Pauli Measurements

12/11/2017 9 minutes to read Contributors

In this article

In the previous discussions, we have focused on computational basis measurements. In fact there are other common measurements that occur in quantum computing that, from a notational perspective, are convenient to express in terms of computational basis measurements. The most common set of these measurements are *Pauli measurements*. In such cases, it is common to discuss measuring a Pauli operator, in general an operator such as X, Y, Z or $Z \otimes Z, X \otimes X, X \otimes Y$ and so forth. Discussing measurement in terms of Pauli operators is especially common in the subfield of quantum error correction. In Q# we follow a similar convention; we now explain this alternative view of measurements.

Before delving into the details of how to think of a Pauli measurement, it is useful to think about what measuring a single qubit inside a quantum computer does to the quantum state. Imagine that we have an n-qubit quantum state; then measuring one qubit immediately rules out half of the 2^n possibilities that state could be in. In other words, the measurement projects the quantum state onto one of two half-spaces. We can generalize the way we think about measurement to reflect this.

In order to concisely identify these subspaces, we need a language for describing them. One way to do this is to describe the two subspaces by specifying them through a matrix that just has two unique eigenvalues, taken by convention to be ± 1. The simplest example of this is:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

The Pauli-Z matrix clearly has two eigenvectors $|0\rangle$ and $|1\rangle$ with eigenvalues ± 1. Thus if we measure the qubit and obtain $|0\rangle$ we are in the $+1$ eigenspace (the set of all vectors that are formed of sums of eigenvectors with only positive or only negative eigenvalues) of the operator and if we measure $|1\rangle$ we are in the -1 eigenspace of Z. This process is referred to in the language of Pauli measurements as "measuring Pauli Z" and is entirely equivalent to performing a computational basis measurement.

Of course any 2×2 matrix that is a unitary transformation of Z also satisfies this criteria. This is to say that we could also consider matrix $A = U^\dagger Z U$, for any unitary matrix U, to give a matrix that defines the two outcomes of a measurement in its ± 1 eigenvectors. The notation of Pauli measurements references this by identifying X, Y, Z measurements as equivalent measurements that one could do to gain information from a qubit. These measurements are given below for convenience.

Pauli Measurement	U
Z	$\mathbf{1}$
X	H
Y	$HS^\dagger$

That is, using this language, "measure Y" is equivalent to applying $HS^\dagger$ and then measuring in the computational basis, where S is the so-called phase gate given by

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

It is also equivalent to applying $HS^\dagger$ to the quantum state vector and then measuring Z. The correct state would then be found by transforming back to the computational basis, which amounts to applying SH to the quantum state vector.

In Q# we say the outcome, i.e., the classical information extracted from interacting with the state, is j which is in the set $0, 1$ if the result is in the $(-1)^j$ eigenspace of the Pauli operator measured.

Measurements of multi-qubit Pauli operators are defined similarly, as seen from:

$$Z \otimes Z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Thus the tensor products of two Pauli-Z operators forms a matrix composed of two spaces consisting of $+1$ and -1 eigenvalues. As with the single-qubit case, both constitute a half-space meaning that half of the accessible vector space belongs to the $+1$ eigenspace and the remaining half to the -1 eigenspace. In general, it is easy to see from the definition of the tensor product that any tensor product of Pauli-Z operators and the identity also obeys this. For example,

$$Z \otimes \mathbf{1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

As before, any unitary transformation of such matrices also describes two half-spaces labeled by ± 1 eigenvalues. For example, $X \otimes X = H \otimes H(Z \otimes Z)H \otimes H$ from the identity that $Z = HXH$. Similar to the one-qubit case, all two-qubit Pauli-measurements can be written as $U^\dagger(Z \otimes 1)U$ for 4×4 unitary

matrices U. We enumerate the transformations in the following table where we introduce for convenience the swap gate which swaps qubits 0 and 1: $\text{SWAP} = \text{CNOT}_{01}\text{CNOT}_{10}\text{CNOT}_{01}$:

Pauli Measurement	U
$Z \otimes \mathbf{1}$	$\mathbf{1} \otimes \mathbf{1}$
$X \otimes \mathbf{1}$	$H \otimes \mathbf{1}$
$Y \otimes \mathbf{1}$	$HS^\dagger \otimes \mathbf{1}$
$\mathbf{1} \otimes Z$	SWAP
$\mathbf{1} \otimes X$	$(H \otimes \mathbf{1})\text{SWAP}$
$\mathbf{1} \otimes Y$	$(HS^\dagger \otimes \mathbf{1})\text{SWAP}$
$Z \otimes Z$	CNOT_{10}
$X \otimes Z$	$\text{CNOT}_{10}(H \otimes \mathbf{1})$
$Y \otimes Z$	$\text{CNOT}_{10}(HS^\dagger \otimes \mathbf{1})$
$Z \otimes X$	$\text{CNOT}_{10}(\mathbf{1} \otimes H)$
$X \otimes X$	$\text{CNOT}_{10}(H \otimes H)$
$Y \otimes X$	$\text{CNOT}_{10}(HS^\dagger \otimes H)$
$Z \otimes Y$	$\text{CNOT}_{10}(\mathbf{1} \otimes HS^\dagger)$
$X \otimes Y$	$\text{CNOT}_{10}(H \otimes HS^\dagger)$
$Y \otimes Y$	$\text{CNOT}_{10}(HS^\dagger \otimes HS^\dagger)$

Here the gate CNOT_{10} appears for the following reason. Each Pauli measurement that does not include the $\mathbf{1}$ matrix is equivalent up to a unitary to $Z \otimes Z$ by the above reasoning. The eigenvalues of $Z \otimes Z$ only depend on the parity of the qubits that comprise each computational basis vector and the controlled-not operations that appear in this list serve to compute this parity and store it in the first bit. Then once the first bit is measured, we can recover the identity of the resultant half-space which is equivalent to measuring the Pauli operator.

One additional note, while it may be tempting to assume that measuring $Z \otimes Z$ is the same as measuring $Z \otimes 1$ and then $1 \otimes Z$, this assumption would be false. The reason is that measuring $Z \otimes Z$ projects the quantum state into either the $+1$ or -1 eigenstate of these operators. Measuring $Z \otimes 1$ and then $1 \otimes Z$ projects the quantum state vector first onto a half space of $Z \otimes 1$ and then onto a half space of $1 \otimes Z$. As there are four computational basis vectors, performing both measurements reduces the state to a quarter-space and hence reduces it to a single computational basis vector.

Another way of looking at measuring tensor products of Paulis such as $X \otimes X$ or $Z \otimes Z$ is that these measurements let you look at information stored in the correlations between the two qubits. Measuring $X \otimes 1$ lets you look at information that is locally stored in the first qubit. While both types of measurements are equally valuable in quantum computing, the former illuminates the power of quantum

computing. It reveals that in quantum computing often the information you wish to learn is not stored in any single qubit but rather stored non-locally in all the qubits at once, and only by looking at it through a joint measurement with $Z \otimes Z$ does this information become manifest.

Arbitrary Pauli operators such as $X \otimes Y \otimes Z \otimes \mathbf{1}$ can also be measured. All such tensor products of Pauli operators have only two eigenvalues ± 1 and both eigenspaces constitute half-spaces of the entire vector space. Thus they coincide with the requirements stated above.

In Q#, such measurements return j if the measurement yields a result in the eigenspace of sign $(-1)^j$. Having this as a built-in feature in Q# is helpful because measuring such operators requires long chains of controlled-NOT gates and basis transformations to describe the diagonalizing U gate needed to express the operation as a tensor product of Z and 1. By simply being able to specify that you wish to do one of these pre-defined measurements, you don't need to worry about how to transform your basis such that a computational basis measurement provides the necessary information. Q# handles all the necessary basis transformations for you automatically. See Q# library reference for Pauli measurements

The No-Cloning Theorem

Quantum information is powerful. It enables us to do amazing things such as factor numbers exponentially faster than the best known classical algorithms, or efficiently simulate correlated electron systems that classically require exponential cost to simulate accurately. However, there are limitations to the power of quantum computing. One such limitation is given by the *No-Cloning Theorem*.

The No-Cloning Theorem is aptly named. It disallows cloning of generic quantum states by a quantum computer. The proof of the theorem is remarkably straightforward. While a full proof of the no-cloning theorem is a little too technical for our discussion here, the proof of the theorem in the case where the quantum computer in question has no additional ancilla qubits is within our scope (ancilla qubits are qubits used for scratch space during a computation and are easily used and managed in Q#, see Working with qubits.) For such a quantum computer, the cloning operation must be a unitary matrix. We disallow measurement, since it would corrupt the quantum state we are trying to clone. The unitary matrix we want must have the property that

$$U|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle,$$

for any state $|\psi\rangle$. The linearity property of matrix multiplication then implies that for any second quantum state $|\phi\rangle$,

$$U\left[\frac{1}{\sqrt{2}}(|\phi\rangle + |\psi\rangle)\right]|0\rangle = \frac{1}{\sqrt{2}}U|\phi\rangle|0\rangle + \frac{1}{\sqrt{2}}U|\psi\rangle|0\rangle$$

$$= \frac{1}{\sqrt{2}}(|\phi\rangle|\phi\rangle + |\psi\rangle|\psi\rangle)$$

$$\neq \left(\frac{1}{\sqrt{2}}(|\phi\rangle + |\psi\rangle)\right) \otimes \left(\frac{1}{\sqrt{2}}(|\phi\rangle + |\psi\rangle)\right).$$

This provides the fundamental intuition behind the No-Cloning Theorem: any device that copies an unknown quantum state must induce errors on at least some of the states it copies. While the key assumption that the cloner acts linearly on the input state can be violated through the addition of ancilla and measurement of the ancilla qubits, such interactions also leak information about the system through the measurement statistics and prevent exact cloning in such cases as well. For a more complete proof of the No-Cloning Theorem see For more information.

The No-Cloning Theorem is important for qualitative understanding of quantum computing because if you could clone quantum states inexpensively then you would be granted a near-magical ability to learn from quantum states. Indeed, you could violate Heisenberg's vaunted uncertainty principle. Alternatively, you could use an optimal cloner to take a single sample from a complex quantum distribution and learn everything you could possibly learn about that distribution from just one sample. This would be like you flipping a coin and observing heads and then upon telling a friend about the result having them respond "Ah the distribution of that coin must be Bernoulli with $p = 0.512643\ldots$!" Such a statement would be non-sensical because one bit of information (the heads outcome) simply cannot provide the many bits of information needed to encode the distribution without substantial prior information. Similarly, without prior information we cannot perfectly clone a quantum state just as we cannot prepare an ensemble of such coins without knowing p.

Information is not free in quantum computing. Each qubit measured gives a single bit of information, and the No-Cloning Theorem shows that there is no backdoor that can be exploited to get around the fundamental tradeoff between information gained about the system and the disturbance invoked upon it.

ⓘ Note

The feedback system for this content will be changing soon. Old comments will not be carried over. If content within a comment thread is important to you, please save a copy. For more information on the upcoming change, we invite you to read our blog post.