CS 593/MA 592 - Intro to Quantum Computing
Spring 2024
Thursday, February 1 - Lecture 4.2

Today's scribe: Anirudh Rao

Reading: Subsections 4.5, 4.6 and Appendix 3 of Nielsen and Chuang.

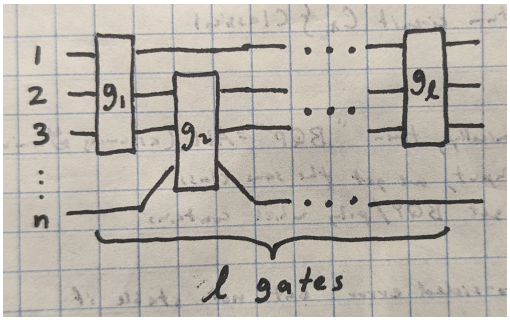
Agenda:

1. Counting Circuits
2. BQP & Solovay-Kitaev

1 Counting Circuits

Fix a gate set $\mathcal{G}$ and let $|\mathcal{G}| = c$. Let's upper bound the number of different unitaries that we can build out of circuits on n qubits of length l (that is, using l gates).

For convenience, let's assume all $g \in \mathcal{G}$ are k-ary. That is, $g : (\mathbb{C}^2)^{\otimes k} \to (\mathbb{C}^2)^{\otimes k}$. Let's furthermore assume that any time $g \in \mathcal{G}$, we have that all gates we get from permuting the input/output wires for g are also in $\mathcal{G}$. This means anytime we want to act with some gate on a choice of k qubits, we might as well assume the qubits enter the gate in a way that respects the ordering of their indices. Such circuits look like this (where any crossings of wires should be ignored):



Then clearly we can upper bound the number of such circuits by:

$$c\binom{n}{k} \cdot c\binom{n}{k} \cdot \cdots \cdot c\binom{n}{k} = c^l \binom{n}{k}^l = O((nc)^{kl}).$$

If our gate set doesn't satisfy the properties we imposed, then we can always make $\mathcal{G}$ bigger to get it to. Thus this upper bound works for all gate sets.

However, simply counting the number of unitaries we can implement *exactly* is not as interesting/pertinent as counting the number of unitaries we can *approximately* implement for a given $\varepsilon > 0$.

As a warm-up, let's consider the following: how big does l need to be for us to be able to guarantee that for any state $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n}$, we may build a circuit C so that $|C|0\cdots 0\rangle - |\psi\rangle| < \varepsilon$? Assume $|\psi\rangle$ is normalized. The set of all such states is the unit sphere $S^{2^{n+1}-1} \subseteq (\mathbb{C}^2)^{\otimes n}$.

$$\left\{ |\psi\rangle = \sum_{l=0}^{2^n-1} z_l |l\rangle \;\middle|\; \sum_{l=0}^{2^n-1} |z_l|^2 = 1 \right\} = S^{2^{n+1}-1}$$

Our question reduces to the following: What is the minimum number of points on $S^{2^{n+1}-1}$ required to guarantee that every point on $S^{2^{n+1}-1}$ is within ε of one of these points?

$$N \geq \frac{\text{Area}(S^{2^{n+1}-1})}{\text{Area(little, spherical disk } D_\varepsilon^{2^{n+1}-1} \text{ of radius } \varepsilon)} = \Omega\left(\left(\frac{1}{\varepsilon}\right)^{2n}\right)$$

Thus, we need circuits of length $l = \Omega\left(\frac{2^n}{\log n}\log\left(\frac{1}{\varepsilon}\right)\right)$ to guarantee there exists a circuit C with $|C|00\cdots0\rangle - |\psi\rangle| < \varepsilon$ for every $|\psi\rangle$.

In particular, you can now easily argue that that for all $\varepsilon > 0$, there exists a unitary $U : (\mathbb{C}^2)^{\otimes n} \to (\mathbb{C}^2)^{\otimes n}$ such that the smallest circuit C that satisfies $\text{Error}(C,U) < \varepsilon$ has length $\Omega\left(\frac{2^n}{\log n}\log\left(\frac{1}{\varepsilon}\right)\right)$.

2 BQP & Solovay-Kitaev

BQP is notation for the complexity class "Bounded error Quantum Polynomial time." BQP is a complexity class of decision problems (a.k.a. "yes-no questions")

$$L : \{0,1\}^* = \bigcup_{l\geq 0}\{0,1\}^l \to \{0,1\} = \{\text{NO}, \text{YES}\}.$$

A priori, BQP depends on two parameters: some number $0 \leq \delta \leq 1$ and a gate set $\mathscr{G}$.

Definition. *$L \in \text{BQP}(\mathscr{G},\delta)$ if there exists a **classical (deterministic) polynomial-time algorithm** which for each bit string $x \in \{0,1\}^n$ outputs a description of a **quantum circuit** C_x over $\mathscr{G}$ such that measuring the first qubit of $C_x|0\cdots0\rangle$ in the computational basis satisfies*

$$prob(\text{Output}(C_x) = L(x)) \geq 1 - \delta.$$

The $1-\delta$ is the "bounded" part or the "B" of BQP, the quantum circuit C_x is the "quantum" part or the "Q" of BQP, and the classical polynomial-time algorithm is the "polynomial" part or the "P" of BQP.[1] This is formalizing the following work flow:

1. Given x.
2. Think hard (but not too hard) about how to build a helpful quantum circuit C_x. (classical part)
3. Use your quantum computer to apply C_x to $|0\cdots0\rangle$. (quantum part)
4. Reduce from "quantum data" to "classical data" (i.e. YES or NO) by measuring in the computational basis.

If we don't insist that our circuits C_x are prepared algorithmically, then we might call the complexity class $\text{BQ}(\mathscr{G},\delta)$. But this is a bad class, because $\text{BQ}(\mathscr{G},\delta)$ = ALL (assuming $\mathscr{G}$ is universal), where ALL is the complexity class consisting of all decision problems. Of course, ALL contains uncomputable problems (in fact, it contains problems that are strictly harder than the halting problem!). Thus, the insistence that our circuits are prepared algorithmically is important, and often described as a requirement of *uniformity* in the quantum circuits we use to solve the problem L.

One might wonder: what if we used a classical *probabilitistic* algorithm to prepare the circuits C_x? Well, we get the same class. This is because we can put all of the classical randomness we might use to prepare C_x into the quantum circuit we prepare. (Assuming we have gates in $\mathscr{G}$ that are able to implement classical coin flips.) So we don't get anything new.

If we insist C_x only depends on the length of the bit string

$$|x| = |(x_1,\ldots,x_l)| = l$$

[1] In more etail: since it takes us at most a polynomial amount of classical thinking to decide which quantum calculation C_x to do, then C_x is at most polynomially large, and thus can be run on our quantum computer in a polynomial amount of time!

and we use $|x_1 \cdots x_l 0 \cdots 0\rangle$ as input instead of $|0 \ldots 00 \ldots 0\rangle$, then we also get the same class.

However, if we drop the algorithmic assumption on *finding* such a circuit that only depends on $|x|$, we get the class BQP/poly, or "quantum polynomial time with classical advice." While not equal to ALL, BQP/poly still contains uncomputable problems.

What is δ doing? BQP($\mathscr{G}$, 0) is a more-or-less sensible complexity class of "zero-sided error" quantum polynomial time algorithsm. However, the property of having zero-sided error is not stable if we change $\mathscr{G}$. In particular, it is unreasonable to expect zero-sided error in a setting where we have to perform error correction. In fact, it is unreasonable to expect one-sided error (on either side).

On the other hand, for $\delta \geq 1/2$, BQP($\mathscr{G}, \delta$) = ALL. You'll show this on your homework 4.

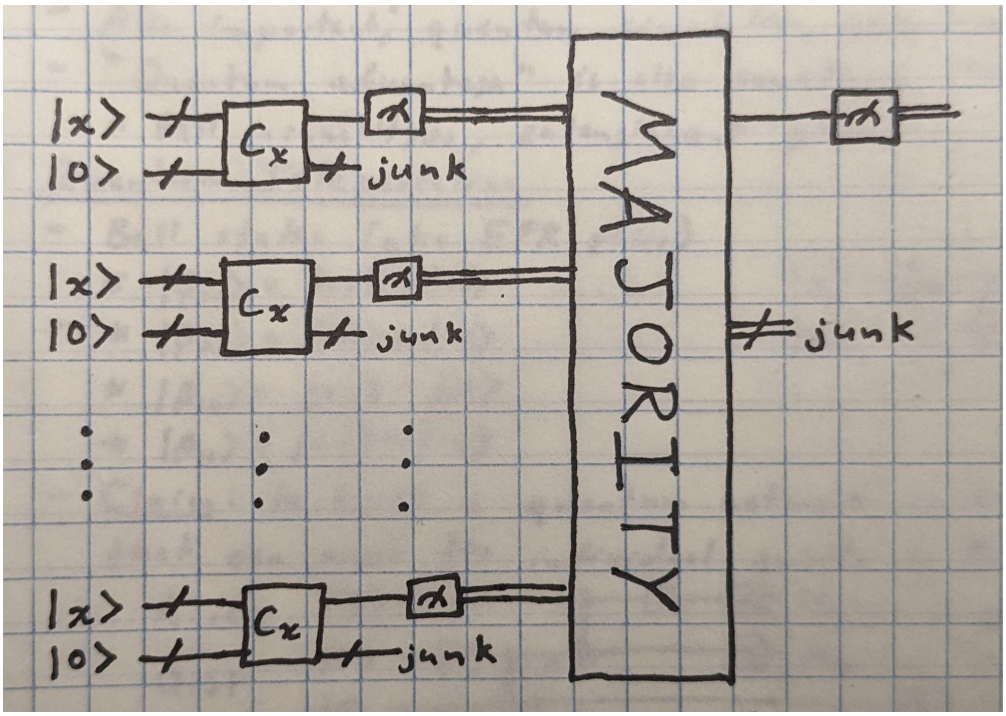
So $0 < \delta < \frac{1}{2}$ seems most appropriate. In fact, once we're in this range, something nice happens:

Lemma 1. *If $0 < \delta \leq \delta' < \frac{1}{2}$ and $\mathscr{G}$ is able to implement a "majority" circuit of arbitrary width, then* BQP($\mathscr{G}, \delta$) = BQP($\mathscr{G}, \delta'$)

Proof. BQP($\mathscr{G}, \delta$) $\subseteq$ BQP($\mathscr{G}, \delta'$) is free. To obtain the other containment, we need to "amplify the success probability."

The basic idea is simple. Compute $C_x|0 \cdots 0\rangle$ several times, take a survey of all the output bits from each run, and then elect the majority. The Chernoff bound shows that this works.

The technical thing to think through is how to implement this majority vote with a single quantum circuit. Indeed, strictly speaking, our definition of BQP does not allow for any "classical post-processing" of our measurements. A circuit like this is close to doing the trick, but you should think about what details need to be modified to finish the argument:



□

The previous lemma shows that when $\mathscr{G}$ is universal (really, "universal enough to implement majority" suffices), the definition of $BQP(\mathscr{G}, \delta)$ gives the same class for all $0 < \delta < 1/2$. To resolve the question of dependency on the choice of gate set $\mathscr{G}$, we will use the Solovay-Kitaev theorem.

First, a definition. The *special unitary group* $SU(d)$ is the group of all $d \times d$ unitary matrices with determinant 1:

$$SU(d) = \{U \in U(d) | \det U = 1\} \leq U(d).$$

The determinant 1 condition is kind of like saying unitaries with "no global phase."[2]

Theorem 2 (Solovay-Kitaev Theorem)**.** *Let $\mathcal{G} \subset SU(\alpha)$ be a finite subset of gates that is closed under inversion ($g \in \mathcal{G} \implies g^{-1} \in \mathcal{G}$) and densely generates. Then there exists a (classically efficient) algorithm that takes any "explicit" $U \in SU(\alpha)$ and any $\varepsilon > 0$ to a word $w = g_l g_{l-1} \cdots g_2 g_1$, $g_i \in \mathcal{G}$ of length $l = O(\log(\frac{1}{\varepsilon})^{\alpha})$ for some constant $\alpha > 1$ and* $\text{Error}(w,U) < \varepsilon$.

The polynomial dependence on $\log(1/\varepsilon)$ is actually not important for our next result, but it is important in the context of error correction and fault tolerance, since it says we get only polylogarithmic overhead in certain error corection procedures.

Corollary 3. *Given $0 < \delta < \frac{1}{2}$, a universal $\mathcal{G}$ that is inverse closed and any other gate set $\mathcal{G}'$:* $\text{BQP}(\mathcal{G},\delta) \geq \text{BQP}(\mathcal{G}',\delta')$ *for some $0 < \delta' < \frac{1}{2}$.*

Definition. *Fix any universal gate set $\mathcal{G}$ (that is inverse closed) and $0 < \delta < \frac{1}{2}$. Then* $\text{BQP} := \text{BQP}(\mathcal{G},\delta)$.

[2]Note that $SU(d) \neq PU(d)$. Indeed, $PU(d)$ is the *quotient* of $U(d)$ we get by identifying two matrices that differ by a global phase.