CSE 291 / Math 277A - Quantum Complexity Theory September 29, 2022

Lecture 3

Lecturer: Daniel Grier *Scribe: Cole Kurashige*

1 Some questions

1.1 Are purifications unique?

Spoiler: no.

Recall the definition of purification. Let ρ be a mixed state comprising k individual pure states $|\psi_i\rangle$ on n qubits with probabilities p_i. Then, the density matrix is given by

$$\rho = \sum_{i=1}^{k} p_i \left|\psi_i\right\rangle \left\langle\psi_i\right| .$$

We may purify it by constructing $|\psi\rangle$, a pure state on $2n$ qubits as follows

$$|\psi\rangle = \sum_{i=1}^{k} \sqrt{p_i} \left|\psi_i\right\rangle_A \otimes \left|i\right\rangle_B ,$$

where A refers to the subsystem on the first n qubits and B refers to the subsystem on the second n qubits (recall that without loss of generality we can assume that $k \leq 2^n$).

As we discussed in the previous lecture, this state has the property that $\mathrm{tr}_B(|\psi\rangle\langle\psi|) = \rho$.

Let us expand the partial trace.

$$\begin{aligned}
\mathrm{tr}_B \left|\psi\right\rangle\left\langle\psi\right| &= \mathrm{tr}_B \left[\left(\sum_i \sqrt{p_i} \left|\psi_i\right\rangle_A \otimes \left|i\right\rangle_B \right) \left(\sum_j \sqrt{p_j} \left\langle\psi_j\right|_A \otimes \left\langle j\right|_B \right) \right] \\
&= \mathrm{tr}_B \left[\sum_{i,j} \sqrt{p_i p_j} \left|\psi_i\right\rangle_A \left\langle\psi_j\right|_A \otimes \left|i\right\rangle_B \left\langle j\right|_B \right] \\
&= \sum_{i,j} \sqrt{p_i p_j} \left|\psi_i\right\rangle_A \left\langle\psi_j\right|_A \mathrm{tr}\left(\left|i\right\rangle_B \left\langle j\right|_B \right)
\end{aligned}$$

Recall that the $|i\rangle, |j\rangle$ states are in the computational basis and are therefore orthogonal to each other when they are not equal. Thus only when $i = j$ will the trace be non-zero (and in that case, it will simply be 1). Continuing with this in mind, we obtain

$$\begin{aligned}
\mathrm{tr}_B \left|\psi\right\rangle\left\langle\psi\right| &= \sum_i \sqrt{p_i p_i} \left|\psi_i\right\rangle_A \left\langle\psi_i\right|_A \mathrm{tr}\left(\left|i\right\rangle_B \left\langle i\right|_B \right) \\
&= \sum_i p_i \left|\psi_i\right\rangle_A \left\langle\psi_i\right|_A \\
&= \rho.
\end{aligned}$$

In class we made an argument for non-uniqueness by showing we could preserve the partial trace under multiplication of the subsystem B by a unitary matrix. I don't want to write more equations so I will instead make an analogous argument without equations.

The representation $|\psi\rangle$ being unique implies something morally wrong: that the choice of computational basis vectors $|i\rangle$ was somehow significant. And it shouldn't be since we're picking them such that we when we trace them out we're left with ρ. The easiest way to think about this is to consider what should happen if instead of pairing $|\psi_i\rangle$ with $|i\rangle$, we paired it with $|k-i-1\rangle$, where k is the number of states in the mixed state. This should make no difference - otherwise this implies that mixed states have some sort of ordering enforced. Indeed, we just need the subsystem B to be (part of) an orthonormal basis so that $|i\rangle\langle j|$ zeroes out for $i \neq j$ and the trace of $|i\rangle\langle i|$ is 1. As it so happens, the columns of a unitary matrix form an orthonormal basis, so this is saying the same thing we said in class in just a wordier (but equation-less) way.

1.2 Why don't we care about systems that preserve the ℓ_p-norm for $p > 2$?

It can be shown that all matrices which preserve the ℓ_p-norm ($p > 2$) are matrices A such that $A = PD$, where P is a permutation matrix and D is a diagonal matrix. So these matrices can only swap around values and scale them.

I mean I think that's kind of cool so I'm not trying to diss the PD matrices, but in lecture it was said that they are uninteresting.

1.3 Why do we use $\mathbb{C}$ over $\mathbb{R}$?

We don't *have* to, but it's nicer. We can represent $\mathbb{C}$ using 2×2 matrices in $\mathbb{R}$.

One argument given for why it's nicer to use $\mathbb{C}$ is that we can take "fractions" of unitary matrices, which is impossible for some real matrices.

Consider for example the matrix

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

The matrix

$$V = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

is such that $V^2 = U$. However, it's impossible for such a matrix to exist with only real-valued entries. Observe that $\det U = -1$. If a matrix V' were such that $V'^2 = U$, then $\det (V')^2 = \det U = -1$ which implies that $\det V' = i$ which is a contradiction.

2 The Quantum Circuit Model

2.1 Types of quantum computers

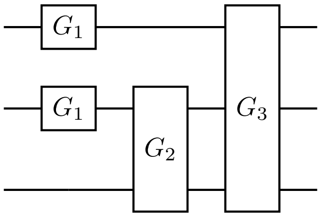
There were four models of quantum computers mentioned in lecture:

1. Circuit Model (this lecture)
2. Quantum Turing Machines, the quantum analog to Turing Machines
3. Measurement-based quantum circuits, where you prepare a complex initial state and *only* take measurements on it
4. Adiabatic computation, which will not be discussed much

2.2 An introduction to gates

Much like how classical logic gates take in some input bits and produce output bits, quantum gates take in some input qubits and produce output qubits. Unlike their classical analogs, all of the gates discussed in this lecture have the same number of output qubits as input qubits. In contrast to classical gates, which can be thought of as producing an (often single) output from several input bits, quantum gates can be instead thought of as evolving an input subsystem of qubits.

Let's look at how gates are visually rendered.



In the above diagram, G_1 is a gate on a single qubit, G_2 is a gate on two qubits, and G_3 is a gate on three qubits.

An entire system of gates is simply a $2^n \times 2^n$ unitary matrix, where n is the number of qubits.

What would this unitary matrix look like for the above diagram?

$$U = G_3 \left(I \otimes G_2 \right) \left(G_1 \otimes G_1 \otimes I \right).$$

A gate-less wire is simply treated as the identity matrix, which makes sense because it doesn't do anything to its qubit.

2.3 Some common gates

2.3.1 CNOT: Controlled NOT

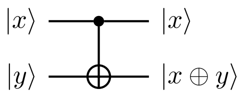
The CNOT gate is a "classical reversible" gate on 2 qubits: it is a gate whose classical inputs you can deduce from its classical outputs. It is defined on classical basis states by the following map:

$$\begin{aligned} |00\rangle &\mapsto |00\rangle \\ |01\rangle &\mapsto |01\rangle \\ |10\rangle &\mapsto |11\rangle \\ |11\rangle &\mapsto |10\rangle \,. \end{aligned}$$

As a matrix, it is

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

On a circuit diagram it appears as follows



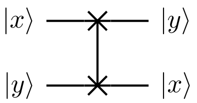
The $x \oplus y$ is the XOR symbol (or addition modulo 2) and represents what happens to classical bits if given as input.

2.3.2 SWAP

The SWAP gate is another "classical reversible" gate on 2 qubits. It swaps its (classical) bits. i.e., it maps:

$$|xy\rangle \mapsto |yx\rangle \,.$$

It looks like the following



and its matrix form is

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

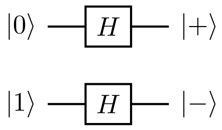
2.3.3 H(adamard)

The Hadamard gate (called H) is a single qubit gate which maps the $|0\rangle$ and $|1\rangle$ states to $|+\rangle$ and $|-\rangle$, respectively. Formally,

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} := |+\rangle$$
$$H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} := |-\rangle$$

.

It looks like the following



and its matrix is

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

We will see more of $|+\rangle$ and $|-\rangle$ in this class: it is worth noting that they form a basis for $\mathbb{C}^2$ (although this is perhaps made obvious by the fact that H is unitary).

2.3.4 Phase gates: S and T

S and T are single-qubit phase gates and the first complex-valued gates we will introduce. We define them by their matrix forms:

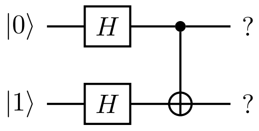
$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}.$$

For those geometrically-inclined, these may be thought of as rotations about the z axis of the Bloch Sphere of $\frac{\pi}{2}$ and $\frac{\pi}{4}$ radians, respectively, hence why they are referred to as phase gates. Note that, consistent with this observation, $S = T^2$.

In class, we used the term "Phase gate" to refer to just the S gate.

2.4 Examples

In class, we considered the example of the following circuit



One way of computing what this circuit does would be to compute the unitary matrix that is $\text{CNOT}(H \otimes H)$, but it is often more helpful to instead look at how the system evolves its input qubits.

After the Hadamard gates, our system looks like $|+\rangle \otimes |-\rangle$, which is in terms of basis states

$$|+\rangle \otimes |-\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

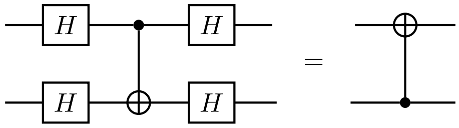
We know what the CNOT does to the standard computational basis, so we can FOIL the above states to get the standard basis and then apply the CNOT to each.

$$\begin{aligned}
\text{CNOT}\left(|+\rangle \otimes |-\rangle\right) &= \text{CNOT}\left(\frac{|00\rangle}{2} + \frac{|10\rangle}{2} - \frac{|01\rangle}{2} - \frac{|11\rangle}{2}\right) \\
&= \frac{\text{CNOT}(|00\rangle)}{2} + \frac{\text{CNOT}(|10\rangle)}{2} - \frac{\text{CNOT}(|01\rangle)}{2} - \frac{\text{CNOT}(|11\rangle)}{2} \\
&= \frac{|00\rangle}{2} + \frac{|11\rangle}{2} - \frac{|01\rangle}{2} - \frac{|10\rangle}{2}
\end{aligned}$$

You can convince yourself pretty easily by FOILing that this is equal to $|-\rangle \otimes |-\rangle$.

This is an example of how a gate that leaves its first qubit alone classically can affect its first qubit if its input states aren't classical.

If we investigate the other possible inputs, we can see that the following identity is true:



2.5 Universality

Universality captures the notion that a small set of gates can be used to generate all possible unitary transformations. The most straightforward type of universality is the following:

Definition 1 (Universal)**.** *We say that a set of gates is* universal *if for all unitaries U there is a circuit built from gates in the set that is equivalent to U.*

Theorem 2. *The Clifford gates (*$\text{CNOT}, \text{SWAP}, H,$ *and S) are not universal.*

In fact, the Clifford gates can be simulated in polynomial time.

However, the Clifford gates with the addition of the T-gate *are* universal in that sense that every unitary can be closely approximated (this notion will be formalized in the next section). In fact, the Clifford gates, with the addition of any unitary operator that can not be derived from the Clifford gates, immediately become universal.

2.6 How many gates are needed?

If we have a complete set of gates, a reasonable question we might ask is, "How many gates are necessary to compute an arbitrary U?" Formally, for some unitary U, let $G_k G_{k-1} \cdots G_2 G_1 = U$, where each G_i is a gate in some given set of gates. What is the k bounded by?

We can make a combinatorial argument based on the number of parameters in the unitary matrix and in the gates (we assume the gates are on a fixed, constant number of qubits) that we need approximately $\Omega(4^n)$ gates.

2.6.1 Approximations

We might also ask how many gates are necessary to approximate an arbitrary U. Formally, for some $\epsilon > 0$ and unitary U, let $G'_k G'_{k-1} \cdots G'_2 G'_1 = V$, where the G'_i are in some given set of gates and

$$\sup_{\psi} \| (U - V) \ket{\psi} \|_2 \leq \epsilon$$

.

Theorem 3 (Solovay-Kitaev [Kit97]). *You can approximate an arbitrary n-qubit unitary U to a precision of ϵ with $\mathcal{O}(4^n \mathrm{polylog}(\frac{1}{\epsilon}))$ gates.*

2.7 The quantum circuit model, formally

Note: as I recall, this was to be elaborated on in the following lecture. So you may be better served reading those lecture notes.

Informally, there are three stages to the circuit model:

1. Prepare qubits
2. Run the circuit on them
3. Measure resulting qubits

We discussed a lot of what is entailed by 2., but not much of the other stages in this lecture.

Formally, the computation model is defined through the help of a classical Turing machine. In particular, to evaluate a function $f : \{0,1\}^n \to \{0,1\}$, the classical Turing machine M reads the input x, and outputs the *specification* of a quantum circuit C. Then, we run the quantum circuit C on the state $\ket{x} \otimes \ket{0 \ldots 0}$ where the zero register are ancillary work qubits. The first qubit is then measured in the end and should give the desired output $f(x)$ with high probability.

Additionally, we require M must fall in some reasonable complexity class, such as BPP. This indirect definition enforces a constraint on the quantum circuits in our model that prevents them from simply hard-coding the results of $f(x)$. The Turing Machine is also limited enough such that the quantum circuit needs to do some sort of non-trivial computation. An analogous definition is used for classical circuits to avoid similar issues.

References

[Kit97] A. Y. Kitaev. Quantum computations: algorithms and error correction. *Russ. Math. Surv.*, 52(6):1191–1249, 1997.