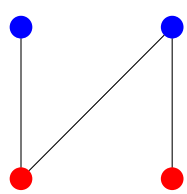
Quantum Circuit Graphs

Benjamin F. Dribus

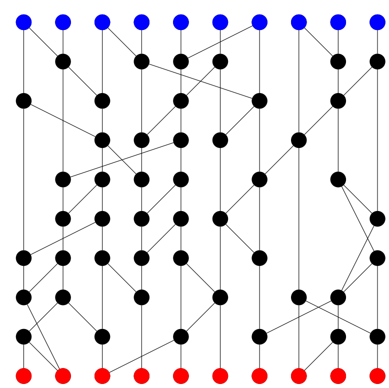
1 Introduction

A **quantum circuit** is a device that performs a unitary transformation on a quantum state involving a finite number of qubits. In this paper, I describe how to represent quantum circuits using mathematical objects called **graphs**, which are diagrams constructed from points and line segments. Graphs provide useful insights and surprising connections between quantum information theory and other areas of physics, including ultra-modern topics like quantum gravity. The graph below on the left represents a quantum circuit called the quantum controlled NOT gate, or QCNOT gate for short, which performs a unitary transformation on two-qubit states. This transformation may also be described by a 4×4 matrix. The other graph represents a 10-qubit quantum circuit, whose unitary transformation requires a 1024×1024 matrix to describe!

QCNOT graph:



10-qubit graph:



The purpose of this paper is to enable talented undergraduates to tackle some interesting research projects in quantum information theory, outlined in section 6 below. I suggest that you check there now to see if any of these projects interest you. Let me briefly describe what's in the rest of the paper. In sections 2 and 3, I go over a few facts about qubits and quantum circuits. You are probably already learning about these topics, but it may help to review some notation and terminology. In section 4, I discuss graphs. This material may be new to you, but it is very simple, with many illustrations. In section 5, I show how graphs may be used to represent quantum circuits, using examples such as the Pauli X, Y, and Z gates and the QCNOT gate. I don't spell out rigorous algorithms here, though the same information could easily be presented in rigorous form. Section 6 contains the project descriptions.

2 Qubits

2.1 Single-Qubit States

Qubits are the basic units of quantum information. Qubit states may be represented by nonzero vectors in a two-dimensional complex vector space $\mathcal{H}$, with orthonormal basis vectors labeled $|0\rangle$ and $|1\rangle$. In this context, the numbers 0 and 1 are called **quantum numbers.** The convention of denoting a vector using the symbols $|\ \rangle$ is called **Dirac notation**. The basis vectors $|0\rangle$ and $|1\rangle$ are usually chosen to correspond to some obvious physical data such as spin up and spin down of an electron, or vertical and horizontal polarization of a photon.

A nonzero vector in $\mathcal{H}$ is a nonzero complex linear combination $\alpha|0\rangle + \beta|1\rangle$ of the basis vectors, called a **superposition**. Two nonzero vectors that are nonzero multiples of each other represent the same physical state. The vector $\alpha|0\rangle + \beta|1\rangle$ may be identified with the column vector $\binom{\alpha}{\beta}$. In particular, the basis vectors $|0\rangle$ and $|1\rangle$ may be identified with the column vectors $\binom{1}{0}$ and $\binom{0}{1}$. To avoid confusion, note that the numbers 0 and 1 in the column vector $\binom{1}{0}$ are the *coefficients* in the expression $|0\rangle = 1|0\rangle + 0|1\rangle$. They have nothing to do with the quantum numbers 0 and 1 appearing in the Dirac notation!

2.2 Multi-Qubit States

Multi-qubit states may be represented by adding additional quantum numbers inside the Dirac symbols $|\ \rangle$. For example, two-qubit states are represented by nonzero vectors in a four-dimensional complex vector space with basis vectors $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$. These basis vectors may be viewed as column vectors in the following way:

$$|00\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}, \qquad |01\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}, \qquad |10\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix}, \qquad \text{and} \qquad |11\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}.$$

A general two-qubit state is represented by a nonzero linear combination $\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$ of these basis vectors; i.e., a superposition. "Two-qubit state" means that two quantum numbers appear in each *basis* vector, but usually all four basis vectors are involved.

A more sophisticated way to describe multi-qubit states is with **tensor products**. Since these can be confusing, let me explain what they mean for us. Tensor products combine two vector spaces into a larger vector space that mixes their information together in a way suitable for quantum mechanics. For us, the two vector spaces each involve a single qubit, so we are combining "two copies of $\mathcal{H}$." The tensor product is written $\mathcal{H} \otimes \mathcal{H}$, where $\otimes$ is the "tensor symbol." $\mathcal{H} \otimes \mathcal{H}$ turns out to be the space of vectors representing two-qubit states that I already described above! The basis vectors of a tensor product space are always given by sandwiching the tensor symbol between basis vectors from the two original vector spaces. Hence, $\mathcal{H} \otimes \mathcal{H}$ has four basis vectors: $|0\rangle \otimes |0\rangle$, $|0\rangle \otimes |1\rangle$, $|1\rangle \otimes |0\rangle$, and $|1\rangle \otimes |1\rangle$. Now, here is the important point: the notation $|00\rangle$ above is just shorthand for $|0\rangle \otimes |0\rangle$; it means exactly the same thing! Likewise, $|01\rangle$ means $|0\rangle \otimes |1\rangle$, $|10\rangle$ means $|1\rangle \otimes |0\rangle$, and $|11\rangle$ means $|1\rangle \otimes |1\rangle$.

Sometimes it's necessary to calculate the tensor product of two linear combinations of basis vectors, say $\alpha|0\rangle + \beta|1\rangle$ and $\gamma|0\rangle + \delta|1\rangle$. For this, you can use the rule, "constants move across the tensor symbol." For example, $\alpha|0\rangle \otimes \delta|1\rangle$ means the same thing as $\alpha\delta|0\rangle \otimes |1\rangle = \alpha\delta|01\rangle$, by "moving δ across." Writing out the whole tensor product gives you

$$(\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle) = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle.$$

It's also sometimes useful to combine tensor product notation with column vector notation. For example, since $|0\rangle$ means $\binom{1}{0}$, and $|00\rangle$ means $|0\rangle \otimes |0\rangle$, it follows that $\binom{1}{0} \otimes \binom{1}{0}$ means $|00\rangle$. To summarize, I have told you four different ways to say the same thing:

$$|00\rangle \qquad \text{means} \qquad |0\rangle \otimes |0\rangle \qquad \text{means} \qquad \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} \qquad \text{means} \qquad \begin{pmatrix}1\\0\\0\\0\end{pmatrix}.$$

Similar identifications may be made for the other three basis vectors $|01\rangle$, $|10\rangle$, and $|11\rangle$. Each of these notations is useful in an appropriate context.

Although the two-qubit vector $\alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle$ looks messy, it's actually simple in the sense that it factors into the tensor product of single-qubit vectors $\alpha|0\rangle + \beta|1\rangle$ and $\gamma|0\rangle + \delta|1\rangle$. If you choose a nonzero linear combination $\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$ at random, it usually won't factor like this. A state represented by a two-qubit vector that doesn't factor is called **entangled**. The reason for this terminology is that you cannot separate the information in such a state into independent information about distinct single-qubit states.

3 Quantum Gates

3.1 Single-Qubit Gates

A **quantum gate** is just a "simple" quantum circuit, viewed as a building block from which more complicated quantum circuits may be constructed. There is no precise rule for how simple a quantum circuit must be to be considered a quantum gate, but most quantum gates manipulate only a few qubits. The simplest type of quantum gate performs a unitary transformation on single-qubit states. Such a quantum gate is called a **single-qubit gate**. Since the space $\mathcal{H}$ of vectors representing single-qubit states is two-dimensional, the unitary transformation of a single-qubit gate is given by a 2×2 unitary matrix. Examples of single-qubit gates include the Pauli X, Y, and Z gates, whose matrices are

$$X = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}, \qquad Y = \begin{pmatrix}0 & -i\\i & 0\end{pmatrix}, \qquad \text{and} \qquad Z = \begin{pmatrix}1 & 0\\0 & -1\end{pmatrix}.$$

To see how such a gate transforms a single-qubit state, use the column vector representations of the basis vectors $|0\rangle$ and $|1\rangle$. For example, the Pauli X gate works like this:

$$X|0\rangle = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}0\\1\end{pmatrix} = |1\rangle \qquad \text{and} \qquad X|1\rangle = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}1\\0\end{pmatrix} = |0\rangle.$$

3.2 Two-Qubit Gates; the QCNOT Gate

The next-simplest type of quantum gate performs a unitary transformation on two-qubit states. Such a quantum gate is called a **two-qubit gate**. Since the vector space $\mathcal{H} \otimes \mathcal{H}$ of vectors representing two-qubit states is four-dimensional, the unitary transformation of a two-qubit gate is given by a 4×4 unitary matrix. An example of a two-qubit gate is the QCNOT gate, mentioned in the introduction, whose matrix is

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Using column vectors, one may show that QCNOT performs the following transformations on the four basis vectors of the space of two-qubit states:

$$\begin{cases} |00\rangle \mapsto |00\rangle \\ |01\rangle \mapsto |01\rangle \\ |10\rangle \mapsto |11\rangle \\ |11\rangle \mapsto |10\rangle. \end{cases}$$

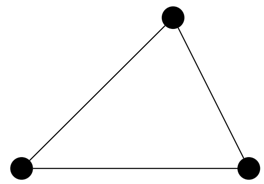
The QCNOT gate is a **universal two-qubit gate**. This means that *any* quantum circuit may be approximated to any desired degree of accuracy by a combination of QCNOT gates and single-qubit gates. Most books on quantum information theory give the proof of this. Because of this important fact, the QCNOT gate is worth examining in some detail.

I will mention some terminology associated with the QCNOT gate. The "NOT" reflects the fact that the second quantum numbers in the last two basis vectors $|10\rangle$ and $|11\rangle$ are flipped by the action of the gate: $|10\rangle$ is transformed to $|11\rangle$, and $|11\rangle$ is transformed to $|10\rangle$. In digital logic, where everything is described in terms of 0's and 1's: "NOT 0" means 1, and "NOT 1" means 0. The "C," which stands for "controlled," refers to the fact that these quantum numbers are *only* flipped for the last two basis vectors, whose first quantum numbers are both 1. For the first two basis vectors, whose first quantum numbers are both 0, the second quantum numbers are not flipped. In this sense, the first quantum number "controls" what happens to the second quantum number. The first quantum number is called the **control**, and the second quantum number is called the **target**. The "Q" stands for "quantum," and is not standard terminology. I include it only because it is sometimes necessary to distinguish between a quantum controlled NOT gate and a classical controlled NOT gate, even in purely quantum settings. See project 4 for an example of this.

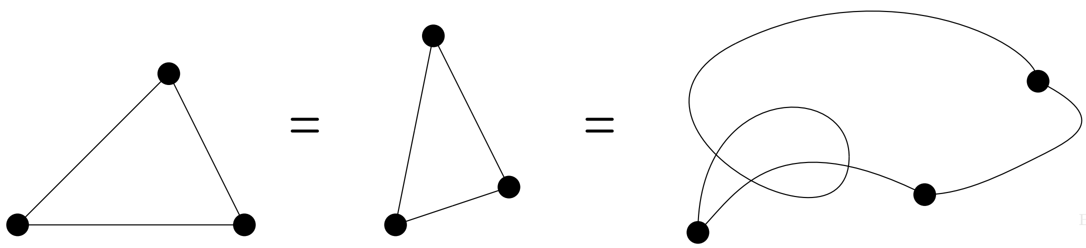
Although I have described the behavior of the QCNOT gate in terms of basis vectors, the two-qubit states we want to transform will usually be represented by superpositions $\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$. What distinguishes the QCNOT gate from the classical controlled NOT gate is that the QCNOT gate applies to such superpositions, not just to basis vectors. To see how the QCNOT gate transforms such a state, apply the gate separately to the basis vectors $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$, multiply the results by α, β, γ, and δ, and add them together. This procedure works because of the **linearity** of quantum theory: if you want to know what happens to a linear combination of states, look at what happens to them separately and add the answers together.

4 Graphs

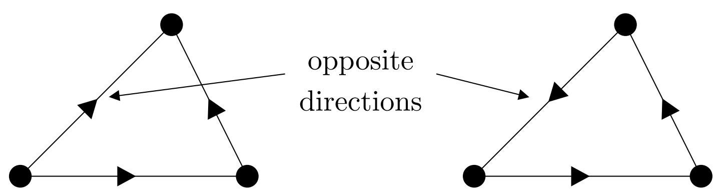
A **graph** is a mathematical object constructed of points, called **vertices**, and line segments, called **edges**. For example, a triangle is a graph with three vertices and three edges:



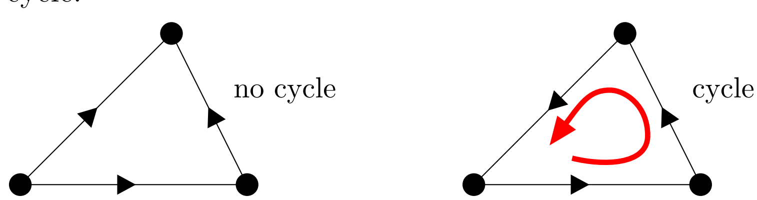
The precise position, orientation, and shape of a graph does not matter; all that matters is the way in which the vertices and edges of the graph are connected to one another. You may think of a graph as being made of elastic; it may be moved, rotated, or stretched in any way without changing its structure. For example, the three graphs below are all considered the same. In particular, it does not matter if the edges cross, as occurs in the third graph below. You may think of such crossings as strands of thread lying on top of each other but not joining; actual joining of edges is always indicated by vertices.



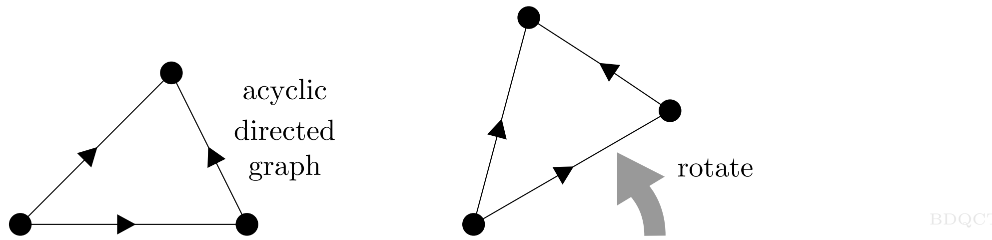
It is sometimes useful to assign directions to the edges, represented by arrows. A graph with directions assigned to its edges is called a **directed graph**. For example, here are two different ways to assign directions to the edges of a triangle:



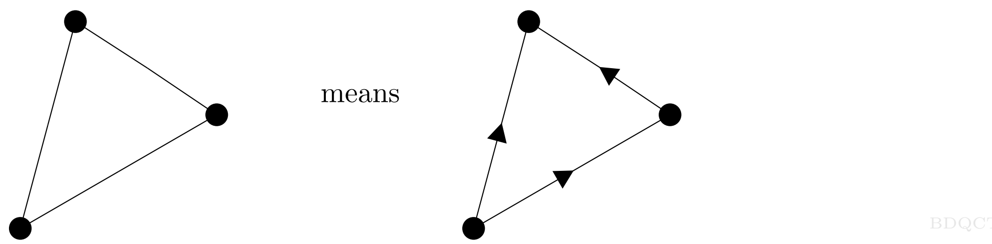
A **cycle** in a directed graph is a directed path along a sequence of edges of the graph, beginning and ending at the same vertex, whose direction agrees with the directions of the edges. Some directed graphs have cycles and some do not. For example, the directed graph on the right below has a cycle, given by going around the triangle counterclockwise, but the directed graph on the left has no cycle:



A directed graph without cycles is called an **acyclic directed graph.** It is possible to redraw any acyclic directed graph so that all the arrows point "up the page." For example, one may redraw the acyclic directed graph on the left by rotating everything counterclockwise. Remember that you are allowed to move, rotate, or stretch a graph.



Since it is always possible to draw an acyclic directed graph so that all the arrows point "up the page," this makes it unnecessary to draw the arrows at all. The rule "up the page" determines the direction of every edge. For example,



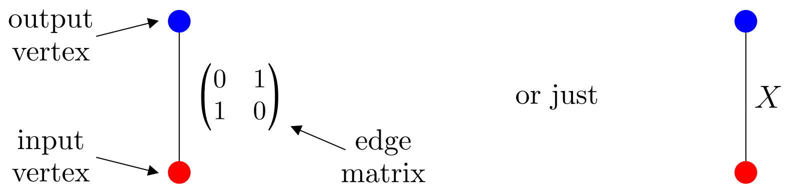
Sometimes it is useful to assign labels containing extra information to the vertices or edges of a graph. These labels could be numbers, vectors, matrices, or something else. For example, the edges of the triangular graph below are labeled with 2×2 matrices, called **edge matrices**:

As I will explain in the next section, **every quantum circuit may be represented by an acyclic directed graph whose edges are labeled by** 2×2 **matrices**. For convenience, I will draw these graphs without arrows. Also, I will just say "graph," rather than repeating "acyclic directed edge-labeled" every time, since all the graphs below fit this description.

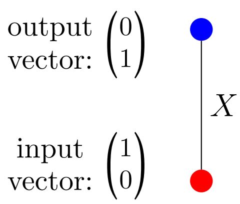
5 Quantum Circuit Graphs

5.1 Graphs for Single-Qubit Gates

Let's begin with the graph for the Pauli X gate, which looks like this:

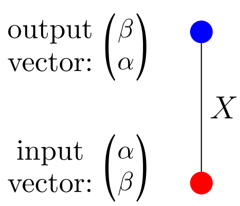


This graph may be interpreted in the following way: imagine feeding the basis vector $|0\rangle = \binom{1}{0}$ into the bottom vertex, colored red. This vertex is the **input vertex**, and $|0\rangle$ is the **input vector**. The input vector flows up through the edge, and is acted upon by the edge matrix $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, which transforms it to the vector $|1\rangle = \binom{0}{1}$. This is the **output vector** at the **output vertex**, colored blue.



action of edge matrix: $$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

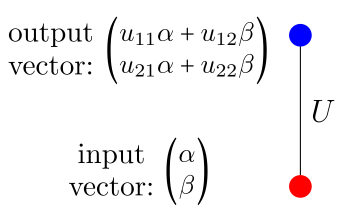
Of course, the input vector will usually be a linear combination $\alpha|0\rangle + \beta|1\rangle$, which corresponds to a column vector $\binom{\alpha}{\beta}$. This is no problem; the edge matrix acts on this column vector in the same way:



action of edge matrix: $$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}$$

Note that this is the same result you would get by separately applying the Pauli X gate to the vectors $|0\rangle$ and $|1\rangle$, multiplying the results by α and β, and adding them together, in accordance with linearity.

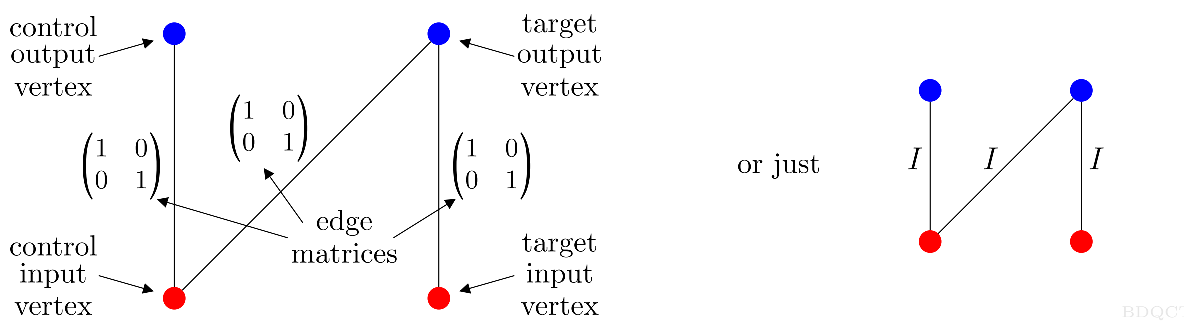
Other single-qubit gates may be assigned graphs in a similar way. Consider a gate whose transformation is given by a unitary matrix U, with entries u_{ij}. The graph for this gate is the same as the Pauli X-graph, except that the edge matrix is changed from X to U:



action of edge matrix: $$\begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} u_{11}\alpha + u_{12}\beta \\ u_{21}\alpha + u_{22}\beta \end{pmatrix}$$

5.2 The QCNOT Graph

Next, let's consider the graph for the QCNOT gate. Since the QCNOT gate transforms two-qubit states, its graph has two input vertices and two output vertices. There are three edges, and each edge matrix is the 2×2 identity matrix $I = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$:



This graph may be interpreted in the following way: suppose we choose the two-qubit vector $|00\rangle = |0\rangle \otimes |0\rangle = \left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right) \otimes \left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right)$ to be the **input vector**. The first single-qubit vector $\left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right)$ in the tensor product $\left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right) \otimes \left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right)$ is the **control input vector**, which flows into the **control input vertex**, and the second single-qubit vector $\left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right)$ is the **target input vector**, which flows into the **target input vertex**. These vectors then flow up through the edges, where they are transformed by the edge matrices, before reaching the top of the graph. In this case, the edge matrices are all the identity, so they have no effect.

The single-qubit vector flowing up the left vertical edge comes through unchanged at the **control output vertex** as the **control output vector**. However, at the **target output vertex**, the single-qubit vectors flowing up the diagonal edge and the right vertical edge must "interact" to produce the **target output vector**. For now, I will just tell you the rule for how this interaction works, and then explain further as we go along. The table on the left below shows what target output vector results from a given pair of single-qubit vectors flowing up the diagonal edge and the right vertical edge. Note that these vectors interact *after* they are transformed by the edge matrices. Right now, this makes no difference, but it will make a difference later on; in other circuits, the vectors interacting where two edges come together will usually *not* be the same as the input vectors. In the present case, the two interacting vectors are both $\left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right)$, so the target output is $\left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right)$ as well. The two-qubit **output vector** is the tensor product:

$$\text{output vector} = (\text{control output vector}) \otimes (\text{target output vector}) = \left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right) \otimes \left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right) = |00\rangle.$$

The table on the right below describes the interaction of the single-qubit vectors flowing up the diagonal edge and the right vertical edge in terms of quantum numbers. To translate between the two tables, note that since the column vector $\left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right)$ stands for $|0\rangle$, which has quantum number 0, and the column vector $\left(\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right)$ stands for $|1\rangle$, which has quantum number 1, each row of the first table agrees with the corresponding row of the second table.

column vector notation:

diagonal	right vertical	target output
$\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}1\\0\end{pmatrix}$
$\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$
$\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$
$\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\end{pmatrix}$

quantum number notation:

diagonal	right vertical	target output
0	0	0
0	1	1
1	0	1
1	1	0

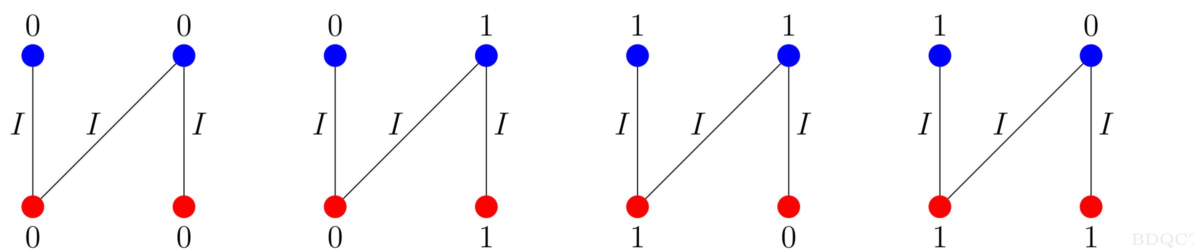
You can confirm that these tables really give the right answer for the target outputs of the QCNOT gate by comparing them to the table of QCNOT transformations of two-qubit basis vectors from section 3.2, reproduced here:

$$\begin{cases} |00\rangle \mapsto |00\rangle \\ |01\rangle \mapsto |01\rangle \\ |10\rangle \mapsto |11\rangle \\ |11\rangle \mapsto |10\rangle. \end{cases}$$

What is happening mathematically at the target output vertex is *modulo two addition of quantum numbers*. This means that the quantum number of the target output vector is given by adding the quantum numbers of the single-qubit vectors flowing up the diagonal edge and the right vertical edge, then reducing the answer modulo 2. While it may be a little troublesome at first to keep switching back and forth between column vectors and quantum numbers, it is easiest to think about what happens on the edges in terms of edge matrices transforming column vectors, and easiest to think about what happens at the target output vertex in terms of adding quantum numbers modulo two. Besides, it is good practice! The reason I wrote out these tables is because **the same interaction rule applies whenever two edges join in a graph representing a quantum circuit.** This means that the behavior of any quantum circuit may essentially be reduced to the action of 2×2 edge matrices and this interaction rule. I won't spell out the details of how you would prove this, but it isn't too hard.

Let's see what the QCNOT graph does if we take $|10\rangle = |1\rangle \otimes |0\rangle = \binom{0}{1} \otimes \binom{1}{0}$ to be the two-qubit input vector. We know that the output vector should be $|11\rangle = |1\rangle \otimes |1\rangle = \binom{0}{1} \otimes \binom{0}{1}$. The control input vector $\binom{0}{1}$ flows upward from the control input vertex through the left vertical edge and the diagonal edge. In each case, the edge matrix is the identity matrix, which leaves this vector unchanged. Nothing further happens at the control output vertex, so the control output vector is $\binom{0}{1}$. Meanwhile, the target input vector $\binom{1}{0}$ flows upward from the target input vertex through the right vertical edge. The edge matrix is the identity matrix, which leaves this vector unchanged. At the target output vertex, it interacts with the vector $\binom{0}{1}$ flowing up the diagonal edge, to produce the target output vector $\binom{0}{1}$. The total output vector is (control output vector) $\otimes$ (target output vector) $= \binom{0}{1} \otimes \binom{0}{1} = |11\rangle$, as we expected.

You can easily check that the same reasoning will give you the correct results for the remaining basis vectors $|01\rangle$ and $|11\rangle$, thereby reproducing the table of QCNOT transformations of two-qubit basis vectors. A convenient way to graphically summarize the action of the QCNOT gate on these vectors is to label the input and output vertices with quantum numbers. For example, labeling the control input vertex and target input vertex with the quantum numbers 1 and 0 indicates that the two-qubit input vector is $|10\rangle$. In this case, the control output vertex and target output vertex are labeled with the quantum numbers 1 and 1, since the two-qubit output vector is $|11\rangle$. Following this procedure for all four basis vectors gives the following four graphs, with labeled vertices and edges:



You can see by examining these graphs, or by consulting the table of quantum numbers above, that the quantum number of the target output vector is 0 if the quantum numbers of the single-qubit vectors interacting at the target output vertex are the same, and 1 if they are the different. Since what is happening mathematically is modulo two addition of quantum numbers, this is rather obvious; it simply reflects the fact that you can these add numbers in either order; i.e., addition is commutative. Physically, this may be interpreted by saying that the single-qubit state represented by the target output vector does not depend individually on the quantum numbers of the single-qubit states that interact to produce it; only the *combination* of these two quantum numbers is significant.

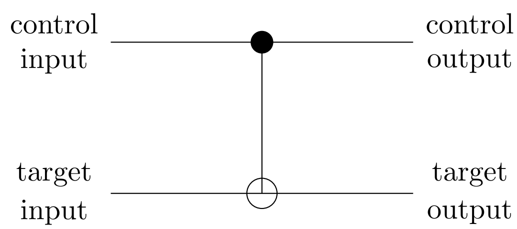
5.3 Some Clarifications

Let me pause here to clarify a few potential issues that could arise regarding the QCNOT graph and its interpretation. First, linearity applies just as in the case of single-qubit gates. A two-qubit input vector will generally be a linear combination $\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$, and the transformation performed on this vector may be computed by passing the individual vectors $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ through the graph, multiplying by the constants α, β, γ, and δ, and adding up the results.

Second, I have so far been working in terms of quantum numbers, 2×1 column vectors, and 2×2 matrices, because it is most natural in the present context to think about the individual behavior of the single-qubit states making up a tensor product. However, if you took the QCNOT graph as a starting point, and didn't already know that this graph represents a quantum circuit, you would need to verify that the procedure I outlined above really defines a unitary transformation on two-qubit states. The easiest way to do this would be to use the 4×1 column vector version of the two-qubit basis vectors, given in section 2.2, to express the table of QCNOT transformations of two-qubit basis vectors as a 4×4 matrix. Of course, this gives you the unitary matrix of the QCNOT gate, given in section 3.2:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Third, you may have seen different-looking diagrams of the QCNOT gate in quantum information books. For example, the QCNOT gate is often represented like this:



I wish I could describe everything in the way you are used to seeing it, but the physical concepts examined in the projects are much harder to understand using diagrams like this. Perhaps the simplest way to explain why this is true is to note that such diagrams aren't simply labeled acyclic directed graphs; they have cumbersome additional structure added. There are many wonderful things you can do with graphs that become muddled if you try to work with such diagrams. I am not suggesting that such diagrams are useless; sometimes they are very useful indeed. But in the present case, we need different methods.

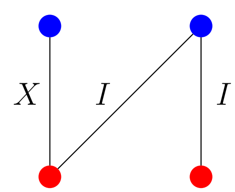
Fourth, let me use the QCNOT graph to warn you about a dangerous conceptual pitfall to avoid. You cannot always break apart a graph, or one of the diagrams you see in quantum information books, and use just a piece of it to represent a quantum circuit. For example, the following graphs, which are parts of the QCNOT graph, make no quantum-theoretic sense by themselves:

The problem with these graphs is that the number of inputs does not equal the number of outputs. This means that they cannot encode unitary transformations, since a unitary transformation is given by a square matrix, which sends a column vector to another column vector of the same size. Hence, these graphs cannot possibly represent quantum circuits. The graph on the right, in particular, gives a famously illegal result if one tries to interpret it in the same way in which I interpreted the QCNOT graph above. If you imagine the single-qubit input vector flowing up both edges and arriving at each of the outputs unchanged, you run smack into the **no-cloning theorem**, which says that you cannot design a device that will duplicate an arbitrary qubit state without destroying it. For the QCNOT graph, you are allowed to think of the control input vector performing two different functions in the sense that it both defines the control output vector and also influences the target output vector, but you cannot observe it unaltered at both outputs. More generally, you may think of things "happening internally" in the quantum circuit that cannot be observed from outside.

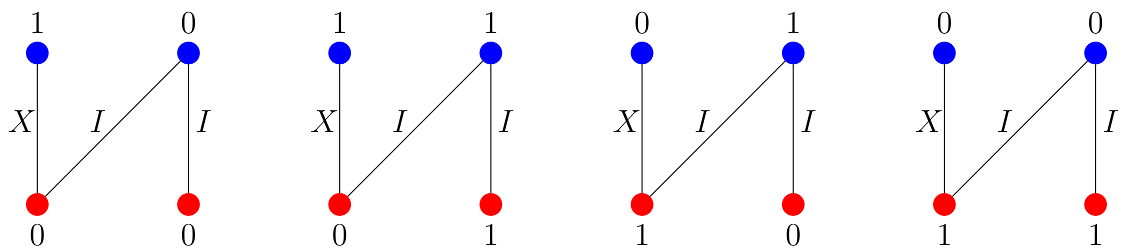
Fifth, when I say that a certain graph "represents" a quantum circuit, I don't mean that the circuit "looks like" the graph physically. If you have ever worked with classical electric circuit diagrams, you know that some aspects of these diagrams resemble the actual physical circuit elements they represent; for instance, "wires" are represented by lines, which resemble physical wires, a pair of parallel line segments representing a capacitor looks like a plate capacitor viewed from the side, and the symbol representing an inductor resembles a coil of wire. In contrast, the rudimentary quantum circuit elements that have thus far been constructed look nothing like graphs physically. What these graphs represent is the *information-theoretic* structure of quantum circuits.

5.4 Graphs for a few other Two-Qubit Gates

Now suppose I change things up a bit and give you a graph whose left vertical edge is labeled with the Pauli X matrix instead of the identity:



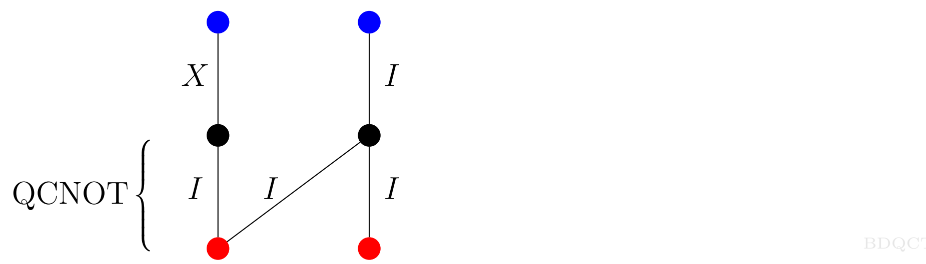
This is no longer the QCNOT gate! Let's see how this new gate transforms the two-qubit basis vectors $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$:



You can understand what's happening here in the same way we analyzed the QCNOT graph. For instance, consider the leftmost graph above, with input quantum numbers 0 and 0, corresponding to the two-qubit input vector $|00\rangle = \binom{1}{0} \otimes \binom{1}{0}$. The only difference between this graph and the QCNOT graph is that the "control input vector" $\binom{1}{0}$ is now transformed to $\binom{0}{1} = |1\rangle$ by the edge matrix X. Hence, the output quantum numbers are 1 and 0. In terms of 4×1 column vectors, the action of the new gate is described by the following unitary matrix:

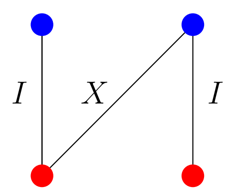
$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

In section 3.2, I mentioned that the QCNOT gate is a universal two-qubit gate, meaning that any quantum circuit may be approximated to any degree of accuracy by using only copies of the QCNOT gate and single-qubit unitary gates. This means that our discussion of graphs for single-qubit unitary gates, along with the QCNOT graph, is actually much more general than it might at first appear. As a first step toward seeing how this works practically, observe that you can reproduce the effect of the new gate I have just introduced by using a QCNOT graph and two single-qubit graphs. The composite graph looks like this:

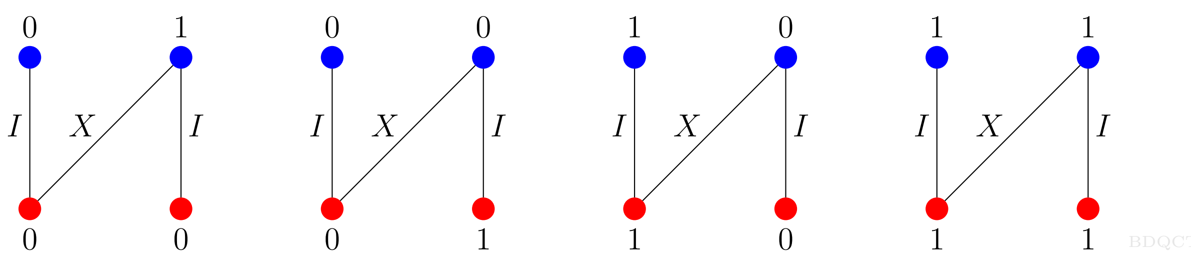


Note that the two vertices in the middle are not colored, since they are neither input vertices nor output vertices of the entire graph, even though they are output vertices of the QCNOT part of the graph at the bottom. Such vertices are called **internal vertices**. The action of this graph is given by first passing a two-qubit state through the QCNOT gate at the bottom, then applying X to the control and I to the target. You can see that this produces exactly the same output as the previous graph. This shows, in particular, that graph representations of unitary transformations are not unique; for instance, you could stack up a string of vertical edges with identity edge matrices, without changing anything. Then again, quantum circuits are not unique either; there are many different combinations of gates that give the same overall unitary transformation.

Let's alter the QCNOT gate in a different way, resulting in the graph on the left below, whose unitary matrix is shown on the right. Beneath are shown the transformations of the two-qubit basis vectors defined by this graph, expressed in terms of quantum numbers:



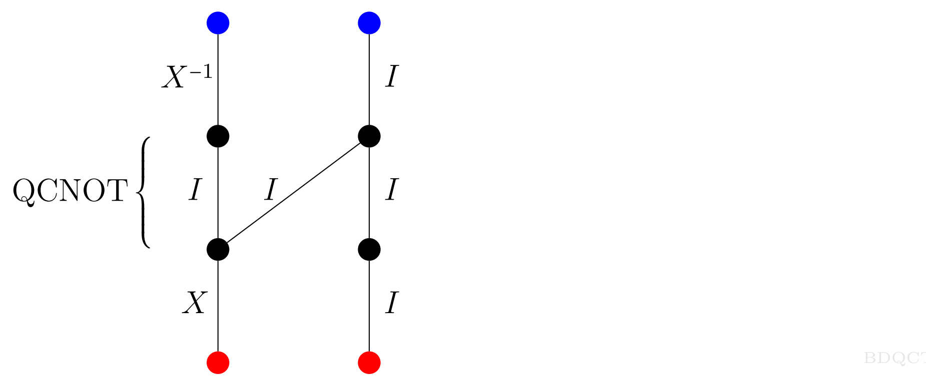
$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



Again, you can understand this gate in the same way we analyzed the QCNOT graph. Consider the second graph at the bottom of the previous figure, with input quantum numbers 0 and 1, corresponding to the two-qubit input vector $|01\rangle = \binom{1}{0} \otimes \binom{0}{1}$. The "control input vector"

$\binom{1}{0}$ flows upward from the bottom left vertex through the left vertical edge and the diagonal edge. The edge matrix on the left vertical edge is the identity matrix, which leaves this vector unchanged, so the "control output vector" is $\binom{1}{0} = |0\rangle$, with quantum number 0. However, the edge matrix on the diagonal edge is $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, which changes this vector to $\binom{0}{1}$. Meanwhile, the "target input vector" $\binom{0}{1}$ flows upward from the bottom right vertex through the right vertical edge. The edge matrix is the identity matrix, which leaves this vector unchanged. At the top right vertex, it interacts with the vector $\binom{0}{1}$ flowing up the diagonal edge, to produce the "target output vector" $\binom{1}{0} = |0\rangle$, with quantum number 0. Hence, the output quantum numbers are 0 and 0, so the two-qubit output vector is $|00\rangle$.

It takes a little more effort to reproduce the effect of this gate using a QCNOT gate and single-qubit unitary gates. Here's one way to do it:



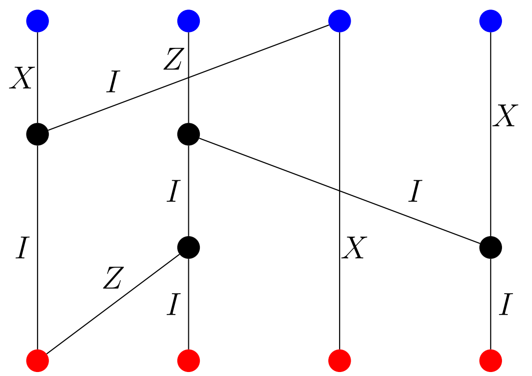
Note that the Pauli X gate is its own inverse, so I could have labeled the top left edge with X rather than X^{-1}. However, if I replaced X with some other single-qubit unitary gate U, I would have to use the inverse matrix here.

5.5 A Glimpse at More Complicated Quantum Circuit Graphs

Using the fact that the QCNOT gate is a universal two-qubit gate, we now know, at least in theory, how to assign a graph to any quantum circuit: first express the circuit as a combination of QCNOT gates and single-qubit unitary gates, whose graphs we have already analyzed, then build the desired graph out of these special graphs. I won't go into the details of how to express a circuit in terms of QCNOT gates and single-qubit gates, since this is well-known and can be found in many different places. However, I will say a few words about the resulting graphs. Any quantum circuit may be represented by a graph satisfying the following four properties:

1. Input vertices are at the bottom of the graph, and output vertices are at the top.
2. The number of inputs is the same as the number of outputs.
3. At most two edges enter each vertex, and at most two edges leave each vertex.
4. Edges are labeled by single-qubit unitary transformations.

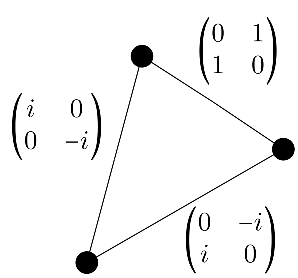
Below is a slightly more complicated example of such a graph, representing a four-qubit quantum circuit:



Below is the 16×16 unitary matrix corresponding to this graph:

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

Finally, I should point out that not every graph satisfying the four properties above represents a quantum circuit. For example, the triangular graph

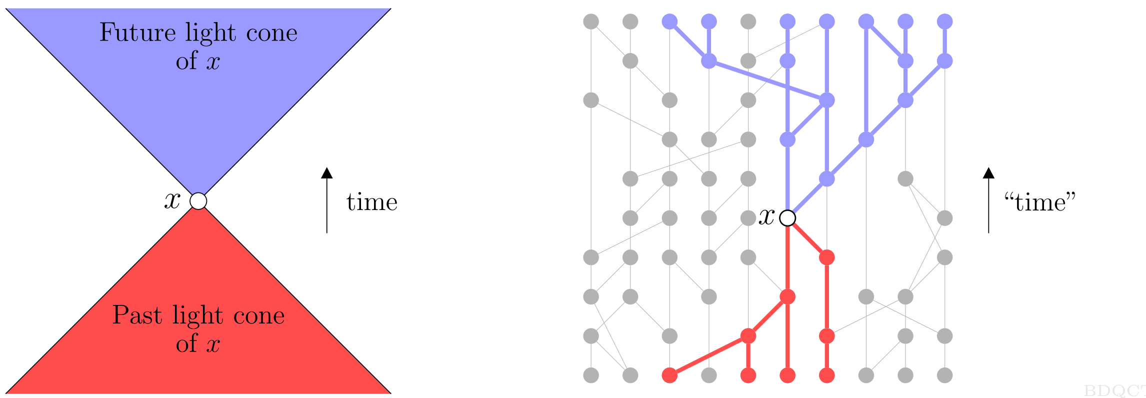


at the end of section 4 doesn't represent a quantum circuit. If you think about it for a second, you can see that this graph represents the relation $XY = iZ$ satisfied by the Pauli matrices. In fact, such graphs have many uses besides quantum computing; they also represent a host of other information-theoretic objects, including classical circuits, networks, materials, objects in advanced algebra, geometry, and topology, and even non-manifold models of spacetime microstructure in the context of quantum gravity.

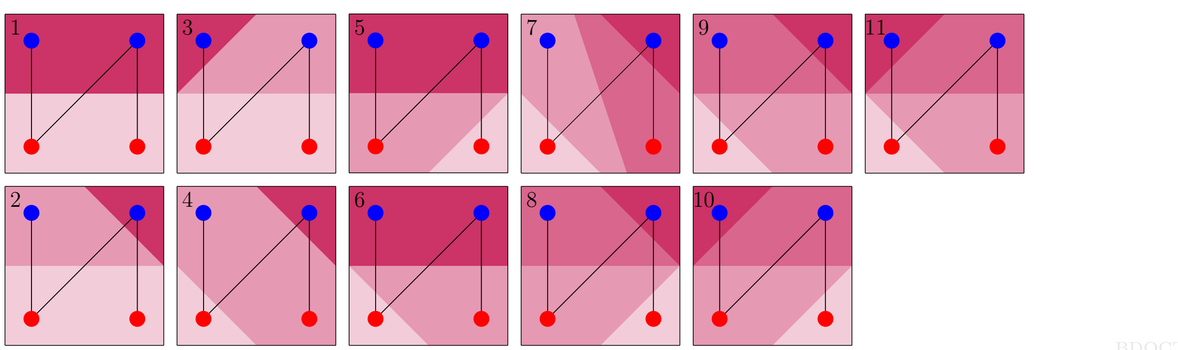
6 PROJECTS ON QUANTUM CIRCUIT GRAPHS

PROJECT 1: Quantum Circuits and Relativity

In relativistic spacetime, the paths traced out by rays of light indicate the scope and direction of information transfer. In particular, no information originating from a spacetime event x can escape the **future light cone** of x, and the **past light cone** of x is the region of spacetime from which information can reach x. In this project, we will examine some interesting similarities between relativistic spacetime and quantum circuit graphs. The following figure shows the light cone of a spacetime event x, alongside a "light cone" in the ten-qubit quantum circuit graph from section 1:



In the figure below, I have illustrated the eleven "frames of reference" for the QCNOT graph from section 5.2. Moving from lighter to darker purple corresponds to moving from "past" to "future:"



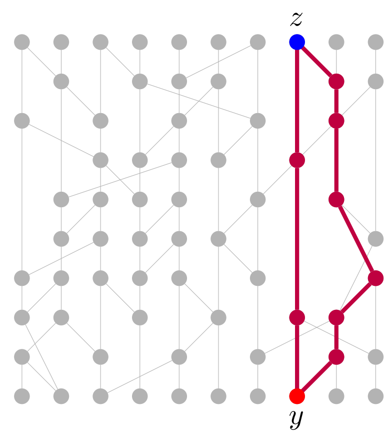
Below are some guiding questions for this project. Carroll's book [4] or online notes are good sources for general relativistic concepts, as is Wikipedia.

1. What is a "light cone" in a quantum circuit graph? What do "past" and "future" mean in this context? What is the significance of these concepts for the flow of quantum information?

2. What is a "frame of reference" in a quantum circuit graph? Do quantum circuit graphs ever have a preferred frame of reference? Explain how the figure above defines frames

of reference for the QCNOT graph.

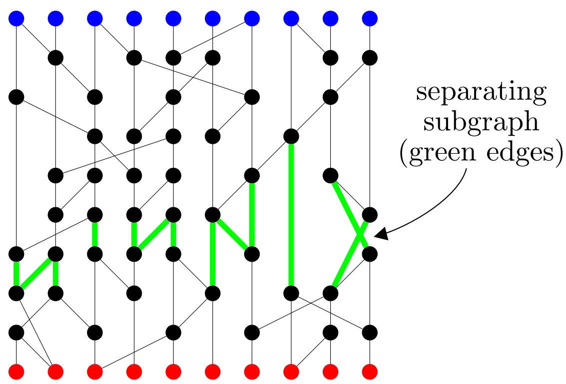
3. Is there an analogue of the **relativity of simultaneity** for quantum circuit graphs?
4. Is there an analogue of **spacelike separation** for quantum circuit graphs?
5. Is there an analogue of the **speed of light** for quantum circuit graphs? Does it make sense to talk about "exceeding the speed of light" in this context?
6. Compare **absolute time** in Newtonian physics, **proper time** in relativity, and **system time** in ordinary computer science. Can you define a notion of proper time for quantum circuits? How does it relate to other ideas of time?
7. Consider the figure below, showing two paths between the vertices y and z in the ten-qubit quantum circuit graph from section 1. Using this as a clue, can you say anything about the **twin paradox** in the context of quantum information theory?

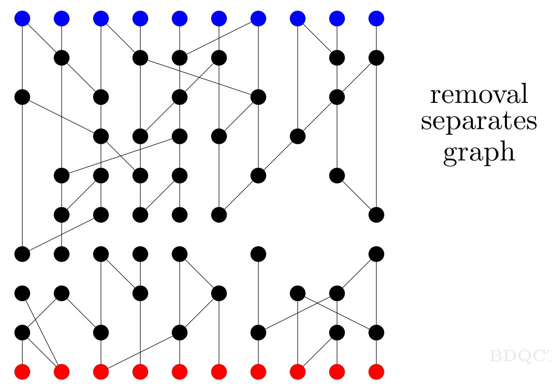


8. Think more generally about **time travel**, both to the future and the past, in the context of quantum computing. Does it have any information-theoretic significance?

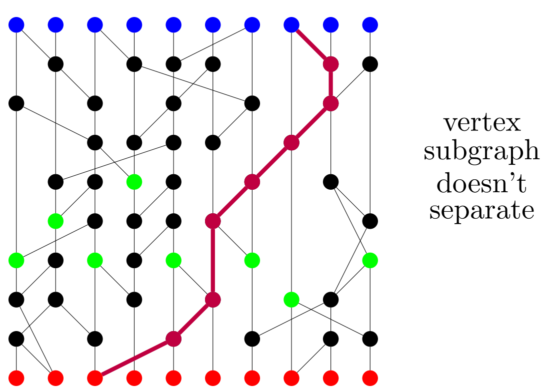
PROJECT 2: Separating Subgraphs of Quantum Circuit Graphs

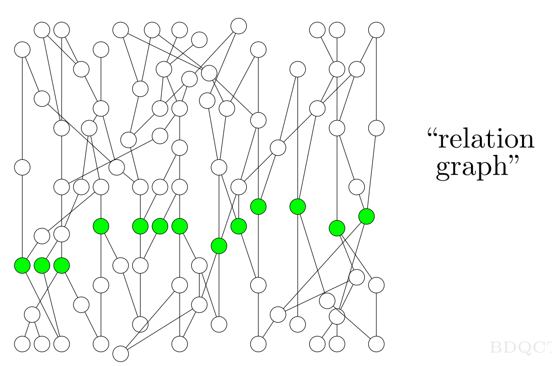
In classical physics, it was believed that if you knew the positions and momenta of every particle in the universe at an "instant in time," you could predict the future with absolute certainty. Quantum theory has altered this view, but the idea of an "instant in time" is still important. In this project, we will explore what an "instant in time" means in quantum information theory by studying special subgraphs of quantum circuit graphs, called **separating subgraphs**, whose removal separates these graphs into two pieces. To accurately resemble an "instant," a separating subgraph should be "as thin as possible." The following figure shows a separating subgraph for the ten-qubit quantum circuit graph from section 1:





The importance of the separating property is that no information can flow between the inputs and outputs of the quantum circuit graph without passing through the separating subgraph. Notice that the separating subgraph above consists of edges rather than individual vertices. Trying to define an "instant in time" in terms of vertices might look like the graph below on the left. The problem is that the green vertices don't separate the graph; for instance, the purple path goes all the way through from bottom to top. You could fix this by making more vertices green, but then you'd no longer have an "instant;" some of the vertices would come "before" others.





Below are some guiding questions for this project. I can't give you many specific references at a suitable level, but a lot of helpful information can be found by searching for individual terms online.

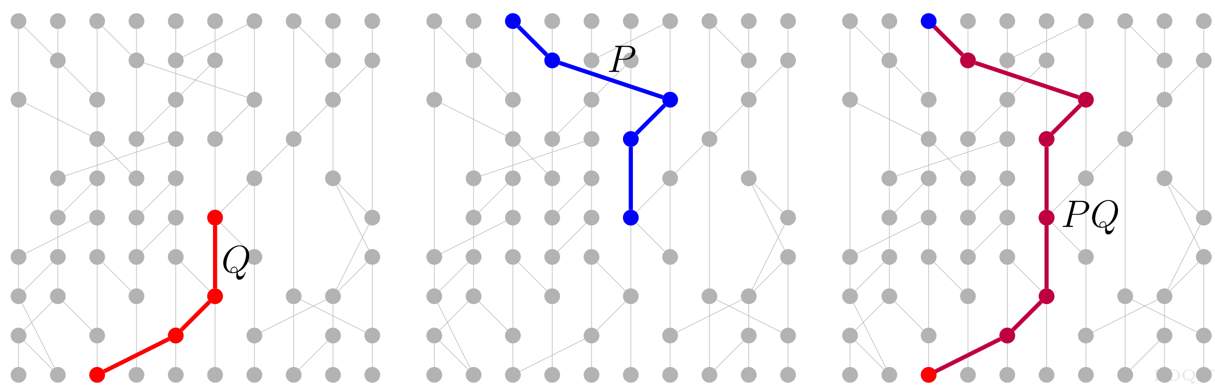
1. Look up **Cauchy surface** on Wikipedia, Google, or in Carroll [4] section 2.7. What is a Cauchy surface? How is it significant? How is it related to a separating subgraph for

a quantum circuit graph?

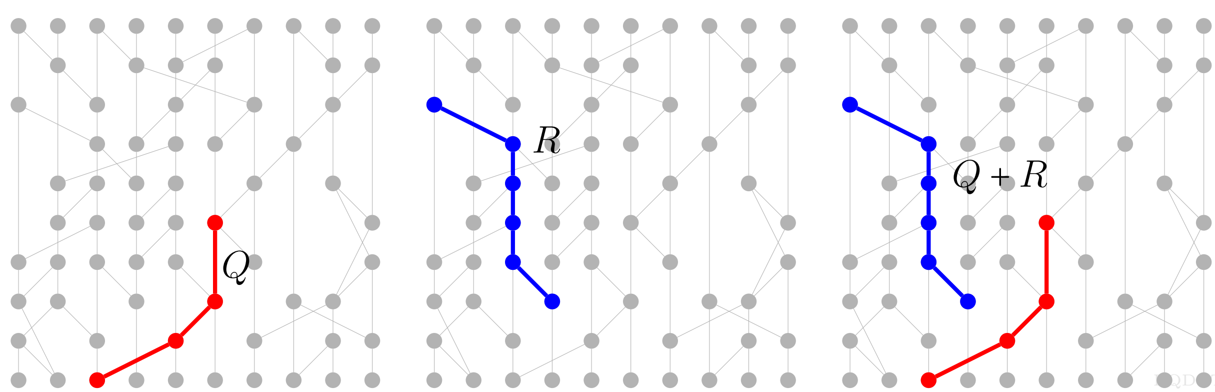
2. What properties should a family of vertices in a quantum circuit graph have to be considered an "instant in time?"
3. Such a family of vertices is called a **maximal antichain**. Why don't maximal antichains usually work as separating subgraphs?
4. Consider the **relation graph** on the right in the previous figure, which has one vertex for every edge in the quantum circuit graph on the left. Can you guess what the edges in the relation graph stand for?
5. In relativity, two spacetime events are **causally related** if information flows from one to the other. Can you guess why the graph on the right in the previous figure is called a "relation graph?"
6. Compare the separating subgraph of green edges in the 10-qubit quantum circuit graph above to the family of green vertices in its corresponding relation graph. How are the two related?
7. Referring back to question 2, is a maximal antichain in the relation graph of a quantum circuit graph always a separating subgraph?
8. The **relational viewpoint** in physics focuses on the relationships between events rather than considering events in isolation. This view is particularly important in modern theories about fundamental spacetime structure, such as **loop quantum gravity** [5] and **causal set theory** [7]. How does this project show the value of the relational viewpoint in quantum information theory?

PROJECT 3: Quantum Circuit Graphs and Path Algebras

A **path algebra** is a mathematical object that precisely encodes the flow of information through a quantum circuit. In this project, we'll work out some of the properties of path algebras and consider their information-theoretic significance. If by some chance you've taken a course on "abstract algebra," you might have already seen "algebras" described as mathematical objects, but I won't assume this. In general, an algebra is something that behaves like the family of all polynomials with real or complex coefficients: you can add or multiply two elements of an algebra, and there are "constant elements," just like there are constant polynomials. In a path algebra, the "variables" are *directed paths in a graph*, and you multiply paths by joining them together. The following figure shows the product of two paths Q and P in the graph for the ten-qubit quantum circuit graph from section 1.



There are a few things to notice here. First, the end vertex of Q is the starting vertex of P. If this is not the case, the product PQ is defined to be zero. In particular, a single vertex v is considered a path of length zero, and the product of v with itself ("v squared") is v. Second, reading the product PQ from left to right goes "back in time" in the sense that P follows after Q in the graph. Third, I've colored the product path PQ purple (a mixture of red and blue) to indicate that it is a *single entity* formed from Q and P. This is what distinguishes multiplication from addition: the sum $P + Q$ is just a family of two paths, not a single path formed by joining two paths together. For this reason, you can add paths that don't share a common vertex. The next figure shows the sum of two paths Q and R. Note that the sum $Q + R$ is *not* a path, it is a family of paths.



Below are some guiding questions for this project. A readable algebra book, such as Dummit and Foote [6], might be useful here, though the internet will, as usual, be the best source.

1. Suppose v and w are two vertices in a quantum circuit graph. What is the product $(v+w)(v+w)$ in the corresponding path algebra? What happens if you take the sum S of a larger family of vertices and multiply S by itself?

2. Suppose P is a path with starting vertex v and end vertex w. What are the products Pv, vP, wP, Pw, wPv, and vPw?

3. You know from working with vector cross products and matrix products that sometimes the order of a product matters; i.e., multiplication can be **noncommutative**. Is multiplication in a path algebra commutative or noncommutative? Suppose A and B are two elements of a path algebra such that $AB = BA$. What can you say about A and B?

4. An **identity element** in an algebra is an element E such that $AE = A$ and $EA = A$ for every element A in the algebra. Does the path algebra of a quantum circuit graph have an identity element? If so, what is it?

5. **Unique factorization** is an important property of polynomials. Unique factorization means that you can always factor a polynomial into a product of **irreducible** polynomials, up to multiplication by constants. Now suppose P is a path in a quantum circuit graph. Does P have a "unique factorization?" If so, what are the "irreducible" factors? What are the "constants?" Does a sum of paths have a unique factorization?

6. Following up on question 5, given a path in a quantum circuit graph, how can you determine algebraically, in terms of the path algebra, how long the path is?

7. The way in which a path algebra encodes information flow in a quantum circuit is complicated by the "interaction" of single-qubit vectors when two edges meet in the quantum circuit graph. For this reason, it's important to be able to determine algebraically when two paths share common vertices. Suppose P and Q are two paths in a quantum circuit graph. How can you tell algebraically if P and Q share the same starting vertex? The same ending vertex? Any vertex at all?

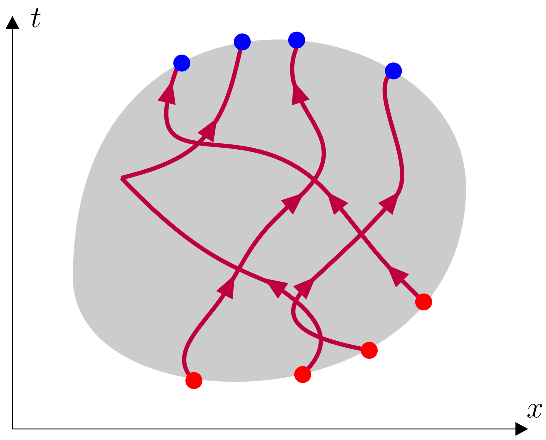
8. Suppose S is a sum of individual vertices in a path algebra. How can you determine algebraically how many vertices are in S?

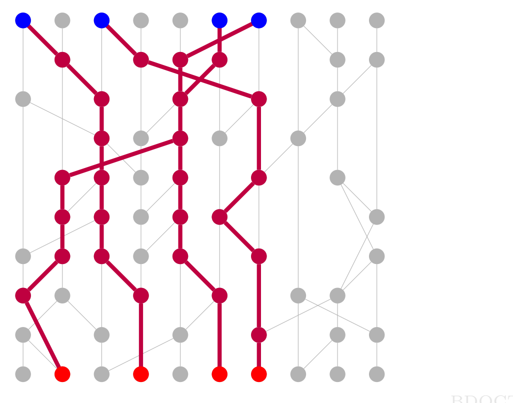
9. How can you determine algebraically when a path in a quantum circuit graph begins at an input vertex or ends at an output vertex?

10. We have so far ignored edge matrices in our examination of path algebras. We can associate a matrix to a path of any length by multiplying together the edge matrices of the individual edges making up the path. We can then define a new algebra called a **twisted path algebra**, whose multiplication rule is "join together paths and multiply matrices." The word "twisted" just means that the multiplication is altered in some way; it doesn't involve physically twisting anything. The "variables" in a twisted path algebra are pairs (P, U), where P is a path and U is a 2×2 unitary matrix. The product of two such variables (Q, V) and (P, U) is (PQ, UV) if the end vertex of Q is the starting vertex of P, and zero otherwise. To what extent is it true that "U describes how a single-qubit vector is transformed as it flows along P?"

PROJECT 4: Sums over Histories for Quantum Circuits

Twenty years after the discovery of modern quantum theory, Richard Feynman introduced a new approach, called the **sum over histories** or **path integral** version of quantum theory. For ordinary quantum mechanics in Newtonian spacetime, Feynman's approach is equivalent to Heisenberg's matrix mechanics and Schrödinger's wave mechanics, but it also applies in situations where these earlier versions of quantum theory have no clear meaning. The sum over histories version of quantum theory takes into account every possible *classical history* of a system. Feynman originally described his approach in terms of spacetime trajectories of particles, as shown on the left below. The conceptual similarity of such trajectories to directed paths in a graph for a quantum circuit, shown on the right, is obvious.



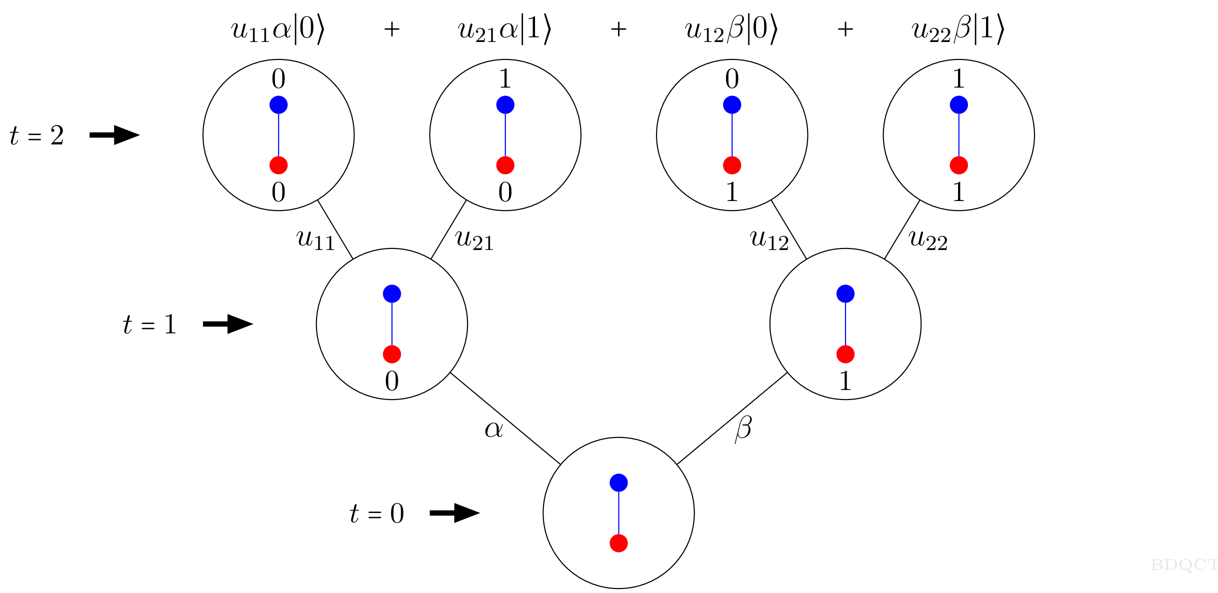


You may be familiar with the famous **double-slit experiment**, in which an electron gun fires an electron at a target screen with a double-slit barrier in between. If so, you probably know that the electron does not "go through one slit or the other;" both slits are significant to the behavior of the electron. Using Feynman's approach, you assign a complex amplitude to each classical trajectory between the gun and the target, then sum up all these amplitudes to get a probability distribution describing where you're most likely to see the electron hit the screen. While it's quantum-mechanically incorrect to talk about exactly where the electron was between firing and detection, each classical trajectory *does* tell a "complete story" about where the electron was at every instant of time. The amplitude associated with this trajectory tells you "how seriously to take this story."

Let's apply Feynman's approach to the simplest type of quantum circuit: a single-qubit quantum gate with unitary transformation U. The input vector will be $\alpha|0\rangle + \beta|1\rangle$, which corresponds to the column vector $\binom{\alpha}{\beta}$. We already know that the output vector should be

$$\begin{pmatrix} u_{11}\alpha + u_{12}\beta \\ u_{21}\alpha + u_{22}\beta \end{pmatrix} = (u_{11}\alpha + u_{12}\beta)|0\rangle + (u_{21}\alpha + u_{22}\beta)|1\rangle,$$

where u_{ij} are the entries of the unitary matrix of U. To define a sum over histories, we must consider each classical description of what could happen during the computation. You might think that the computation consists of applying U to $\binom{\alpha}{\beta}$, but that is thinking quantum-mechanically. Thinking classically, what happens is a one or a zero goes into the gate and a one or a zero comes out. Thus, there are four different classical histories, one for each input-output pair. Remember that it makes no classical sense to put the vector $\binom{\alpha}{\beta}$ into the gate, any more than Schrödinger's cat makes classical sense; you must choose between zero and one! We may imagine three instants of "system time," say $t = 0$, $t = 1$, and $t = 2$, representing the three stages of a classical history: $t = 0$ is before anything happens, $t = 1$ is when the input value is chosen, and $t = 2$ is when the output value is chosen. We may then arrange the classical histories into the following "graph of graphs:"



Each path from the bottom of the graph, at $t = 0$, to the top of the graph, at $t = 2$, represents a classical history. The edges between $t = 0$ and $t = 1$ are labeled with the coefficients α and β of the basis vectors $|0\rangle$ and $|1\rangle$ in the input vector. The edges between $t = 1$ and $t = 2$ are labeled with the entries from U that determine how the basis vectors are transformed. The amplitude assigned to each history is the product of edge labels along the corresponding path in the graph. The sum of the "weighted outputs" at the top of each path is the correct quantum-mechanical output vector!

In this project, we'll further explore how Feynman's sum over histories version of quantum theory applies to quantum circuits. In my opinion, everyone interested in quantum theory ought to read Feynman's original paper [8] at some point, but it's especially useful for understanding the background of this particular project. Here are some guiding questions:

1. Can you work out a sum over histories for the QCNOT gate? Remember that since each history is classical, it involves **classical controlled NOT gates** (CCNOT), rather than QCNOT. You only recover the behavior of QCNOT by taking the sum!

2. You can get the right answer for QCNOT by using three steps of system time, just like we did for the single-qubit gate above. However, this is not the only possible method. To see why, think about the "QCNOT frames of reference" in Project 1.

3. Look up **configuration space** on Wikipedia or elsewhere. How is the "graph of graphs" in the figure above like a **configuration space**? How is it different?

4. Put the words **tree graph theory** into Google and see what you get. Or, find trees in the glossary of graph theory on Wikipedia. Why is the "graph of graphs" above a tree? What does this mean physically? Will every "graph of graphs" for every quantum circuit be a tree?

5. How many steps of system time are involved in Feynman's original sum over histories in Newtonian spacetime?

6. From the **general theory of relativity**, we know that spacetime curves in the presence of matter and energy. Thus, the different trajectories in Feynman's approach make spacetime curve in different ways, though this point is often ignored in practice. Knowing this, is it really correct to think about these trajectories as all occurring in the same spacetime? If not, how would you fix this?

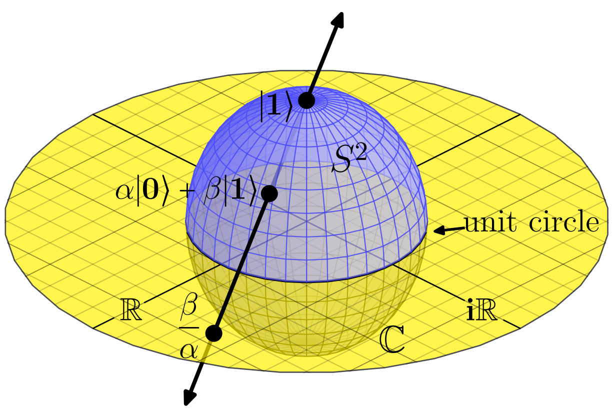
7. Does the "graph of graphs" above suffer from the same issues as spacetime does in question 6?

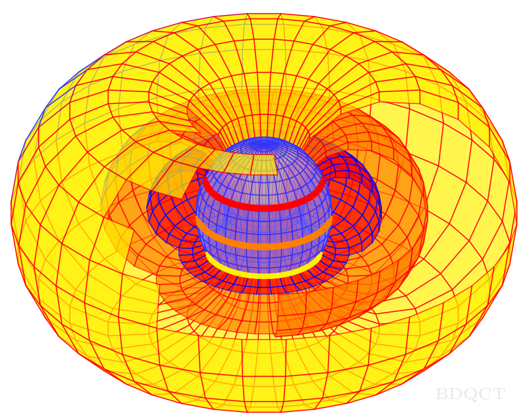
8. Look up the term **background independence**, and think about this concept in the context of questions 6 and 7. Is ordinary quantum theory background independent? What about a quantum circuit? What about general relativity? What about the "graph of graphs" for a quantum circuit?

PROJECT 5: Qubits and the Hopf Fibration

The **Hopf fibration** is a beautiful geometric construction involving spheres, circles, and tori. There are striking connections between the Hopf fibration and quantum information theory. In this project we will explore some of these connections. This project is different from the others in the paper in two ways. First, the Hopf fibration is not directly related to quantum circuit graphs. Second, this is the only project on which you'll find a lot already written online; people have been thinking about the Hopf fibration in the context of quantum information theory since at least 2001. The reason for including this project, besides its inherent interest, is that projects 6 and 7 below use ideas involving the Hopf fibration and quantum circuit graphs together. Hence, this project serves as good background for the next two projects.

A visualization of the Hopf fibration is shown below on the right. On the left is a diagram showing how single-qubit states may be identified with elements of the two-dimensional sphere S^2, or alternatively, with complex numbers. The method of drawing a line connecting the north pole of the sphere S^2 with another point of the sphere, then identifying this point with the intersection point of the same line with the equatorial plane, is called **stereographic projection**.





To connect this to our previous discussion of single-qubit states, recall from section 2.1 that two nonzero vectors in the two-dimensional complex vector space $\mathcal{H}$ that are nonzero multiples of each other represent the same physical state. In particular, if $\alpha|0\rangle + \beta|1\rangle$ is a vector representing a single-qubit state, then the *ratio* of β to α is all that matters in determining the state. In particular, if α is nonzero, then we can divide through by α; hence, the vector $|0\rangle + (\beta/\alpha)|1\rangle$ represents the same state, so we can reconstruct the state if we know the complex number β/α. In this way, *almost* every single-qubit state may be identified with a complex number. The only exception is the case where $\alpha = 0$; in this case, the state is represented by the vector $|1\rangle$, which corresponds to the north pole of the sphere. In the context of quantum information theory, the sphere is called the **Bloch sphere**. In the context of stereographic projection in geometry, the sphere is usually called the **Riemann sphere**.

Now let's discuss the Hopf fibration. The word **fibration** comes from the word **fiber**, which has a special meaning in mathematics. If f is a function, usually called a **map**, between two sets A and B, then the **fiber over an element** b in B is the set of all elements a in A such that $f(a) = b$. A fibration is a map whose fibers are "uniform and well-behaved." For example, the fibers of the Hopf fibration are all circles! The ingredients of the Hopf fibration are:

- **The two-dimensional sphere S^2.** This is the blue sphere in the above diagram. It is the set of all points one unit from the origin in three-dimensional space. In the present context we view S^2 as the Bloch sphere; every point represents a different **state** for a single qubit.
- **The three-dimensional sphere S^3.** I can't draw this all at once, but it's the set of all points one unit from the origin in *four*-dimensional space. The entire three-dimensional space containing the blue sphere and the yellow, orange, and red tori in the above diagram is *most of* S^3, flattened out in a certain way. In fact, only one point is missing, but we'll examine this as part of the project. Every point of S^3 represents a special type of unitary transformation. Just to be precise, these points come in pairs; two different points represent the same transformation. Again, we'll revisit this in the course of the project.
- **A map h sending each point of S^3 to a point of S^2.** I'll give formulas for this map below in terms of both complex and real coordinates. In the diagram, h sends each point on the red torus to a point on the red circle, each point on the orange torus to a point on the orange circle, and each point on the yellow torus to a point on the yellow circle. Points on larger and larger tori map nearer and nearer to the "north pole," and points on smaller and smaller tori map nearer and nearer to the "south poll."

Now let's look at the map h more precisely. Let $\alpha|0\rangle + \beta|1\rangle$ be a vector representing a single-qubit state. Since constant multiples of this vector represent the same state, we can assume that $|\alpha|^2 + |\beta|^2 = 1$, where $|\alpha|$ and $|\beta|$ are the **complex moduli** (i.e., "lengths") of α and β. In terms of the complex numbers α and β, the Hopf fibration is defined by the formula

$$h(\alpha, \beta) = (2\alpha\bar{\beta}, |\alpha|^2 - |\beta|^2),$$

where $\bar{\beta}$ is the **complex conjugate** of β. Let me explain a bit more what this formula means. The first "coordinate" $2\alpha\bar{\beta}$ is a complex number, understood to belong to the yellow plane in the diagram above. This number may be thought of as a *pair* of real coordinates. The second "coordinate" $|\alpha|^2 - |\beta|^2$ is a real number, which is measured in the vertical direction in the diagram above. Hence, the pair $(2\alpha\bar{\beta}, |\alpha|^2 - |\beta|^2)$ really gives you coordinates for a point in three-dimensional space. It turns out that this point is actually on the two-dimensional sphere S^2, colored blue in the diagram above.

It's nice to have a formula in terms of only real numbers, so I'll give you one. First, break α and β into real and imaginary parts:

$$\alpha = x_1 + ix_2 \quad \text{and} \quad \beta = x_3 + ix_4.$$

Then (x_1, x_2, x_3, x_4) are coordinates in four-dimensional space, and S^3 is defined by the equation $x_1^2 + x_2^2 + x_3^2 + x_4^2 = 1$. In terms of these real coordinates, the Hopf fibration is given by the formula

$$h(x_1, x_2, x_3, x_4) = (2x_1x_3 + 2x_2x_4, 2x_2x_3 - 2x_1x_4, x_1^2 + x_2^2 - x_3^2 - x_4^2).$$

Below are some guiding questions for this project. The papers by Lyons [10], Kitagawa [11], and Mosseri and Rossen [12] are all worth looking at here.

1. You may remember from studying complex numbers that $z + \bar{z} = 2\text{Re}(z)$, and $z - \bar{z} = 2\text{Im}(z)$, where $\text{Re}(z)$ and $\text{Im}(z)$ are the **real** and **imaginary parts** of a complex number z. Can you show that $x_1x_3 + x_2x_4 = \text{Re}(\alpha\bar{\beta})$ and $x_2x_3 - x_1x_4 = \text{Im}(\alpha\bar{\beta})$? Why must this be true for the real and complex formulas for the Hopf fibration, given above, to agree?

2. In quantum mechanics, the **expectation value** of an operator U operating on a state Ψ is written in Dirac notation as $\langle\Psi|X|\Psi\rangle$. If Ψ is a single-qubit state $\alpha|0\rangle + \beta|1\rangle$ and U is represented by a 2×2 matrix with entries u_{ij}, then the expectation value is just the number

$$\langle\Psi|U|\Psi\rangle = \begin{pmatrix}\bar{\alpha} & \bar{\beta}\end{pmatrix}\begin{pmatrix}u_{11} & u_{12} \\ u_{21} & u_{22}\end{pmatrix}\begin{pmatrix}\alpha \\ \beta\end{pmatrix} = u_{11}|\alpha|^2 + u_{12}\bar{\alpha}\beta + u_{21}\alpha\bar{\beta} + u_{22}|\beta|^2.$$

   Let X, Y, and Z be the Pauli spin matrices. Can you show that

$$\langle\Psi|X|\Psi\rangle = 2\ \text{Re}(\alpha\bar{\beta}), \quad \langle\Psi|Y|\Psi\rangle = 2\ \text{Im}(\alpha\bar{\beta}), \quad \text{and} \quad \langle\Psi|Z|\Psi\rangle = |\alpha|^2 - |\beta|^2?$$

   Do you recognize these expectation values?

3. Suppose that the single-qubit state represented by the vector Ψ is the spin state of an electron. What do the expectation values $\langle\Psi|X|\Psi\rangle$, $\langle\Psi|Y|\Psi\rangle$, and $\langle\Psi|Z|\Psi\rangle$ represent physically?

4. Look up **special unitary group** on Wikipedia. Why can we assume that the unitary matrices representing transformations of single-qubit states are elements of the second special unitary group SU(2)?

5. Concretely, an element of SU(2) is represented by a 2×2 matrix

$$\begin{pmatrix}\alpha & -\bar{\beta} \\ \beta & \bar{\alpha}\end{pmatrix},$$

   where α and β are complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$. What is the relationship between SU(2) and the three-dimensional sphere S^3? Where else does SU(2) show up in physics?

6. Type **SO(3)** into Wikipedia. SO(3) is a group of 3×3 matrices called the third **special orthogonal group**. What does SO(3) represent physically? How is SO(3) related to SU(2) and S^3?

7. Type **quaternion** into Wikipedia. Can you relate the quaternions to qubits and the Hopf fibration?

8. There are "higher" Hopf fibrations involving four, seven, eight, and fifteen-dimensional spheres. Can you think of how these might be connected to quantum information theory?

9. Type **octonion** into Wikipedia. Can you relate the octonions to multi-qubit states and higher Hopf fibrations?

References

A few remarks may help with the references I have listed below. First and foremost, **the best way to get started is to read the intro and project descriptions in this paper, consult your quantum information theory and quantum mechanics books, and see what you can find online!** For the specific references below, I have also used online sources with clickable links whenever possible. I have tried to use sources containing understandable material, though you shouldn't feel discouraged if you find them difficult. After all, if there were already easy introductions to all the topics in the projects, there would be no need for me to write about them.

The textbooks referenced below are intended only as general sources. You won't find specific information about the projects in them. Barnett [1] is a modern book on quantum information containing standard material such as the proof that the QCNOT gate is a universal two-qubit quantum gate. Ballentine [2] and Shankar [3] are graduate-level quantum mechanics textbooks, but are easier to read and more informative than most "undergraduate-level" books on the subject. Carroll [4] is a very popular textbook on general relativity, again usually used at the graduate-level, but more accessible than many other such books. It contains a concise overview of special relativity, and much of it is available for free in Carroll's online notes. Pullin and Gambini [5] is one of very few well-written books on quantum gravity and fundamental spacetime structure designed specifically for undergraduates. I include it as a reference because these topics form much of the motivation for the point of view I develop in the projects. Dummit and Foote [6] is a standard algebra textbook.

The papers referenced below are more specific in content, though unfortunately no specific sources yet exist for some of the projects. In terms of their accessibility, the papers are a mixed bag. Lyons [10] and Kitagawa [11] are specifically designed to offer accessible explanations, and my essay [9] is at least partially accessible, though the last few sections cover a great deal of advanced and speculative material in a short space. The remaining papers are research papers, primarily intended to announce research results rather than teach concepts. However, some of them are still beautifully written, particularly Sorkin [7] and Feynman [8].

Textbooks

[1] Stephen M. Barnett. *Quantum Information.* Oxford University Press, 2009.

[2] Leslie E. Ballentine. *Quantum Mechanics.* World Scientific, 1998.

[3] Rammamurti Shankar. *Principles of Quantum Mechanics.* Plenum Press, 1980.

[4] Sean M. Carroll. *Spacetime and Geometry: An Introduction to General Relativity.* Addison-Wesley, 2004. Partial online version: http://arxiv.org/abs/gr-qc/9712019.

[5] Jorge Pullin and Rodolfo Gambini. *A First Course in Loop Quantum Gravity.* Oxford University Press, 2011.

[6] David S. Dummit and Richard M. Foote. *Abstract Algebra.* Wiley and Sons, 2004.

Papers

[7] Rafael Sorkin. *Light, Links and Causal Sets*. Journal of Physics Conf. Ser. 174: 012018, 2009. Preliminary online version: http://arxiv.org/abs/0910.0673.

[8] R.P. Feynman. *Space-Time Approach to Non-Relativistic Quantum Mechanics*. Rev. of Mod. Phys., 20, 367, 1948. Review by James Hartle: http://arxiv.org/abs/gr-qc/9210004.

[9] Benjamin F. Dribus. *On the Foundational Assumptions of Modern Physics*. FQXi Essay Contest: Questioning the Assumptions, 2012. URL: http://www.fqxi.org/community/essay/winners/2012.1#dribus.

[10] David W. Lyons. *An Elementary Introduction to the Hopf Fibration*. URL: http://www.nilesjohnson.net/hopf-articles/Lyons_Elem-intro-Hopf-fibration.pdf.

[11] Kitagawa. *Physical Interpretation of Hopf Fibration*. URL: http://www.people.fas.harvard.edu/~tkitagaw/expositional_paper/physical_hopf.pdf.

[12] Remy Mosseri and Dandoloff Rossen. *Geometry of Entangled States, Bloch Spheres, and Hopf Fibrations*. URL: http://arxiv.org/abs/quant-ph/0108137.