate probabilities if a, b share any common factors they can be cancelled, just like when reducing fractions to lowest form, because cancellation doesn't change the probability calculations for what we can actually observe.

We then ask the interesting question: "Is there any mist which passes through the PETE box unchanged?" At first glance the answer seems "obviously not!", because a PETE box does the evolution $\{a\bigcirc, b\newmoon\} \to \{(a+b)\bigcirc, (a-b)\newmoon\}$ and the equations

$$\begin{cases} a = a + b \\ b = a - b \end{cases}$$

do not have nontrivial solutions. So we need to change the question to: "Is there any mist which passes through the PETE box such that the probabilities of observing $\bigcirc$ or $\newmoon$ are unchanged?" For that we have to solve

$$\begin{cases} a^2 = \dfrac{(a+b)^2}{(a+b)^2 + (a-b)^2} \\ b^2 = \dfrac{(a-b)^2}{(a+b)^2 + (a-b)^2} \end{cases}$$

and the solution for this is that the ratio $\frac{a}{b}$ needs to be an irrational number. That is yet another aspect that makes these "irreducible" misty states interesting (besides the construction that resembles the Stern-Brocot tree). Tell the story of poor Hippasus.

This leads to the first eigenvector of the Hadamard gate.

[1]This is not in the book, it's from an email dated October 4, 2024.

Now consider the following well-formed (but irreducible) misty state:

{🍎, {🍉, 🍒}}

It says that three type of fruit are possible for a snack: apple, watermelon and cherries. The specific item that we end up with is determined probabilistically (via measurement). We do know it will be one of the three items listed. In this misty state the watermelon and cherries are in equal superposition with each other (let's call that state s_1) and the apple is in equal superposition with the state s_1. When we estimate the probability of each outcome we find that the probability of receiving an apple is $p(\text{🍎}) = \frac{1}{2}$ while for the other two items $p(\text{🍉}) = p(\text{🍒}) = \frac{1}{4}$ (so each has a probability of 0.25).

If we modify the misty state to {🍎, {🍎, 🍒}} we have a minimal non-classical situation. One would expect the probability of getting an apple to still be 0.75, when in effect it becomes 0.853 (thus lowering the chance of receiving cherries to 0.147). The reason, of course, is that the probability amplitudes add up, but the probabilities do not.

The resulting state is:

$$|\Psi\rangle = \frac{\sqrt{2+\sqrt{2}}}{2}|\text{🍎}\rangle + \frac{\sqrt{2-\sqrt{2}}}{2}|\text{🍒}\rangle = \cos\frac{\pi}{8}|\text{🍎}\rangle + \sin\frac{\pi}{8}|\text{🍒}\rangle$$

This is yet another situation where the transition from classical to quantum challenges "common-sense" expectations. We can model (produce) this state in Wolfram quantum framework as follows:

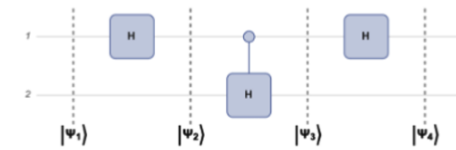


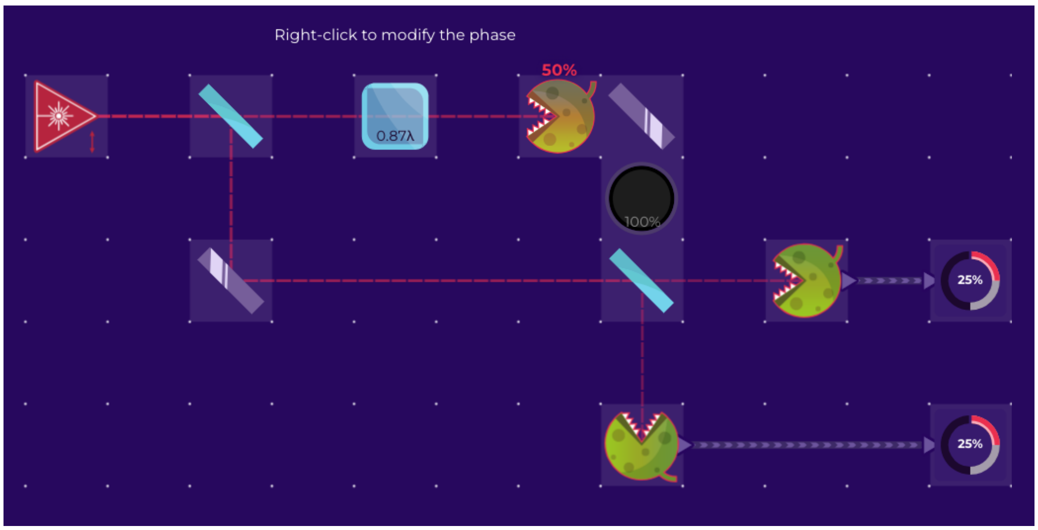
Figure 7: Quantum circuit that produces the Hadamard eigenvectors

We want to illustrate here how the state is generated. This will give a feel for how the quantum abacus (or, the misty state formalism) works. We start with $|\Psi_1\rangle = \bigcirc\bigcirc$ and after we apply the first Hadamard the state becomes $|\Psi_2\rangle = \{\bigcirc, \bullet\}\bigcirc = \{\bigcirc\bigcirc, \bullet\bigcirc\}$. With the first qubit as control we get $|\Psi_3\rangle = \{\bigcirc\bigcirc, \bullet\{\bigcirc, \bullet\}\}$. We then do a Hadamard on the first qubit and observe it. If we see $\bigcirc$ then we will have collapsed the second qubit to the desired state. Proof:

$$
\begin{aligned}
|\Psi_4\rangle &= \{\mathrm{H}(\bigcirc)\bigcirc, \mathrm{H}(\bullet)\{\bigcirc, \bullet\}\} \\
&= \{\{\bigcirc, \bullet\}\bigcirc, \{\bigcirc, \overline{\bullet}\}\{\bigcirc, \bullet\}\} \\
&= \{\{\bigcirc\bigcirc, \bigcirc\{\bigcirc, \bullet\}\}, \{\bullet\bigcirc, \overline{\bullet}\{\bigcirc, \bullet\}\} \\
&= \{\bigcirc\{\bigcirc, \{\bigcirc, \bullet\}\}, \bullet\{\bigcirc, \overline{\{\bigcirc, \bullet\}}\}\}.
\end{aligned}
$$

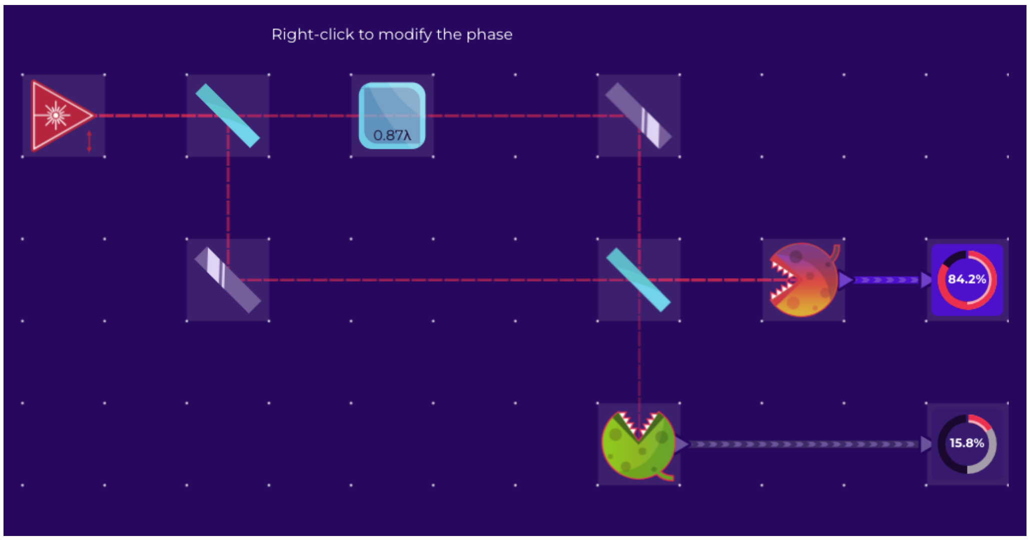
The simplest one qubit circuit that produces this state is a rotation $R_y(\frac{\pi}{4})$. But that gate is not part of the original misty state formalism. We can translate it into a seven gate circuit with two H, four S and one $R_z(-\frac{\pi}{4})$ gate then use the quantum synthesis algorithm **`gridsynth`** [62] to convert rotations into an optimal sequence of Clifford + T gates but that results into a circuit of infinite depth. As a matter of fact more can be seen in the derivation above: regardless of what we measure in the first qubit we obtain one (or the other) of the two eigenvectors of the Hadamard matrix in the second qubit.

Here's an experimental setup in Quantum Flytrap that tries to clarify what is happening in the quantum snack situation. First, as we mentioned a long time ago, we now have a delay on one of the branches. Furthermore, when we measure probabilities of occurrence on each branch we still obtain the classical values: 50%, 25% and 25%. If we know where the apple is coming from, then, nothing unusual happens and calculations are as expected (i.e., classical).



The second part of the experiment will bring up the "indistinguishability" argument of Scarani: we're going to let the paths come together. The horizontal direction means the snack is "apple" and the vertical: "cherries."

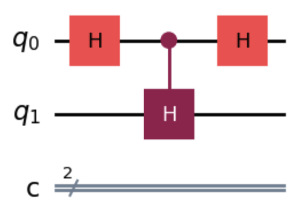
Now the paths are coming together so we no longer know where the apple is coming from: is it transmitted through the first beam splitter and then reflected by the second one, or is it reflected by the first beam splitter and then transmitted through the second one?



Since we can't know (and given the specific delay on the top branch) the probability of getting an apple changes: it increases to 85%.

Let's now address two things as we finalize this document. The first one is obtaining confirmation from Qiskit that what we think happens in that circuit does indeed happen as expected:

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




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

$$0.8535533906|00\rangle + 0.1464466094|01\rangle + \tfrac{\sqrt{2}}{4}|10\rangle - \tfrac{\sqrt{2}}{4}|11\rangle$$

For this we calculate:

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

This translates to:

$$\frac{1}{\sqrt{2}}|0\rangle\left(\frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}|1\rangle\right)\right)+\frac{1}{\sqrt{2}}|1\rangle\left(\frac{1}{\sqrt{2}}|0\rangle-\frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}|1\rangle\right)\right)$$

Which then becomes:

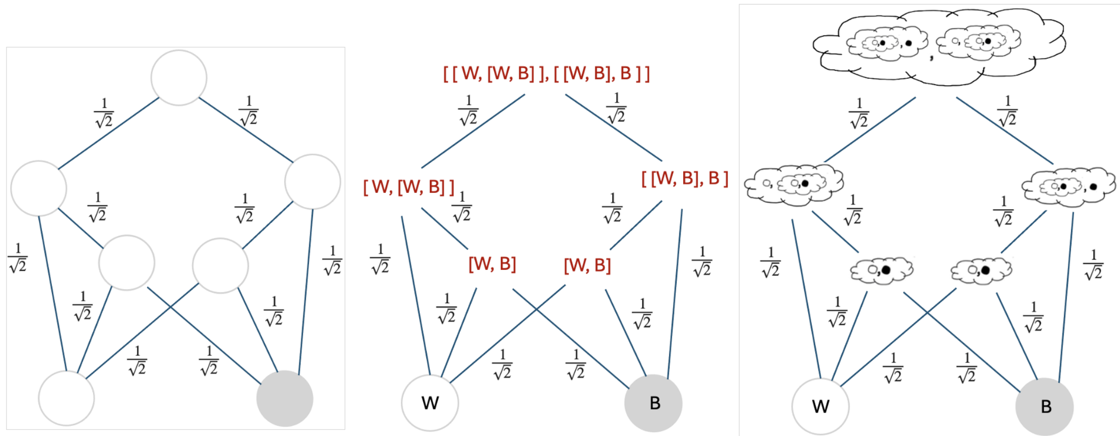
$$\frac{1}{2}|00\rangle + \frac{1}{2\sqrt{2}}\Big(|00\rangle + |01\rangle\Big) + \frac{1}{2}|10\rangle - \frac{1}{2\sqrt{2}}\Big(|10\rangle + |11\rangle\Big)$$

And finally:

$$\frac{1}{2}\left(1 + \frac{1}{\sqrt{2}}\right)|00\rangle + \frac{1}{2\sqrt{2}}|01\rangle + \frac{1}{2}\left(1 - \frac{1}{\sqrt{2}}\right)|10\rangle - \frac{1}{2\sqrt{2}}|11\rangle$$

Which is what Qiskit reports (modulo their ordering for qubits).

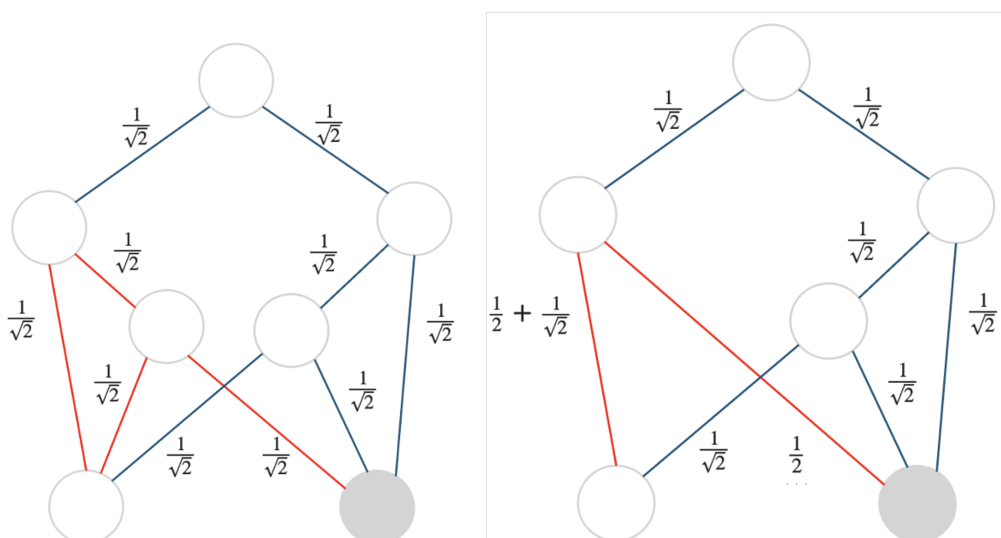
From here, how do we graduate to the full conventional mathematical apparatus? Let's simplify $\{\{\{\bigcirc, \bullet\}, \bullet\}, \{\bigcirc, \{\bigcirc, \bullet\}\}\}$ below:



This is what we're actually trying to calculate and simplify:

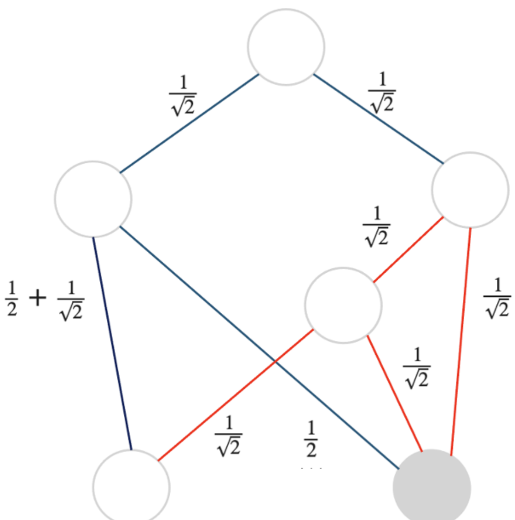
$$\frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}|1\rangle\right)+\frac{1}{\sqrt{2}}|1\rangle\right)+\frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}|1\rangle\right)\right)$$

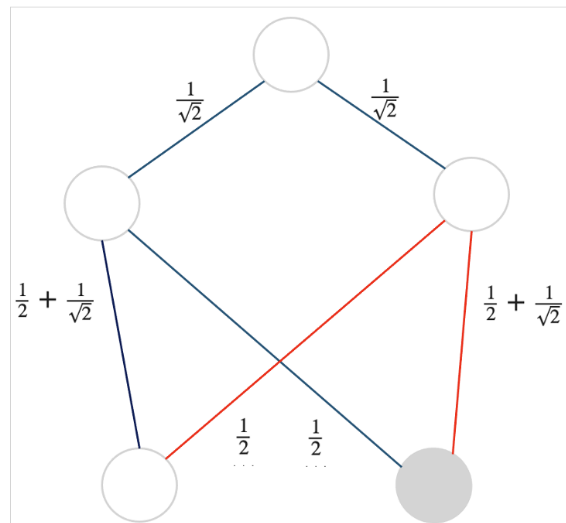
We apply these simple, pre-algebra reductions:



Notice in the process we eliminate nodes, weights get updated.

This particular expression is symmetrical:

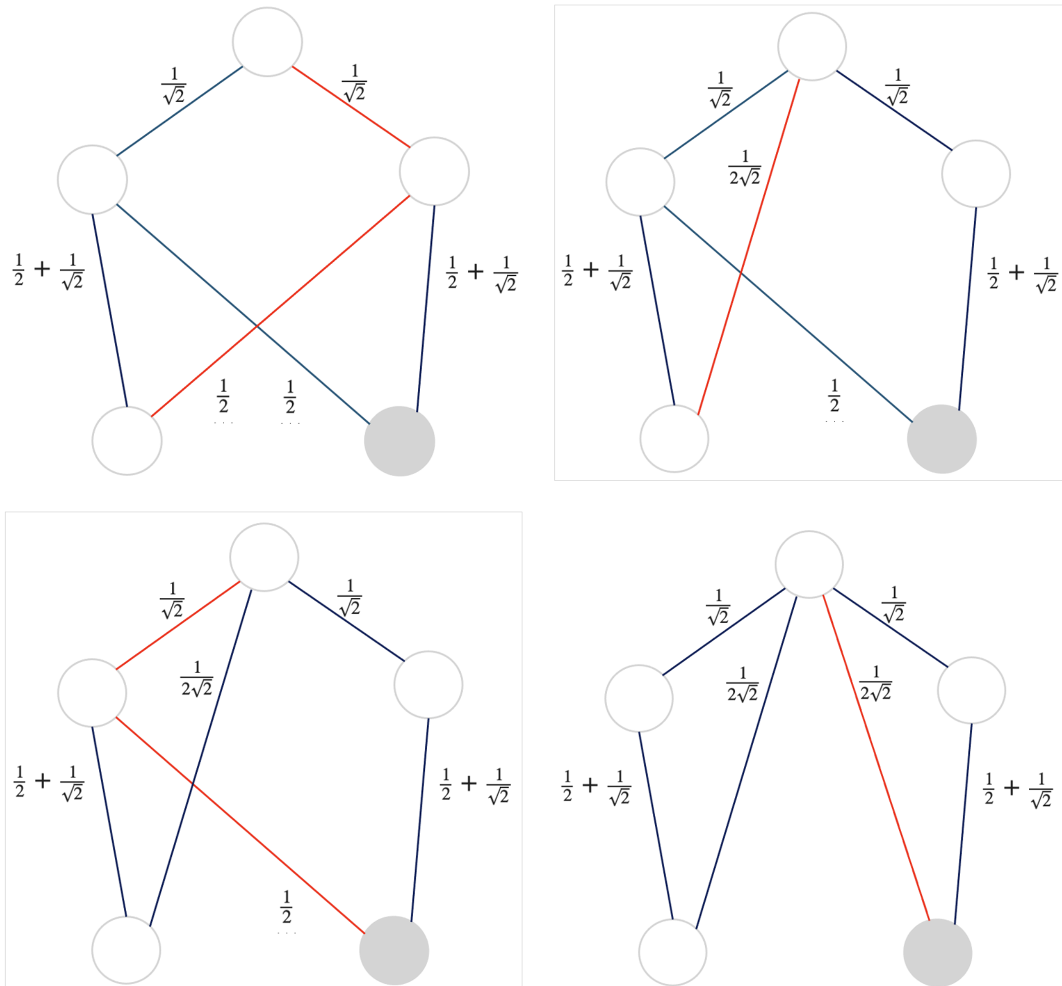




So what we do on the left we also do on the right[2].

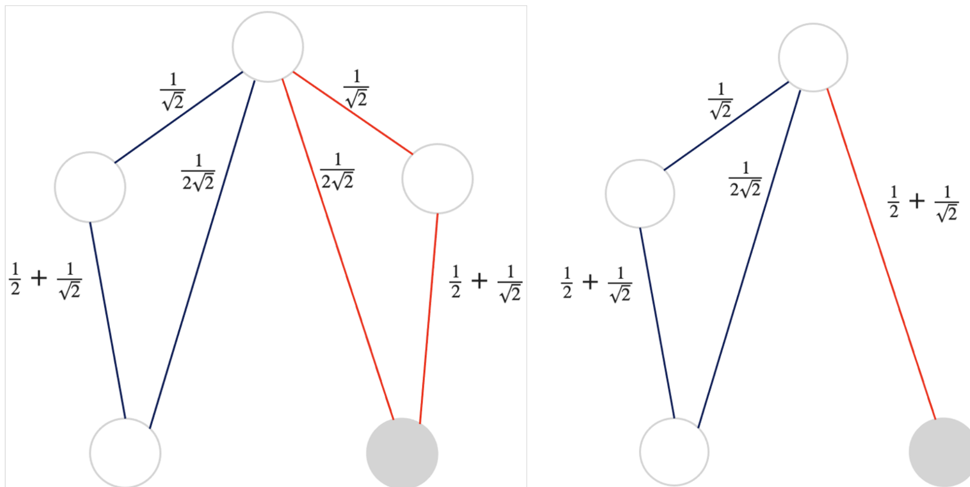
[2]Note that as the graph is being reduced the expression is getting reduced too. This process can be automated (and animated) and we have plans to do that in Mathematica soon. For now it is a good exercise to trace the evolution of the mathematical expression as the graph is being reduced (simplified). But I am not doing it here, in this draft.

We continue the process:

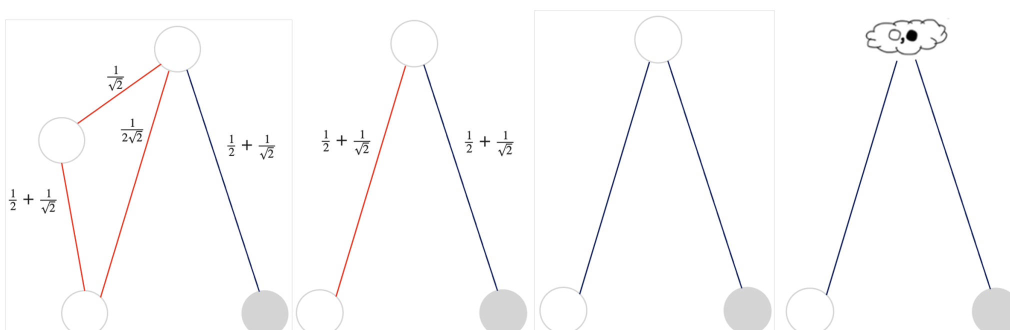


We are now left with two more nodes to eliminate in this graph (one creates a triangle on the left and the other one a triangle on the right).

We first proceed on the right side of the graph



Then the same thing on the left and we're finished:



This procedure is entirely general and despite the fact that the constituent states are irreducible in the original (pure) misty state formalism we were in fact able to prove that:

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

As I said at this point in the semester (or six weeks daily summer session) we have three different ways in which we discuss the GHZ game and then we're done with what we wanted to present in class.

After that we cover topics in general (e.g., error correction, Simon's algorithm, Shor, entanglement swapping, QKD etc.) or revisit class material that has already been presented but this time with linear algebra, matrices, etc. We look through Nielsen and Chuang, Rieffel and Polak, Mermin, the online book by Ekert and others (IBM Watrous videos) to see what we have learned and know already so we can then learn (or practice) the traditional mathematics on those topics.

This document ends here. I do not attach a bibliography at this time but these three papers are relevant to what was said above:

- the original[3] CS2023 knowledge unit proposal
- discussion/analysis[4] of QED-C data collected
- the very brief summary[5] included in CS2023

The first paper was presented at SIGCSE 2023 in Toronto, Canada. The second one (as a poster) to ITiCSE 2023 in Turku, Finland. The third is part of the official, final CS2023 report.

[3] https://dl.acm.org/doi/pdf/10.1145/3545945.3569845

[4] https://legacy.cs.indiana.edu/~dgerman/2023/curricular-maps/cs2023-quantum.pdf

[5] https://legacy.cs.indiana.edu/~dgerman/2024/abstract-reformatted-jan-5.pdf