A1-A5

June 13, 2020

1

1.1 Identity or Pauli X?

Input: An operation that implements a single-qubit unitary transformation: either the identity (**I** gate) or the Pauli X gate (**X** gate). The operation will have Adjoint and Controlled variants defined.

Output: 0 if the given operation is the **I** gate, 1 if the given operation is the **X** gate.

You are allowed to apply the given operation and its adjoint/controlled variants exactly **once**.

Solution

The only way to extract information out of a quantum system is measurement. Measurements give us information about the states of a system, so to get information about the gate, we need to find a way to convert it into information about a state. If we want to distinguish two gates, we need to figure out to prepare a state and perform a measurement on it that will give us a result that we can interpret. To do this, we'll need to find a qubit state that, by applying to it I gate or X gate, will be transformed into states that can be distinguished using measurement, i.e., orthogonal states. Let's find such state.

As a reminder, here are the matrices that correspond to the given gates:

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Consider the effects of these gates on the basis states:

$$I|0\rangle = |0\rangle, I|1\rangle = |1\rangle$$
$$X|0\rangle = |1\rangle, X|1\rangle = |0\rangle$$

We see that the **I** gate leaves the $|0\rangle$ state unchanged, and the **X** gate transforms it into the $|1\rangle$ state. So the easiest thing to do is to prepare a $|0\rangle$ state, apply the given unitary to it and measure the resulting state in the computational basis: * If the measurement result is `Zero`, the measured state was $|0\rangle$, and we know the unitary applied to it was the **I** gate. * If the measurement result is `One`, the measured state was $|1\rangle$, and we know the unitary applied to it was the **X** gate.

> In Q#, the freshly allocated qubits start in the $|0\rangle$ state, so you don't need to do anything to prepare the necessary state before applying the unitary to it. You also have to return the qubits you allocated to the $|0\rangle$ state before releasing them. You can do that by measuring the qubit using the `M` operation and applying the **X** gate if it was measured in the $|1\rangle$ state, or you can use `MResetZ` operation that wraps this measurement and fixup into one operation.

```
In [ ]: open Microsoft.Quantum.Measurement;

        operation DistinguishIfromX (unitary : (Qubit => Unit is Adj+Ctl)) : Int {
            using (q = Qubit()) {
                unitary(q);
                return MResetZ(q) == Zero ? 0 | 1;
            }
        }
```

1.2 Identity or Pauli Z?

Input: An operation that implements a single-qubit unitary transformation: either the identity (**I** gate) or the Pauli Z gate (**Z** gate). The operation will have Adjoint and Controlled variants defined.

Output: 0 if the given operation is the **I** gate, 1 if the given operation is the **Z** gate.

You are allowed to apply the given operation and its adjoint/controlled variants exactly **once**.

Solution

As a reminder, $Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

We won't be able to distinguish **I** from **Z** by applying them to the basis states, since they both leave the $|0\rangle$ state unchanged and add a phase to the $|1\rangle$ state:

$$I|0\rangle = |0\rangle, I|1\rangle = |1\rangle$$
$$Z|0\rangle = |0\rangle, Z|1\rangle = -|1\rangle$$

However, if we try applying these gates to a superposition of basis states, we'll start seeing a difference between the resulting states:

$$I(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$
$$Z(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)) = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

These two states are orthogonal and can be distinguished by measuring them in the $\{|+\rangle, |-\rangle\}$ basis using `MResetX` operation (which is equivalent to applying an **H** gate and measuring in the computational basis).

> The task of distinguishing these two states is covered in more detail in the Measurements kata and the corresponding workbook.

```
In [ ]: open Microsoft.Quantum.Measurement;

        operation DistinguishIfromZ (unitary : (Qubit => Unit is Adj+Ctl)) : Int {
            using (q = Qubit()) {
                H(q);
                unitary(q);
                return MResetX(q) == Zero ? 0 | 1;
            }
        }
```

1.3 Z or S?

Input: An operation that implements a single-qubit unitary transformation: either the **Z** gate or the **S** gate. The operation will have Adjoint and Controlled variants defined.

Output: 0 if the given operation is the **Z** gate, 1 if the given operation is the **S** gate.

You are allowed to apply the given operation and its adjoint/controlled variants at most **twice**.

Solution

As a reminder, $S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$

This task differs from the previous two in that it allows you to apply the given unitary **twice**. Let's treat this as a hint that it is, and check how the given gates looks when applied twice. If you square the corresponding matrices (which is quite simple to do for diagonal matrices), you'll get

$$Z^2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I, S^2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = Z$$

This means that the task of identifying the *square* of the given unitary transformation is the same as distinguishing **I** from **Z** gates - and that's exactly the Section **??**!

```
In [ ]: open Microsoft.Quantum.Measurement;

        operation DistinguishZfromS (unitary : (Qubit => Unit is Adj+Ctl)) : Int {
            using (q = Qubit()) {
                H(q);
                unitary(q);
                unitary(q);
                return MResetX(q) == Zero ? 0 | 1;
            }
        }
```

1.4 Z or −Z?

Input: An operation that implements a single-qubit unitary transformation: either the **Z** gate or the minus **Z** gate (i.e., the gate $-|0\rangle\langle 0| + |1\rangle\langle 1|$). The operation will have Adjoint and Controlled variants defined.

Output: 0 if the given operation is the **Z** gate, 1 if the given operation is the $-$**Z** gate.

You are allowed to apply the given operation and its adjoint/controlled variants exactly **once**.

Solution

This task is more interesting: the given gates differ by a global phase they introduce (i.e., each of them is a multiple of the other), and the results of applying them to any single-qubit state are going to be indistinguishable by any measurement you can devise.

Fortunately, we are given not just the unitary itself, but also its controlled variant, i.e., the gate which applies the given unitary if the control qubit or qubits are in the $|1\rangle$ state, and does nothing if they are in the $|0\rangle$ state. This allows us to use so called "phase kickback" trick, in which applying a controlled version of a gate allows us to observe the phase introduced by this gate on the control qubit. Indeed,

State	Controlled Z	Controlled $-Z$
$\|00\rangle$	$\|00\rangle$	$\|00\rangle$
$\|01\rangle$	$\|01\rangle$	$\|01\rangle$
$\|10\rangle$	$\|10\rangle$	$-\|10\rangle$
$\|11\rangle$	$-\|11\rangle$	$\|11\rangle$

We see that both controlled gates don't modify the states with the control qubit in the $|0\rangle$ state, but if the control qubit is in the $|1\rangle$ state, they introduce a -1 phase to different basis states. We can take advantage of this if we apply the controlled gate to a state in which the *control qubit* is in superposition, such as $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle$:

$$\text{Controlled Z}\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle$$

$$\text{Controlled } -\text{Z}\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \otimes |0\rangle$$

After this we can measure the first qubit to distinguish $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ from $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, like we did in Section **??**.

> In Q# we can express controlled version of a gate using Controlled functor: the first argument of the resulting gate will be an array of control qubits, and the second one - the arguments of the original gate (in this case just the target qubit).

```
In [ ]: open Microsoft.Quantum.Measurement;

        operation DistinguishZfromMinusZ (unitary : (Qubit => Unit is Adj+Ctl)) : Int {
            using (qs = Qubit[2]) {
                // prep (|0 + |1)  |0
                H(qs[0]);

                Controlled unitary(qs[0..0], qs[1]);

                return MResetX(qs[0]) == Zero ? 0 | 1;
            }
        }
```

1.5 $I \otimes X$ or CNOT?

Input: An operation that implements a single-qubit unitary transformation: either the $I \otimes X$ (the **X** gate applied to the second qubit) or the **CNOT** gate with the first qubit as control and the second qubit as target. * The operation will accept an array of qubits as input, but it will fail if the array is empty or has one or more than two qubits. * The operation will have Adjoint and Controlled variants defined.

Output: 0 if the given operation is $I \otimes X$, 1 if the given operation is the **CNOT** gate.

You are allowed to apply the given operation and its adjoint/controlled variants exactly **once**.

Solution

Let's consider the effect of these gates on the basis states:

State	$I \otimes X$	**CNOT**
$\|00\rangle$	$\|01\rangle$	$\|00\rangle$
$\|01\rangle$	$\|00\rangle$	$\|01\rangle$
$\|10\rangle$	$\|11\rangle$	$\|11\rangle$
$\|11\rangle$	$\|10\rangle$	$\|10\rangle$

We can see that applying these two gates to states with the first qubit in the $|1\rangle$ state yields identical results, but applying them to states with the first qubit in the $|0\rangle$ state produces states that differ in the second qubuit. This makes sense, since the **CNOT** gate is defined as "apply **X** gate to the target qubit if the control qubit is in the $|1\rangle$ state, and do nothing if it is in the $|0\rangle$ state".

Thus, the easiest solution is: allocate two qubits in the $|00\rangle$ state and apply the unitary to them, then measure the second qubit; if it is `One`, the gate is $I \otimes X$, otherwise it's **CNOT**.

In [ ]:
```
open Microsoft.Quantum.Measurement;

operation DistinguishIXfromCNOT (unitary : (Qubit[] => Unit is Adj+Ctl)) : Int {
    using (qs = Qubit[2]) {
        unitary(qs);
        return MResetZ(qs[1]) == One ? 0 | 1;
    }
}
```