In [1]:
```
%run utils.py
```

COMS 4281 - Intro to Quantum Computing

Problem Set 2, Quantum Info Basics

Due: October 9, 11:59pm

Collaboration is allowed and encouraged (teams of at most 3). Please read the syllabus carefully for the guidlines regarding collaboration. In particular, everyone must write their own solutions in their own words.

Name: Chandan Suri

UNI: CS4090

Late Days Consumed: 1, Late Days Left: 4 (Out of 5)

Write your collaborators here:

Collaborated with: AS6413

Problem 1: Non-standard Basis Measurements

a) Consider an orthonormal basis $B = \{|b_1\rangle, \dots, |b_d\rangle\}$ for $\mathbb{C}^d$. As we learned in class, measuring a quantum state $|\psi\rangle \in \mathbb{C}^d$ according to the basis B yields outcome $|b_j\rangle$ with probability $|\langle b_j|\psi\rangle|^2$.

In class we also learned that this process was equivalent to first applying a unitary U on $|\psi\rangle$, and then measuring the resulting state in the standard basis. In other words, the probability of obtaining standard basis outcome $|j\rangle$ when measuring $U|\psi\rangle$ in the standard basis, equal to $|\langle b_j|\psi\rangle|^2$. What unitary U accomplishes this? Give a description of U and prove that it works.

Solution

Considering the probability of measuring or observing the outcome $|b_j\rangle$ when $|\psi\rangle \in \mathbb{C}^d$ is being measured with respect to the basis B. Let's assume that we get the probability of getting the outcome $|b_j\rangle$ as $|\langle b_j|\psi\rangle|^2$ which we learnt in class.

Now, to find a unitary U which when applied on $|\psi\rangle$ gives the probability of obtaining standard basis outcome $|j\rangle$ when measuring $U|\psi\rangle$ in the standard basis, equal to $|\langle b_j|\psi\rangle|^2$, let's consider the j^{th} entry in the resulting vector. Essentially, this entry is given by the inner product of the j^{th} row of U and ψ. This gives us the following fact that the entry should be $\langle b_j|\psi\rangle$. For this entry to be equivalent to the same, we will need a unitary that basically defines the change of the basic matrix from B to the standard basis.

Thus, this unitary should be such that the j^{th} row of U will be the coefficients $\alpha_{i,j}$ when b_j can be written as the linear combination of the standard basis vectors which should be: $b_j = \sum_i \alpha_{i,j} |i\rangle$.

Following from the above description of U and the fact about the j_{th} entry will be $\langle b_j|\psi\rangle$ which is basically the same as $\sum_i \alpha_{i,j}\psi_i$. As we reach the same conclusion from both the ends, we know that measuring the same now will give us the probability of $|\langle b_j|\psi\rangle|^2$. The same can be arranged using a unitary which has the coefficients b_j following the formulation $b_j = \sum_i \alpha_{i,j} |i\rangle$.

Hence, we have proven the usage of the unitary U and that it suffices the scenario.

b) Now let's implement the unitary for measuring in the following basis B:

$$|\psi_0\rangle = \cos(\pi/8)|0\rangle + \sin(\pi/8)|1\rangle$$

and

$$|\psi_1\rangle = -\sin(\pi/8)|0\rangle + \cos(\pi/8)|1\rangle$$

First, write down the measurement probabilities if we measure the following states in the basis B:

$$|1\rangle, |-\rangle, |+\rangle, \cos(\pi/8)|0\rangle + \sin(\pi/8)|1\rangle$$

Solution

When considering the basis as:

$$|\psi_0\rangle = \cos(\pi/8)|0\rangle + \sin(\pi/8)|1\rangle$$

and

$$|\psi_1\rangle = -\sin(\pi/8)|0\rangle + \cos(\pi/8)|1\rangle$$

The measurement probabilities can be computed as follows:

When measuring $|1\rangle$:

We observe $|\psi_0\rangle$ with the probability $|\langle 1|\psi_0\rangle|^2$ which comes out to be $\sin^2(\pi/8)$

We observe $|\psi_1\rangle$ with the probability $|\langle 1|\psi_1\rangle|^2$ which comes out to be $\cos^2(\pi/8)$.

When measuring $|-\rangle$:

We observe $|\psi_0\rangle$ with the probability $|\langle -|\psi_0\rangle|^2$ which comes out to be $\frac{(\cos(\pi/8)-\sin(\pi/8))^2}{2}$.

We observe $|\psi_1\rangle$ with the probability $|\langle -|\psi_1\rangle|^2$ which comes out to be $\frac{(\cos(\pi/8)+\sin(\pi/8))^2}{2}$.

When measuring $|+\rangle$:

We observe $|\psi_0\rangle$ with the probability $|\langle +|\psi_0\rangle|^2$ which comes out to be $\frac{(\cos(\pi/8)+\sin(\pi/8))^2}{2}$.

We observe $|\psi_1\rangle$ with the probability $|\langle +|\psi_1\rangle|^2$ which comes out to be $\frac{(\cos(\pi/8)-\sin(\pi/8))^2}{2}$.

When measuring $\cos(\pi/8)\,|0\rangle + \sin(\pi/8)\,|1\rangle$:

We observe $|\psi_0\rangle$ with the probability

$$\left[\begin{pmatrix}\cos\pi/8 & \sin\pi/8\end{pmatrix}\begin{pmatrix}\cos\pi/8 \\ \sin\pi/8\end{pmatrix}\right]^2$$

which comes out to be 1.

We observe $|\psi_1\rangle$ with the probability

$$\left[\begin{pmatrix}\cos\pi/8 & \sin\pi/8\end{pmatrix}\begin{pmatrix}-\sin\pi/8 \\ \cos\pi/8\end{pmatrix}\right]^2$$

which comes out to be 0.

c) In the code below, write the matrix U that implements the change of basis from the standard basis to the basis above.

In [2]:

```
# ========= BEGIN CODE =================
U = [[np.cos(np.pi/8), np.sin(np.pi/8)],
    [-np.sin(np.pi/8), np.cos(np.pi/8)]]
# ========= END CODE =================

def perform_basis_measurement(initial_state: List[float]) -> QuantumCircuit:
    qr = QuantumRegister(1, name="input state")
    cr = ClassicalRegister(1, name="output")
    qc = QuantumCircuit(qr, cr)
    qc.initialize(initial_state)
    qc.append(UnitaryGate(U), qr)
    qc.measure(qr, cr)
    return qc
```

Now we'll test your basis change on some states and plot their measurement statistics. You should use this to check whether you implemented the right basis change U.

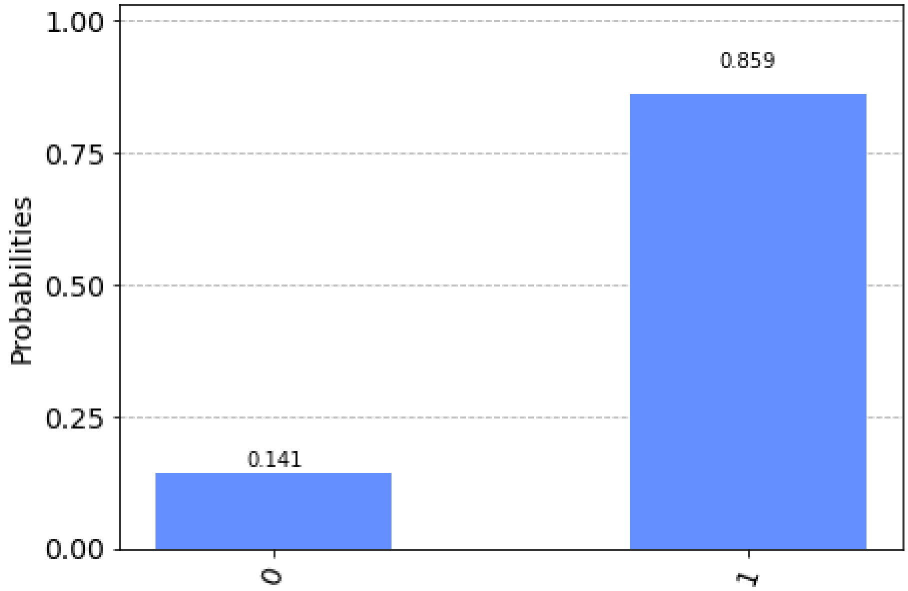
In [3]:

```
#First, we test it on the |1> state
qc1 = perform_basis_measurement([0.0, 1.0])
qc1.draw(output='mpl')
backend = Aer.get_backend('qasm_simulator')
job_sim = backend.run(transpile(qc1, backend), shots=5024)
```

```python
# Grab the results from the job.
result_sim = job_sim.result()
counts = result_sim.get_counts(qc1)
plot_histogram(counts)
```

```
<frozen importlib._bootstrap>:219: RuntimeWarning: scipy._lib.messagestream.Mess
ageStream size changed, may indicate binary incompatibility. Expected 56 from C
header, got 64 from PyObject
```

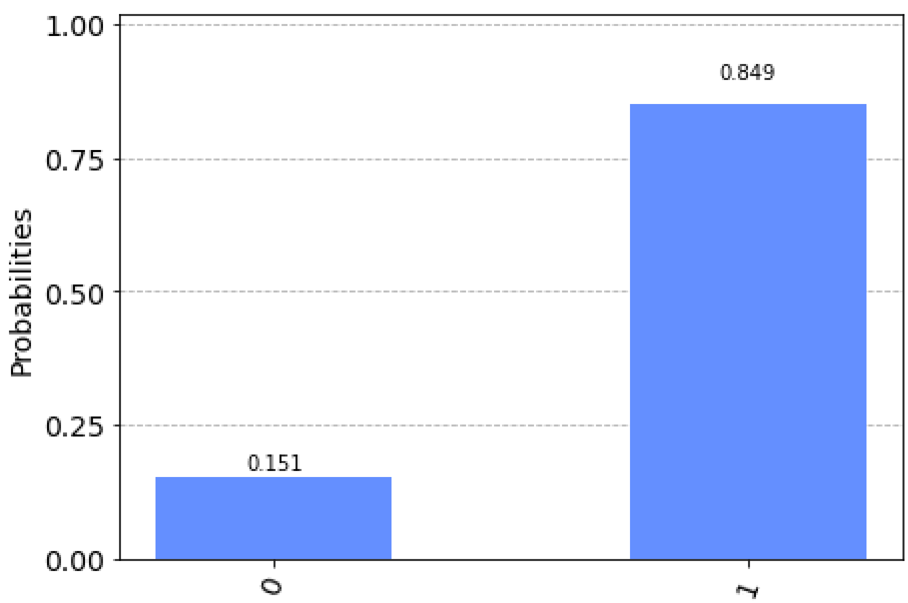
Out[3]:



In [4]:

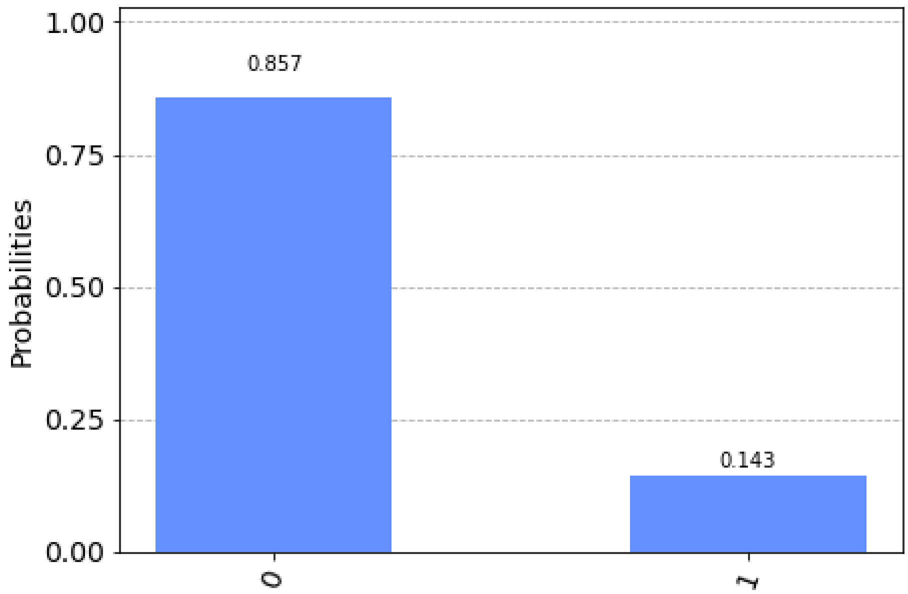
```python
# Next we try it on the |-> state
qc1 = perform_basis_measurement([1.0/np.sqrt(2), -1.0/np.sqrt(2)])
qc1.draw(output='mpl')
backend = Aer.get_backend('qasm_simulator')
job_sim = backend.run(transpile(qc1, backend), shots=5024)

# Grab the results from the job.
result_sim = job_sim.result()
counts = result_sim.get_counts(qc1)
plot_histogram(counts)
```

Out[4]:



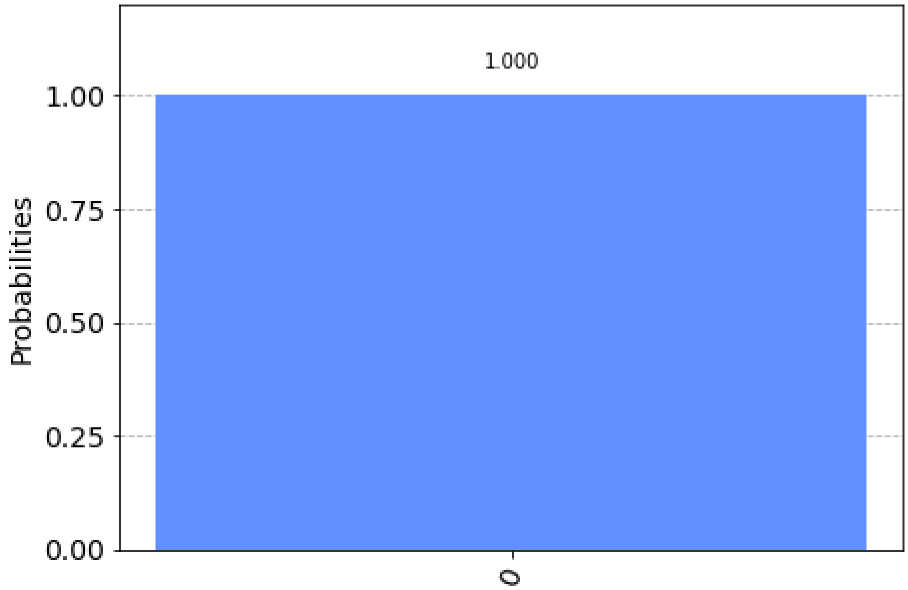
In [5]:
```
#...and the |+> state
qc1 = perform_basis_measurement([1.0/np.sqrt(2), 1.0/np.sqrt(2)])
qc1.draw(output='mpl')
backend = Aer.get_backend('qasm_simulator')
job_sim = backend.run(transpile(qc1, backend), shots=5024)

# Grab the results from the job.
result_sim = job_sim.result()
counts = result_sim.get_counts(qc1)
plot_histogram(counts)
```

Out[5]:



In [6]:
```
#and now the cos(pi/8) |0> + sin(pi/8) |1> state
qc1 = perform_basis_measurement([np.cos(math.pi/8), np.sin(math.pi/8)])
qc1.draw(output='mpl')
backend = Aer.get_backend('qasm_simulator')
job_sim = backend.run(transpile(qc1, backend), shots=5024)

# Grab the results from the job.
result_sim = job_sim.result()
counts = result_sim.get_counts(qc1)
plot_histogram(counts)
```

Out[6]:



Problem 2: EPR Pair Properties

Let's examine properties of the EPR pair

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)\,.$$

In what follows, let's suppose that Alice is given the left qubit of the EPR pair, and Bob is given the right qubit, and they are separated by a large distance.

a) Let $A = \{|a_1\rangle, |a_2\rangle\}$ be some orthonormal basis for $\mathbb{C}^2$. Suppose Alice measures her qubit using basis A. What are the statistics of the measurement outcomes (i.e. what are the probability of $|a_1\rangle$ or $|a_2\rangle$)?

Solution

For computing the statistics of the measurement outcomes, let:

$$|a_1\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

and

$$|a_2\rangle = \gamma\,|0\rangle + \delta\,|1\rangle$$

So, the probability of $|a_1\rangle$ can be computed as:

$$\frac{1}{\sqrt{2}} * \langle a_1|0\rangle\,|0\rangle + \frac{1}{\sqrt{2}} * \langle a_1|1\rangle\,|1\rangle$$

Computing this gives us:

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

This gives us the probability as:

$$\frac{\alpha^2}{2} + \frac{\beta^2}{2} = \frac{\alpha^2 + \beta^2}{2}$$

Since, $\alpha^2 + \beta^2 = 1$ being an orthonormal basis, we get the probability as $\frac{1}{2}$.

Further, the probability of $|a_2\rangle$ can be computed as:

$$\frac{1}{\sqrt{2}} * \langle a_2|0\rangle |0\rangle + \frac{1}{\sqrt{2}} * \langle a_2|1\rangle |1\rangle$$

Computing this gives us:

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

This gives us the probability as:

$$\frac{\gamma^2}{2} + \frac{\delta^2}{2} = \frac{\gamma^2 + \delta^2}{2}$$

Since, $\gamma^2 + \delta^2 = 1$ being an orthonormal basis, we get the probability as $\frac{1}{2}$.

b) Show that if Alice obtains measurement outcome $|a_i\rangle$ for some $i \in \{1, 2\}$, the post-measurement state of the EPR pair is $|a_i\rangle \otimes |a_i\rangle^*$ where $|a_i\rangle^*$ is the **complex conjugate** of $|a_i\rangle$ (i.e. the j-th entry is the complex conjugate of the j-th entry of $|a_i\rangle$).

This is interesting because Alice might have decided on the basis only after Bob was sent away, yet Alice's measurement causes Bob's qubit to instantaneously collapse into one of the basis states of A (up to complex conjugation). This is a phenomenon called **quantum steering**, because Alice is able to **steer** Bob's qubit, even though she is only acting on **her** qubit.

Solution

Since the measurement outcome of Alice is $|a_i\rangle$. As we know, the post measurement state will be computed as:

$$|a_i\rangle \otimes \frac{a}{||\,|a\rangle\,||}$$

where

$$a = a_i \otimes I\,|\psi\rangle$$

This gives us the following:

$$|a\rangle = \frac{1}{\sqrt{2}}(\langle a_i|0\rangle \otimes |0\rangle + \langle a_i|1\rangle \otimes |1\rangle)$$

Assuming a_i is a column vector with coefficients as α and β as we had assumed in the previous part and using the equation for $|a\rangle$ above, we get the following:

$$|a\rangle = \frac{1}{\sqrt{2}}(\bar{\alpha} \times |0\rangle + \bar{\beta} \times |1\rangle)$$

As we can see above, the term inside the braces is the complex conjugate of the $|a_i\rangle$ term. Thus, the term $|a\rangle$ becomes:

$$|a\rangle = \frac{a_i^*}{\sqrt{2}}$$

Also, the length of the vector is $\frac{1}{\sqrt{2}}$.

Therefore, the post measurement state formulation becomes:

$$|a_i\rangle \otimes \frac{\frac{a_i^*}{\sqrt{2}}}{1/\sqrt{2}}$$

Canceling out the terms gives us:

$$|a_i\rangle \otimes |a_i^*\rangle$$

Hence, Proved.

c) Suppose that Bob then measures his qubit using an orthonormal basis $B = \{|b_1\rangle, |b_2\rangle\}$. What are the statistics of his measurement outcomes, conditioned on Alice's outcome?

Solution

After Alice has measured her qubit which she measured as a_i, Bob's qubit, as it is entangled with that of Alice's qubit will be steered to the $|a_i^*\rangle$. This was the fact essentially proven above.

Now, when Bob measures his qubit with respect to the basis B, it will yield $|b_j\rangle$ with the probability $|\langle b_j|a_i^*\rangle|^2$.

This means that:

The probability of outcome $|b_1\rangle$ will be $|\langle b_1|a_i^*\rangle|^2$.

And, the probability of outcome $|b_2\rangle$ will be $|\langle b_2|a_i^*\rangle|^2$.

d) Suppose the order of measurements were reversed: Bob measures his qubit first using basis B, and then Alice measures her qubit using basis A. Show that the **joint** probability distribution of their measurement outcomes is the same as before.

Solution

When Alice measures first and then Bob measures, the joint probability distribution formulation will be as follows:

$$\frac{1}{2} * (|\langle b_j | a_i^* \rangle|^2)$$

(We got the first term from part a and second term from part c).

Now, when Bob measures first and then Alice measures, the joint probability distribution formulation can be computed as:

Following the similar steps as in the part a for Bob this time in the basis B we will get the same probability for the outcomes as $\frac{1}{2}$.

Next, when Alice performs the measurement after Bob this time, following similar steps as in part c, this will come out to be $|\langle a_i | b_j^* \rangle|^2$.

Thus, the joint probability formulation becomes:

$$\frac{1}{2} * (|\langle a_i | b_j^* \rangle|^2)$$

The second term in both of the above joint probability distribution formulations is just an inner product of the same vectors (one of them being the conjugate of the other) and the first term is the same fraction which thus, shows us that:

$$\frac{1}{2} * (|\langle b_j | a_i^* \rangle|^2) = \frac{1}{2} * (|\langle a_i | b_j^* \rangle|^2)$$

.

Therefore, it does not matter in which order the states are observed, the joint probability distribution over the observed states remains the same. Hence, proved.

e) What can you conclude about the effectiveness of using quantum entanglement and quantum steering as a method for faster-than-light communication? In other words, can Alice and Bob, by only making local measurements on their entangled state, send information to each other?

Solution

No, essentially using quantum entanglement and steering, Alice or Bob, given the measurement outcome, knows the state of the other qubit but they can neither control nor send the information to each other as being a faster-than-light communication. Since, the chnages in the qubit states are random, neither Alice knows the outcome nor does Bob. Considering one case here, when Alice knows the measurement outcome, she know about the state of Bob's qubit but Bob does not know the outcome and thus, cannot predict the state of his qubit. This clearly shows that this is not a faster-than-light communication as it's not instantaneous.

Problem 3: Quantum Teleportation with Noise

We saw how to teleport quantum states in class. Let's consider a twist on the standard teleportation protocol. Let's imagine that when Alice and Bob meet up to create an entangled state, the settings on their lab equipment was screwed up and they accidentally create the following two-qubit entangled state

$$|\theta\rangle = \frac{1}{\sqrt{3}}|00\rangle - \frac{1}{\sqrt{6}}|01\rangle + \frac{1}{\sqrt{6}}|10\rangle + \frac{1}{\sqrt{3}}|11\rangle \ .$$

Only Alice realizes this after they haven each taken a qubit each and gone their separate ways.

Suppose that Alice now gets a gift qubit $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$. Is there a way that she can still teleport $|\psi\rangle$ to Bob, using their corrupted entangled state $|\theta\rangle$ and the classical communication channel? Like in the standard teleportation protocol, Alice can only apply unitaries and measurements to her two qubits, and Bob will apply the same corrections as in the standard teleportation protocal (since he's not aware of the corruption).

a) Show how the teleportation protocol can be adapted for the corruption from Alice's side and analyze the correctness of your proposed protocol.

Solution

We have:

$$|\theta\rangle = \frac{1}{\sqrt{3}}|00\rangle - \frac{1}{\sqrt{6}}|01\rangle + \frac{1}{\sqrt{6}}|10\rangle + \frac{1}{\sqrt{3}}|11\rangle$$

This can reformulated as:

$$|\theta\rangle = \left(\frac{1}{\sqrt{3}}|0\rangle + \frac{1}{\sqrt{6}}|1\rangle\right) \otimes |0\rangle + \left(\frac{-1}{\sqrt{6}}|0\rangle + \frac{1}{\sqrt{3}}|1\rangle\right) \otimes |1\rangle$$

Next, we know that this entamgled state is noisy and we would need to do some of kind of rotation such that we do not get the terms $|01\rangle$ and $|10\rangle$. Let's do the same on the 1^{st} qubit of $|\theta\rangle$. Since, we do not know the exact rotation we would need to perform, let's try and use a rotation matrix with the parameter α. The rotation matrix would look something like:

$$\begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}$$

Furthermore, to perform the rotation, let's apply the same as $R \otimes I\,|\theta\rangle$. This gives us the following formulation:

$$R \otimes I\,|\theta\rangle = \left(\frac{1}{\sqrt{3}}\cos\alpha\,|0\rangle - \frac{1}{\sqrt{6}}\sin\alpha\,|1\rangle + \frac{1}{\sqrt{3}}\sin\alpha\,|1\rangle + \frac{1}{\sqrt{6}}\cos\alpha\,|1\rangle\right) \otimes |0\rangle + \left(-\frac{1}{\sqrt{6}}\mathrm{c}\right.$$

We can clearly see from the above formulation that to remove the noisy terms, we would need to set:

$$\frac{1}{\sqrt{3}}\sin\alpha + \frac{1}{\sqrt{6}}\cos\alpha = 0$$

From here, we get that:

$$\alpha = -\tan^{-1}\frac{1}{\sqrt{2}}$$

We also know that in the equation for $R \otimes I\,|\theta\rangle$ above, we should get:

$$\frac{1}{\sqrt{3}}\cos\alpha - \frac{1}{\sqrt{6}}\sin\alpha = \frac{1}{\sqrt{2}}$$

When we use the value of alpha in the equation above, this successfully satisfies the equation.

Therefore, this tells us that performing the unitary operation of rotation using the R matrix above on the 1^{st} qubit in $|\theta\rangle$ removes the noise and gives us the original EPR pair.

Post this, regular quantum teleportation can be performed!

b) Now let's implement Alice's teleportation protocol using the noisy EPR pair with qiskit.

Write code in `create_alice_noisy_tp_circuit` function below, which takes as as input a QuantumRegister (consisting of two qubits) and a ClassicalRegister (consisting of two 2 bits).

Important Note: the register indices in Alice's and Bob's functions are **local** (0-indexed), meaning that from Alice or Bob's point of view, her zeroth qubit is the gift qubit, and her first qubit is the first half of the EPR pair. From Bob's point of view, he only has the other half of the EPR pair, which he considers his zeroth qubit.

In [7]:

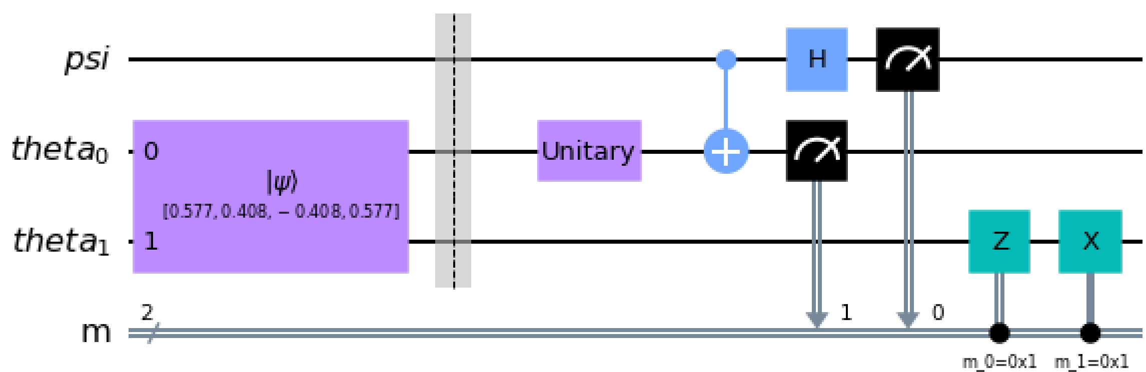
```
def initialize_noisy_epr_pair(qc: QuantumCircuit, qubits: List[int]) -> QuantumC
    # For qc.initialize, the ordering of the states are |00>, |01>, |10>, |11>
    #if the top wire corresponds to the rightmost bit (recall the little endian
    qc.initialize([np.sqrt(1/3.0), np.sqrt(1/6.0), -np.sqrt(1/6.0), np.sqrt(1/3.
    qc.barrier()
    return qc

def create_base_noisy_tp_circuit() -> QuantumCircuit:
    qr1 = QuantumRegister(1, name="psi")
    qr2 = QuantumRegister(2, name="theta")
    cr = ClassicalRegister(2, name="m")
    qc = QuantumCircuit(qr1, qr2, cr)
    return initialize_noisy_epr_pair(qc, [1, 2])

def create_alice_noisy_tp_circuit(qr: QuantumRegister, cr: ClassicalRegister) ->
    qc = QuantumCircuit(qr, cr)
    # Alice has two qubits (index 0,1) and access to two classical registers (in
    # ========= BEGIN CODE ================

    gamma = -np.arctan(1/np.sqrt(2))
```

```python
    U = [[np.cos(gamma), -np.sin(gamma)],
         [np.sin(gamma), np.cos(gamma)]]

    qc.append(UnitaryGate(U), [qr[1]])
    qc.cx(qr[0], qr[1])
    qc.h(qr[0])

    qc.measure(qr[0], cr[0])
    qc.measure(qr[1], cr[1])

    # ========= END CODE =================
    return qc

def create_bob_noisy_tp_circuit(qr: QuantumRegister, cr: ClassicalRegister) -> Q
    qc = QuantumCircuit(qr, cr)
    qc.z(0).c_if(cr[0], 1) # Apply gates if the registers
    qc.x(0).c_if(cr[1], 1) # are in the state '1'
    return qc
```

In [8]:

```python
noisy_tp_circuit = create_base_noisy_tp_circuit()
noisy_tp_circuit = append(noisy_tp_circuit, create_alice_noisy_tp_circuit, [0,1]
noisy_tp_circuit = append(noisy_tp_circuit, create_bob_noisy_tp_circuit, [2], [0
noisy_tp_circuit.draw(output='mpl')
```

Out[8]:



In [9]:

```python
test_noisy_teleportation(noisy_tp_circuit)
```

```
Testing noisy teleportation...
OK.
```

Problem 4: Transferring Entanglement

Here we explore a task to **transfer entanglement**. Let's say there are three parties, Alice, Bob, and Carol. Alice shares an EPR pair with Bob, and Bob shares an EPR pair with Carol (so Alice has one qubit, Bob has two qubits, and Carol has one qubit).

a) Design and analyze a protocol that involves only classical communication between the pairs (Alice,Bob), and (Bob,Carol), such that at the end Alice and Carol --- who never directly interacted with each other --- now share an EPR pair.

Hint: use the teleportation protocol as inspiration.

Solution

We have 4 bits in total here out of which the first 2 bits are entangled and the other 2 bits are entangled.

To design a protocol through which Alice and Carol can share an EPR pair without directly interacting with each other, we will need to do something with the bits that are with Bob.

Then, Bob will have to share the bits through the classicial registers so that the bit with Alice and the bit with Carol are entangled with each other.

The two EPR pairs can be represented as follows:

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Initially the state would be as follows:

$$|\psi_1\rangle \otimes |\psi_2\rangle = \frac{1}{2}(|0000\rangle + |0011\rangle + |1100\rangle + |1111\rangle)$$

If the qubits can be represented as ab_0, ab_1, bc_0, and bc_1.

Now, we will apply a CNOT between the 2nd and 3rd qubit above (Control: ab_1 and Target: bc_0). This gives us:

$$\frac{1}{\sqrt{2}}(|0000\rangle + |0011\rangle + |1110\rangle + |1101\rangle)$$

Now, we apply H gate on the 2nd qubit (ab_1).

$$\frac{1}{2}(|0000\rangle + |0100\rangle + |0011\rangle + |0111\rangle + |1010\rangle - |1110\rangle + |1001\rangle - |1101\rangle)$$

As we need to entangle the 1st and the 4th qubit together (ab_0 and bc_1), we will measure the 2nd and 3rd qubit and store it in the classical registers.

Further, considering the values in the classical registers, we will have to apply the X and Z gates on the 4th qubit (bc_1). The scenarios are shown below:

When have $|00\rangle$ as the 2nd and 3rd qubit, then the state (coefficients are not shown for simplicity) will be:

$$|00\rangle + |11\rangle$$

When have $|01\rangle$ as the 2nd and 3rd qubit, then the state will be:

$$|01\rangle + |10\rangle$$

When have $|10\rangle$ as the 2nd and 3rd qubit, then the state will be:

$$|00\rangle - |11\rangle$$

When have $|11\rangle$ as the 2nd and 3rd qubit, then the state will be:

$$|01\rangle - |10\rangle$$

For the cases above:

1st Case: Nothing has to be applied.

2nd Case: X gate has to be applied.

3rd Case: Z gate has to be applied.

4th Case: X and Z both have to be applied.

b) Now let's implement Alice's, Bob's and Carol's parts of the entanglement swapping circuit. You will have to implement what Alice, Bob, and Carol do with their qubits, and how they classically communicate with each other. Fill in the functions in the places indicated below.

Important note: see the important note in Problem 3 regarding the local indexing of qubits in the Alice, Bob and Carol functions.

In [10]:

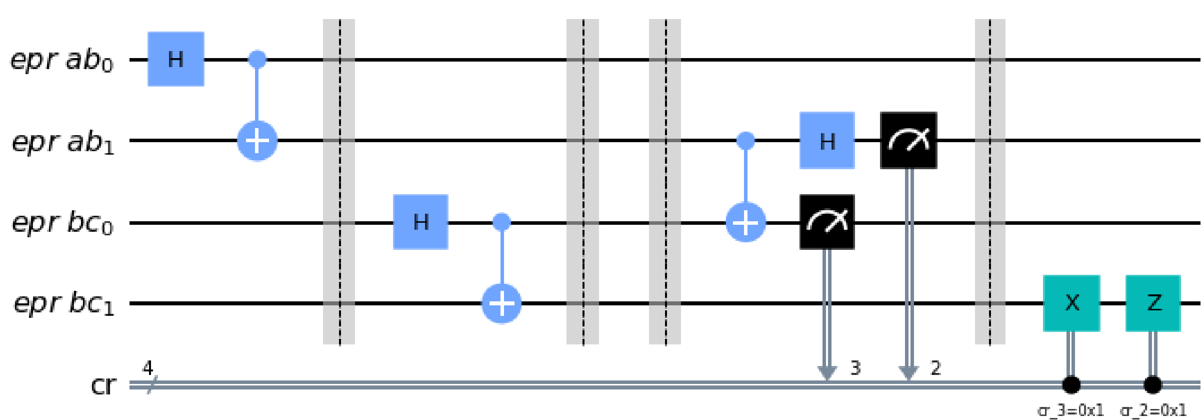
```
def alice_circuit(qr: QuantumRegister, cr: ClassicalRegister) -> QuantumCircuit:
    qc = QuantumCircuit(qr, cr)
    # Alice has one qubit (index 0) and access to two classical registers (index
    # ========= BEGIN CODE =================
    # ========= END CODE =================
    return qc

def bob_circuit(qr: QuantumRegister, cr: ClassicalRegister) -> QuantumCircuit:
    qc = QuantumCircuit(qr, cr)
    # Bob has two qubits (index 0,1) and access to four classical registers (ind
    # ========= BEGIN CODE =================

    qc.cx(qr[0], qr[1])
    qc.h(qr[0])
    qc.measure(qr[0], cr[2])
    qc.measure(qr[1], cr[3])

    # ========= END CODE =================
    return qc

def carol_circuit(qr: QuantumRegister, cr: ClassicalRegister) -> QuantumCircuit:
    qc = QuantumCircuit(qr, cr)
    # Carol has one qubit (index 0) and access to two classical registers (index
    # look up qiskit documentation for using classical registers to control quan
    # ========= BEGIN CODE =================

    qc.x(qr[0]).c_if(cr[1], 1)
    qc.z(qr[0]).c_if(cr[0], 1)

    # ========= END CODE =================
    return qc
```

```python
def add_epr_pair(qc: QuantumCircuit, a, b):
    qc.h(a)
    qc.cnot(a,b)
    qc.barrier()
    return qc

def create_entanglement_swapping_circuit_base() -> QuantumCircuit:
    """
    This creates a circuit with 2 EPR pairs in registers {0, 1} and {2, 3} respe
    and four classical registers (labelled {0,1,2,3}).
    Alice will have access to qubit 0, and the first two classical registers ({0
    Bob will have access to qubits 1 and 2, and all the classical registers ({0,
    Carol will have access to qubit 3, and the last two classical registers ({2,
    """
    qr1 = QuantumRegister(2, name="epr ab")
    qr2 = QuantumRegister(2, name="epr bc")

    num_classical_bits = 4

    cr = ClassicalRegister(num_classical_bits, name="cr")
    global_circuit = QuantumCircuit(qr1, qr2, cr)
    global_circuit = add_epr_pair(global_circuit, 0, 1)
    global_circuit = add_epr_pair(global_circuit, 2, 3)
    return global_circuit

def create_entanglement_swapping_circuit() -> QuantumCircuit:
    qc = create_entanglement_swapping_circuit_base()

    qc = append(qc, alice_circuit, [0],[0,1])
    qc.barrier()
    qc = append(qc, bob_circuit, [1, 2], [0, 1,2,3])
    qc.barrier()
    qc = append(qc, carol_circuit, [3], [2, 3])

    return qc
```

In [11]:

```python
entanglement_swapping_circuit = create_entanglement_swapping_circuit()
entanglement_swapping_circuit.draw(output = 'mpl')
```

Out[11]:



In [12]:

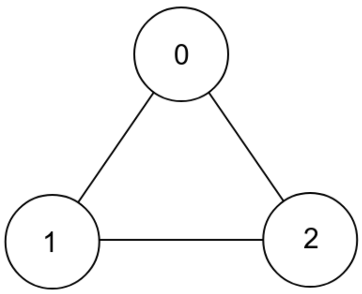
```python
test_entanglement_swapping(entanglement_swapping_circuit)
```

```
Testing entanglement swapping...
OK.
```

Problem 5: Let's Play a (Nonlocal) Game

In class we learned about the CHSH game, let's consider another, slightly more complicated game.

Let's say that you and your best friend want to pull a nasty prank on your mortal enemy*. You decide to try to convince him that the vertices in the following graph can be colored red or blue such that no two adjacent vertices have the same color (this is known as being 2-colorable):



Clearly, this isn't possible, but you decide to give it a shot. You propose the following non-local game to your enemy:

1. Your enemy picks a vertex s in the graph (0, 1, or 2) uniformly at random.
2. Your enemy gives you s.
3. Your enemy gives your friend either s or $s+1 \mod 3$, with 50% probability (call this vertex t).
4. You and your friend return colors (red or blue).
5. If the vertex your enemy gave you and your friend were the same vertex, he checks that the colors are the same. Otherwise he checks that the colors are different.

(The enemy is the referee of this game, and you and your friend are like Alice and Bob in CHSH)

a) What is the best probability that any classical strategy can win this game with?

Solution

The best probability that any classical strategy can win this game is $\frac{1}{2}$.

Let's take one of the strategies as when the enemy assigns any of the vertices to me and my friend.

Let's call one friend Alice and another friend Bob.

Alice has 3 possible vertex assignments and the same for Bob. The total number of possible combinations of vertex choices are $3 * 3 = 9$.

Now, for each of these 9 combinations, there will be 2 cases of color choices for each one of them such that either Alice and Bob answer Red or Blue. This in turn would mean that either Alice and Bob can answer the same color or different color. Thus, the total number of combinations become $9 * 2 = 18$.

Alice	Bob	Same	Diff
0	0	S -> Win	D
0	1	S	D -> Win
0	2	S	D -> Win
1	0	S	D -> Win
1	1	S -> Win	D
1	2	S	D -> Win
2	0	S	D -> Win
2	1	S	D -> Win
2	2	S -> Win	D

So, out of these 18 cases, now we want to know for what all cases they would win the game such that when the vertex given is the same, they should answer the same color or when the vertex given is dfferent they should answer different colors. This would give us 9 winning cases out of those 18. Also, however they answer, this is going to be the same.

Thus, for any classical strategy the winning probability will be $\frac{1}{2}$.

b) Let's say you and your friend are resourceful and happened to share a single EPR pair before playing the non-local game with your enemy. Fill in the following functions that take in a quantum register (initialized to an EPR pair), and a question (from 0, 1, 2), and perform a measurement that plays the game (outputting the result to the classical register). Try to get the best winning probability you can.

Hint: use the CHSH strategy for inspiration.

In [13]:
```
def alice_game_circuit(qr: QuantumRegister, cr: ClassicalRegister, question: int
    qc = QuantumCircuit(qr, cr)
    # Alice will need to apply a unitary gate, and then measure her qubit
    # ========= BEGIN CODE =================

    if question == 0:
        angle = 0
    elif question == 1:
        angle = np.pi/6
    else:
        angle = np.pi/3

    U = [[np.cos(angle), np.sin(angle)],
        [-np.sin(angle), np.cos(angle)]]
    qc.append(UnitaryGate(U), qr)
    qc.measure(qr[0], cr[0])
```

```python
    # ========= END CODE =================
    return qc

def bob_game_circuit(qr: QuantumRegister, cr: ClassicalRegister, question: int)
    qc = QuantumCircuit(qr, cr)
    # Bob will need to apply a unitary gate, and then measure his qubit
    # ========= BEGIN CODE =================

    if question == 0:
        angle = np.pi/12
    elif question == 1:
        angle = np.pi/4
    else:
        angle = (5*np.pi)/12

    U = [[np.cos(angle), np.sin(angle)],
        [-np.sin(angle), np.cos(angle)]]
    qc.append(UnitaryGate(U), qr)
    qc.measure(qr[0], cr[0])

    # ========= END CODE =================
    return qc
```

In [14]:

```python
def play_game(question1: int, question2: int) -> float:
    hidden_state = QuantumRegister(2, name="epr_pair")
    answers = ClassicalRegister(2, name="answer")
    global_circuit = QuantumCircuit(hidden_state, answers)
    global_circuit = add_epr_pair(global_circuit, 0, 1)
    global_circuit = append(global_circuit, lambda qr, cr : alice_game_circuit(q
    global_circuit = append(global_circuit, lambda qr, cr : bob_game_circuit(qr,
    total_shots = 5024
    backend = Aer.get_backend('qasm_simulator')
    job_sim = backend.run(transpile(global_circuit, backend), shots=total_shots)
    result_sim = job_sim.result()
    measurements = result_sim.get_counts(global_circuit)
    winning_shots = 0
    if question1 == question2:
        for measurement in measurements:
            if measurement[0] == measurement[1]:
                # Win this game
                winning_shots += measurements[measurement]
    else:
        for measurement in measurements:
            if measurement[0] != measurement[1]:
                # Win this game
                winning_shots += measurements[measurement]
    return winning_shots / total_shots

winning_probability = 0.0
for i in range(3):
    winning_probability += play_game(i, i)
    winning_probability += play_game(i, (i + 1) % 3)
print("Average Winning Probability: ", winning_probability / 6)
```

```
Average Winning Probability:  0.7196788747346071
```

c) Describe the strategy that you chose and it's expected winning probability.

Solution

The strategy is explained as below: We take six basis corresponding to each of the players (Alice or Bob) and also corresponding to each question (0, 1 or 2). These basis can be denoted by $a_{i,j}$ and $b_{i,j}$ for each of Alice and Bob. First index corresponds to the question asked and second one corresponds to the basis vector corresponding to the question. Simply, $a_{i,j}$ is the j_{th} basis vector for Alice when she observes or gets the i_{th} question.

Let me explain the same with an example. If Alice measures her first qubit when she has got the 0_{th} question, she will get either $a_{0,0}$ or $a_{0,1}$. Since, there's is a 50% chance for each of the colors chosen, both of these have an equal probability.

Now, if Alice observes $a_{0,0}$ and Bob's question was 0 as well, his probability of measuring $b_{0,0}$ will be $|\langle b_{0,0}|a^*_{0,0}\rangle|^2$ considering the change in basis. For this case, as they have got the same question (0_{th}), they would need to answer the same color in order to win the game. Let's say if Alice choose a color say Red, then Bob will also have to say Red when he leasures $b_{0,0}$. This, in turn, means that we want to maximize $|\langle b_{0,0}|a^*_{0,0}\rangle|^2$. Conclusively, this means that these two vectors should be the same but at the same time we also want to minimize the probability of the cases when they would lose in this scenario. Now, let's consider the case when Alice got the 0_{th} question and observed $a_{0,0}$ but this time Bob's question was 1. Let's say Alice says Red, then Bob needs to answer Blue to win the game. However, if Bob answers Red as well and measures $b_{1,1}$, we will have to minimize the $|\langle b_{1,1}|a^*_{0,0}\rangle|^2$. This shows us changing the basis of the states accordingly will be the strategy. To do the same, we will have basis for each question for both Alice and Bob. One of the vectors for the basis will correspond to the red color and the other one will correspond to the blue color.

The basis vectors are shown below:

Alice:

Question 0: (Mapped to Red and Blue respectively.)

$$\cos(0)\,|0\rangle + \sin(0)\,|1\rangle\,, -\sin(0)\,|0\rangle + \cos(0)\,|1\rangle$$

Question 1: (Mapped to Blue and Red respectively.)

$$\cos\left(\frac{\pi}{6}\right)|0\rangle + \sin\left(\frac{\pi}{6}\right)|1\rangle\,, -\sin\left(\frac{\pi}{6}\right)|0\rangle + \cos\left(\frac{\pi}{6}\right)|1\rangle$$

Question 2: (Mapped to Red and Blue respectively.)

$$\cos\left(\frac{\pi}{3}\right)|0\rangle + \sin\left(\frac{\pi}{3}\right)|1\rangle\,, -\sin\left(\frac{\pi}{3}\right)|0\rangle + \cos\left(\frac{\pi}{3}\right)|1\rangle$$

Bob:

Question 0: (Mapped to Red and Blue respectively.)

$$\cos\left(\frac{\pi}{12}\right)|0\rangle + \sin\left(\frac{\pi}{12}\right)|1\rangle\,, -\sin\left(\frac{\pi}{12}\right)|0\rangle + \cos\left(\frac{\pi}{12}\right)|1\rangle$$

Question 1: (Mapped to Blue and Red respectively.)

$$\cos\left(\frac{\pi}{4}\right)|0\rangle + \sin\left(\frac{\pi}{4}\right)|1\rangle\,, -\sin\left(\frac{\pi}{4}\right)|0\rangle + \cos\left(\frac{\pi}{4}\right)|1\rangle$$

Question 2: (Mapped to Red and Blue respectively.)

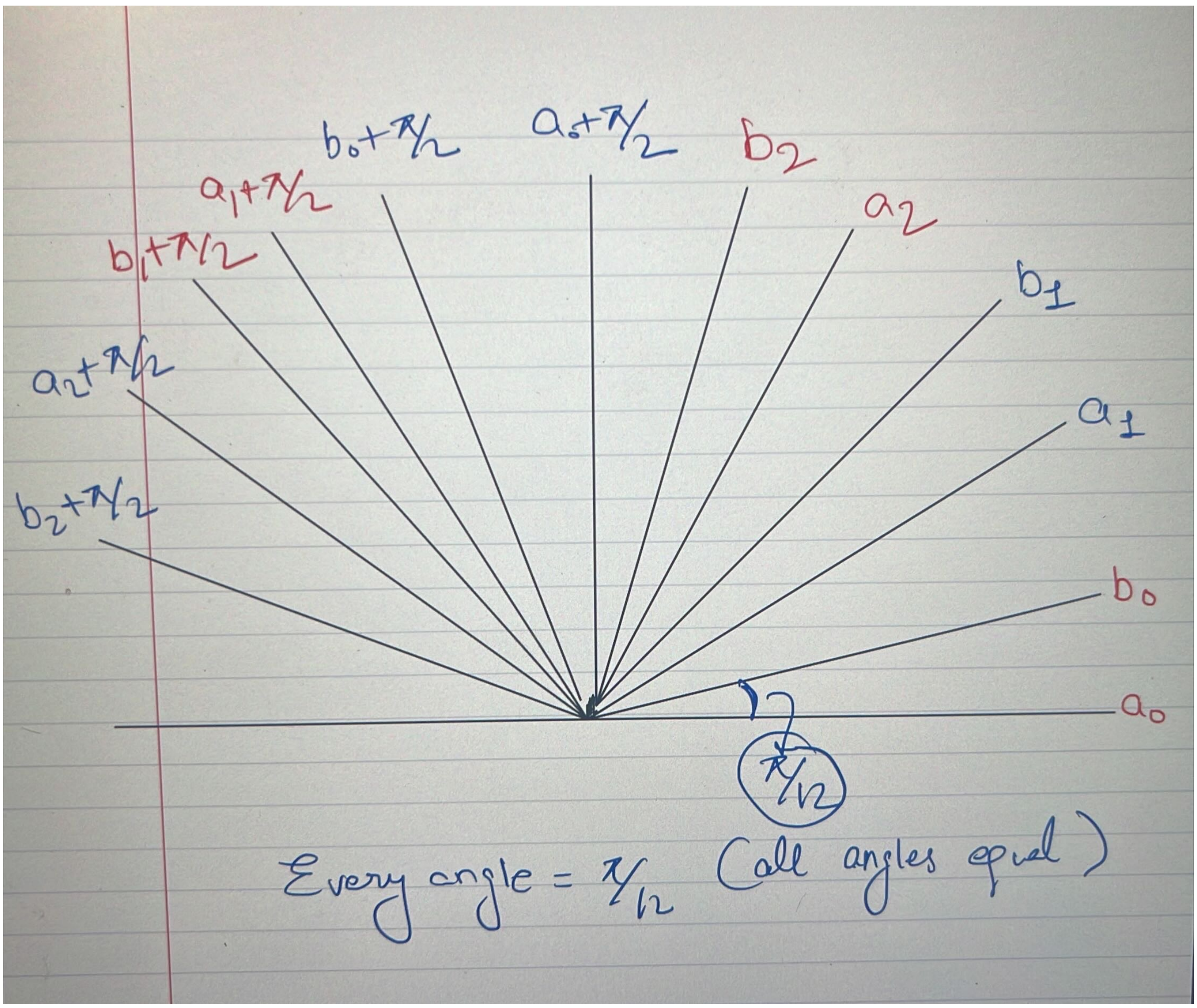
$$\cos\left(\frac{5\pi}{12}\right)|0\rangle + \sin\left(\frac{5\pi}{12}\right)|1\rangle\,, -\sin\left(\frac{5\pi}{12}\right)|0\rangle + \cos\left(\frac{5\pi}{12}\right)|1\rangle$$

This is also demonstrated in the picture below:

In [2]:

```python
from IPython.display import Image
Image("Problem5Image.jpeg")
```

Out[2]:



BONUS PROBLEM If you think you have the optimal quantum strategy for this game, give a proof that there is no better quantum strategy. You may assume Alice and Bob use 1 EPR pair as their shared state.

Many extra points if you give a proof that considers all possible quantum strategies (any entangled state, any possible measurements),

In [ ]:

```
In [ ]:
```