QCQI: Chapter 4 Summary

Winter in Data Science 2022

Krishna N Agaram

January 31, 2023

Abstract

We give a short summary of the material covered in *Quantum Computation and Quantum Information* (Nielsen and Chuang). This chapter focuses on Quantum Circuits - the fundamental substance of Quantum Computation. A Quantum Circuit is an efficient and powerful language to describe a quantum algorithm. This construction will enable us to quantify the cost of an algorithm in terms of things like the total number of gates required, or the circuit depth. Universality of a small set of gates is also described in detail.

1 Basics and Notation

- Qubits and Bloch sphere representation of a qubit: A single qubit is a vector $|\psi\rangle = a|0\rangle + b|1\rangle$ parameterized by $a, b \in \mathbb{C}$ satisfying $\| |\psi\rangle \|^2 = |a|^2 + |b|^2 = 1$. A qubit can be visualized as a point (θ, φ) on the unit sphere, where $a = \cos(\theta/2), b = e^{i\varphi}\sin(\theta/2)$, and a can be taken to be real because the overall phase of the state is unobservable. Hence we have the **bloch sphere representation** of a qubit:
$$|\psi\rangle = |\psi(\theta,\varphi)\rangle = \cos(\theta/2)|0\rangle + e^{i\varphi}\sin(\theta/2)|1\rangle \leftrightarrow (\theta,\varphi) \in [0,\pi] \times [0, 2\pi)$$
- Pauli matrices X, Y, Z, I:
$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix},\ Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},\ I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$
$$\vec{\sigma} := [X, Y, Z]^T \text{ (The vector of Pauli matrices)}$$
The eigenvectors of the Pauli matrices lie at the intersections of the Bloch sphere with the x, y, z axes.
- Other useful gates:
$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},\ P \text{ (Phase) } = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix},\ T \text{ (or) } \pi/8 = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$
- Controlled gates:
$$\text{controlled} - X = \begin{bmatrix} I & 0 \\ 0 & X \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix},\ \text{Toffoli} = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & X \end{bmatrix}$$

2 Single qubit operations

We start with the simplest quantum system of all- a single qubit. Operations on a qubit are constrained to preserve this norm, and thus must be described by 2×2 unitary matrices. Common examples include Pauli$-X, Y, Z$ gates and the Hadamard gate.

2.1 Rotation gates

Consider the operator

$$R_x(\theta) := \cos\left(\frac{\theta}{2}\right) I - i \sin\left(\frac{\theta}{2}\right) X = \exp(-i\theta X/2)$$

It has the action of rotating the bloch vector of the statevector it operates on by θ about the x-axis. Similarly, define $R_y(\theta) := \exp(-i\theta Y/2)$ and $R_z(\theta) := \exp(-i\theta Z/2)$. It can be shown that the operator

$$R_{\hat{n}}(\theta) := e^{-i\hat{n}\cdot\vec{\sigma}/2} = R_x(n_x)R_y(n_y)R_z(n_z)$$

has the action of rotating the bloch vector by θ about the unit vector $\hat{n} \in \mathbb{R}^3$. Of course, this general rotation operator reduces to R_x, R_y, R_z when $\hat{n} = \hat{x}, \hat{y}, \hat{z}$ respectively.

Theorem 1. *Suppose U is a unitary operation on a single qubit. Then there exist real numbers α, β, γ and δ such that*

$$U = e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta)$$

Proof. Follows from the fact that a general unitary operator can be written in the form

$$U = \begin{bmatrix} e^{i(\alpha-\beta/2-\delta/2)} \cos\frac{\gamma}{2} & e^{i(\alpha-\beta/2+\delta/2)} \sin\frac{\gamma}{2} \\ e^{i(\alpha+\beta/2-\delta/2)} \sin\frac{\gamma}{2} & e^{i(\alpha+\beta/2+\delta/2)} \cos\frac{\gamma}{2} \end{bmatrix}$$

and matrix multiplication. □

And now, here is the crux to the construction of controlled multi-qubit operations:

Corollary 1.1. *Suppose U is a unitary gate on a single qubit. Then there exist unitary operators A, B, C on a single qubit such that*

$$ABC = I \text{ and } U = e^{i\alpha} AXBXC,$$

where α is some overall phase factor.

Proof. We make use of the following identity:

$$XYX = iZX = i^2 Y = -Y \implies XR_y(-\theta)X = X\left(\cos\left(\frac{\theta}{2}\right) I + i\sin\left(\frac{\theta}{2}\right) Y\right) X = R_y(\theta)$$

Similarly we have $XR_z(-\theta)X = R_z(\theta)$. Now we construct A, B, C making use of the decomposition $U = e^{i\alpha}R_z(\beta)R_y(\gamma)R_z(\delta)$. Consider $A = R_z(\beta)R_y(\gamma/2)$. So $BC = A^{-1} = R_y(-\gamma/2)R_z(-\beta)$. We need a $R_z(\delta)$ at the end of $AXBXC$, so let's take $C = R_z((\delta-\beta)/2)$ which gives $B = R_y(-\gamma/2)R_z(-(\beta+\delta)/2)$. Now

$$XBX = XR_y(-\gamma/2)XXR_z(-(\beta+\delta)/2)X = R_y(\gamma/2)R_z((\beta+\delta)/2)$$

so that

$$e^{i\alpha}AXBXC = e^{i\alpha}R_z(\beta)R_y(\gamma)R_z(\delta) = U$$

and $ABC = I$ as needed. □

3 Controlled Operations

'If A is true, then do B'. This type of controlled operation is one of the most useful in computing, both classical and quantum - what would a programming language be without conditional statements? Consider the prototypical controlled-NOT or CNOT gate. This is a gate with two input qubits, which we call the *control* (usually the first qubit in the statevector is taken to be the control) and *target* qubits. The action of the gate is described by its action on the canonical basis $\{|0\rangle|0\rangle, |0\rangle|1\rangle, |1\rangle|0\rangle, |1\rangle|1\rangle\}$ for the 2-qubit state space. The action is given by

$$|c\rangle|t\rangle \rightarrow |c\rangle|t \oplus c\rangle$$

where $c, t \in \{0, 1\}$. In other words, if the control qubit is set to $|1\rangle$, the target is flipped, and otherwise the target is left alone. $|t \oplus c\rangle$ can also be written $X^c|t\rangle$. Finally, in the computational basis $|\text{control, target}\rangle$ the matrix representation of CNOT is

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

The controlled-U gate is similar: If the control bit is set to $|1\rangle$, U is applied to the target, and otherwise the target is left as is. That is, for $c, t \in \{0,1\}$, the c-U gate takes

$$|c\rangle|s\rangle \to |c\rangle U^c|s\rangle.$$

What is the matrix representation of c-U? - Well, when $c = 0$, it is the identity on $|t\rangle$, and when $c = 1$, it is U on $|t\rangle$. This gives the matrix

$$\begin{bmatrix} I & 0 \\ 0 & U \end{bmatrix}$$

where we note that the first two columns are the images of $|0\rangle|t\rangle$ and the next two of $|1\rangle|t\rangle$. Note the difference between the controlled U gate and the $I \otimes U$ gate: The latter applies U to the target in either case, giving the matrix representation

$$I \otimes U = \begin{bmatrix} U & 0 \\ 0 & U \end{bmatrix} \neq \text{c-}U$$

Finally, an interesting point - the controlled X gate as we saw it was defined as flipping the target qubit iff the control qubit was set in the computational basis. This fully described the operator, yes, but there is no reason to assume that the first bit is always the one that "controls" the operation on the other. For example, consider the Hadamard basis $\{|+\rangle|+\rangle, |+\rangle|-\rangle, |-\rangle|+\rangle, |-\rangle|-\rangle\}$. The action of the CNOT gate is infact

$$\begin{aligned} |+\rangle|+\rangle &\to |+\rangle|+\rangle \\ |+\rangle|-\rangle &\to |-\rangle|-\rangle \\ |-\rangle|+\rangle &\to |-\rangle|+\rangle \\ |-\rangle|-\rangle &\to |+\rangle|-\rangle \end{aligned}$$

which means the second qubit being in $|+\rangle$ or $|-\rangle$ decides whether X will be applied to the first - that is, the second controls the first!

3.1 Implementing a general controlled 2-qubit gate

We use the corollary to Theorem 1 here. We would like to implement the gate $|c\rangle|s\rangle \to |c\rangle U^c|t\rangle$ using only single-qubit operations and CNOT gates. Essentially, if $c = 0$, we must apply $I = ABC$ and must apply $e^{i\alpha}AXBXC$ otherwise. Notice that the unitary operator (where CNOT(1, 2) means that the first qubit is control and the 2nd is the target)

$$(I \otimes A)\text{CNOT}(1,2)(I \otimes B)\text{CNOT}(1,2)(I \otimes C)$$

takes $|0\rangle|t\rangle$ to $|0\rangle ABC|t\rangle = |0\rangle|t\rangle$ and $|1\rangle|t\rangle$ to $|1\rangle AXBXC|t\rangle = |0\rangle e^{-i\alpha}U|t\rangle$ - we are almost there. We only need to get that controlled phase gate. We use a slight trick here: We want a controlled-$e^{i\alpha}I$ gate, with matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{i\alpha} & 0 \\ 0 & 0 & 0 & e^{i\alpha} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{bmatrix} \otimes I$$

so we can replace this with the - exactly-identical-in-action - single qubit gate $\begin{bmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{bmatrix}$ applied to the first qubit. Let's add this to the front (could be anywhere) of our almost complete unitary operator - and we're done!

3.2 General controlled operations

In general, we can have many controls and many targets - the (possibly multiple-qubit) operator U is applied to the target qubit(s) if all the control bits are set. If there are n control bits, then such a controlled operator is called a $C^n(U)$ gate. Otherwise the target qubit(s) are left alone. One very useful example is the Toffoli gate - 2-control 1-target X [$C^2(X)$]gate. How might we implement this, and a general $C^2(U)$ gate? Here is a very elegant construction due to [check names]. Consider any unitary operator V that squares to U. Then the following circuit constructs the $C^2(U)$ gate using only single-qubit and C^1 gates:

Finally, how might we construct a C^n gate? We use $n-1$ ancilla qubits (that all start in $|0\rangle$) and keep using Toffolis to compute $|c_1 \cdot c_2 \dots c_n\rangle$, and finally when we have this stored in a qubit, a simple controlled U gate with this qubit as control gives us the C^n gate. To be able to reuse the ancilla bits used elsewhere, we reset all of them to $|0\rangle$ before ending the computation. The circuit is shown in Figure 3:

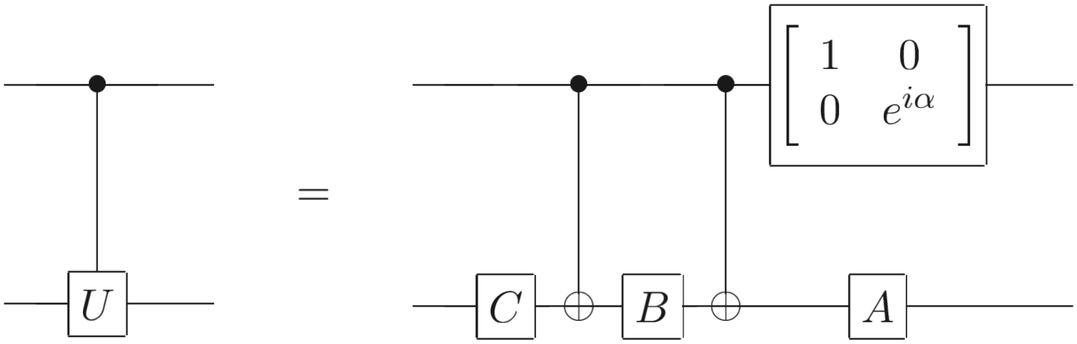

Figure 1: Constructing a general C^1 gate

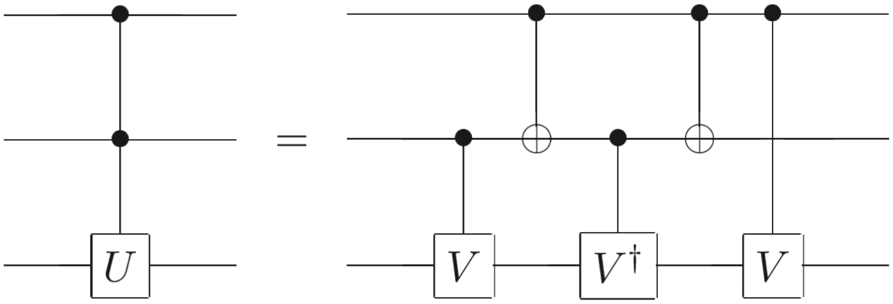

Figure 2: Constructing a general C^2 gate

4 Measurement

A final element used in quantum circuits, almost implicitly sometimes, is measurement. A projective measurement in the computational basis is usually denoted by a "meter" symbol. There are two important useful principles regarding measurement in quantum circuits. Both principles are rather obvious; however, they are of such great utility that they are worth emphasizing early.

The first principle is that classically conditioned operations can be replaced by quantum conditioned operations:

Proposition (Priniciple of Deferred Measurement)**.** *Measurements can always be moved from an intermediate stage of a quantum circuit to the end of the circuit; if the measurement results are used at any stage of the circuit then the classically controlled operations can be replaced by conditional quantum operations.*

The second principle is the following:

Proposition (Principle of implicit measurement)**.** *Without loss of generality, any unterminated quantum wires (qubits which are not measured) at the end of a quantum circuit may be assumed to be measured.*

Why is this true? Consider a composite system AB. Suppose we measure only A. And in another experiment we measure both A and B. Then it can be shown that the reduced density matrix of A is the same in either case, which means that if we only wanted to measure A, we could measure both and still get the same statistics for A that we wanted.

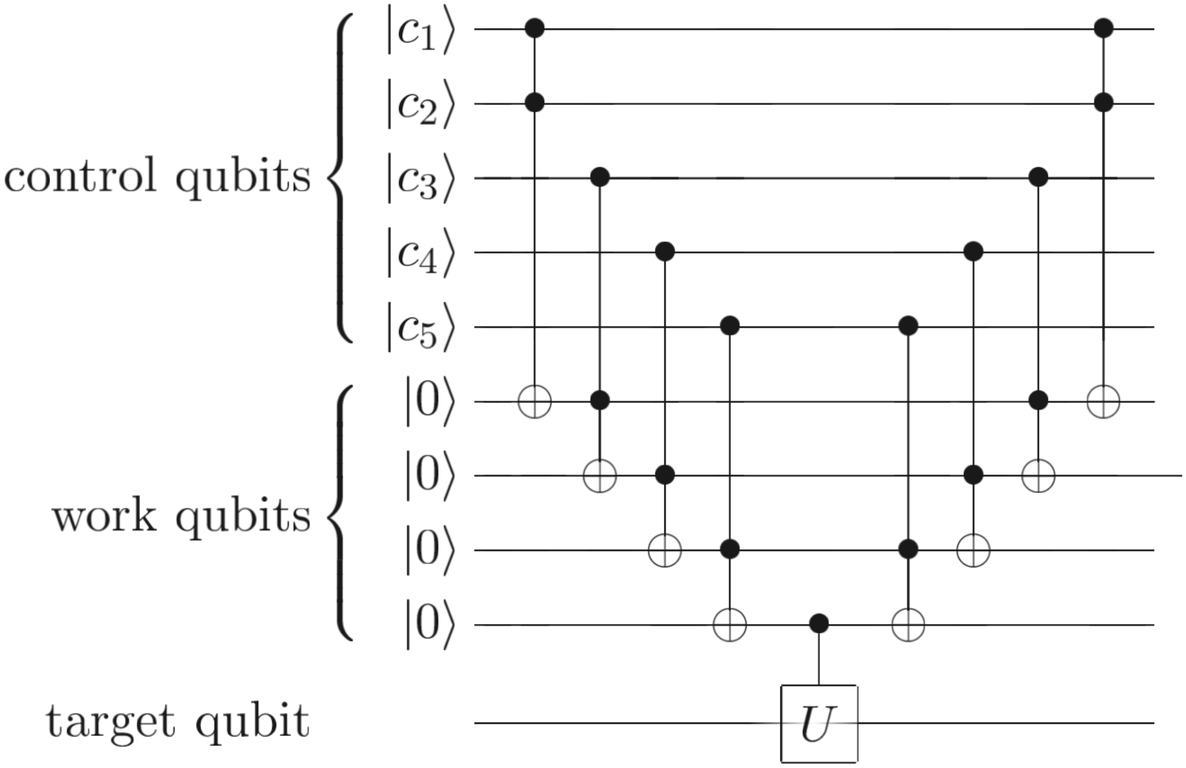

Figure 3: Constructing a general C^n gate

5 Universality

A small set of gates (for example, the set {NOT, AND}, the set {NOR} etc) can be used to compute any classical boolean function. A similar universality result is true for quantum computation, where a set of gates is said to be *universal for quantum computation* if any unitary operation (on an arbitrary number of qubits) may be approximated to **arbitrary accuracy** by a quantum circuit involving only those gates.

Theorem 2. *Hadamard, Phase,* CNOT, $\pi/8$ *is a universal set for quantum computation.*

Proof. Here is a very brief sketch of the proof:

- First show that an arbitrary unitary operator may be expressed exactly as a product of unitary operators that each acts non-trivially only on a subspace spanned by two computational basis states [two-level unitaries are universal].
- Next, prove that any two-level unitary operator may be expressed exactly using single qubit and CNOT gates, which shows that single-qubit and CNOT gates are universal for quantum computation.
- Finally show that single qubit operation may be approximated to arbitrary accuracy using the Hadamard, phase, and $\pi/8$ gates. This in turn implies that any unitary operation can be approximated to arbitrary accuracy using Hadamard, phase, CNOT, and $\pi/8$ gates, as required.

□

Our constructions say little about efficiency - how many (polynomially or exponentially many) gates must be composed in order to create a given unitary transform. It turns out that there exist unitary transforms which *require* exponentially many gates to approximate. Of course, the goal of quantum computation is to find interesting families of unitary transformations that can be performed *efficiently*.