CMSC 858L: Quantum Complexity
Unit 2: Universal and Non-Universal Quantum Gate Sets

Instructor: Daniel Gottesman

Fall 2026

1 Universal Gate Sets

Recall:

Definition 1. *Let $\mathcal{G}$ be a set of quantum gates acting on a bounded number of qubits (usually either 2 or 3).*

- *$\mathcal{G}$ is a* universal *set of gates if for any unitary U, there exists a circuit (of any size) containing gates from $\mathcal{G}$ that realizes the transformation U.*
- *$\mathcal{G}$ is* approximately universal *if, for any unitary U and any error ϵ, there exists a circuit composed of gates from $\mathcal{G}$ that realizes a unitary U_ϵ such that $\|U - U_\epsilon\|_{sup} < \epsilon$.*

The choice of norm here is not very important — while different norms will give different values, if one of them can be arbitrarily small, other norms will be as well. The "sup" norm I used here is defined as

$$\|M\|_{sup} = \max_{|\psi\rangle} |\langle\psi|M|\psi\rangle|, \tag{1}$$

or equivalently, the largest eigenvalue of M (also called the "infinity norm"). It quantifies the worst-case behavior of the approximation U_ϵ.

The standard example of a universal set of gates is $\mathcal{G} = \{\text{single-qubit gates}, CNOT\}$. Two standard examples of approximately universal sets of gates are $\mathcal{G} = \{H, R_{\pi/8}, CNOT\}$ and $\{H, R_{\pi/4}, Tof\}$. Here $CNOT$ is the controlled-NOT, H is the Hadamard transform, R_θ is the diagonal phase gate $R_\theta|0\rangle = e^{-i\theta}|0\rangle$, $R_\theta|1\rangle = e^{i\theta}|1\rangle$, and Tof is the 3-qubit Toffoli gate (also called the controlled-controlled-NOT). (Often, $R_{\pi/4}$ is called S and $R_{\pi/8}$ is called T.)

One advantage of dealing with approximately universal gate sets is that we can now use a finite gate set. Note that a finite gate set can only ever be approximately universal and not exactly universal, since the number of circuits with a finite gate set is countable whereas the set of all unitaries is uncountable.

Also, I will largely be discussing universality of unitary gate sets. It is also possible and sensible to allow measurements in the middle of a circuit and adapting later quantum gates depending on the result of the measurements. For instance, this capability is extremely useful in fault-tolerant quantum computation and therefore many platforms have developed the experimental capability of doing mid-circuit measurements. From a complexity-theoretic point of view, mid-circuit measurements are not stronger than a universal or approximately universal set of quantum gates, as you will explore in the problem set. However, if the adaptation to the measurement results involves a classical computation, we should count classical gates in the same way as we count quantum gates. In addition, mid-circuit measurement and adaptive quantum circuits can make weaker gate sets universal when paired with appropriate initial states or measurements.

1.1 Approximating different gate sets

We said earlier that the definition of BQP doesn't depend on which universal gate set we pick. This is a consequence of the Solovay-Kitaev theorem, which says that any approximately universal gate set can not just approximate any unitary to any desired accuracy $\epsilon > 0$, but can do so with a number of gates that is only polylogarithmic in $1/\epsilon$. Moreover, the theorem is constructive.

Theorem 1 (Solovay-Kitaev)**.** *Let $\mathcal{G}$ be an approximately universal set of gates. Then for any unitary V in a fixed Hilbert-space dimension D, there exists a classical algorithm to find, for any $\epsilon > 0$, a quantum circuit of size $O(\text{poly}(\log(1/\epsilon)))$ which realizes U_ϵ such that $\|V - U_\epsilon\| < \epsilon$. The classical algorithm runs in time $O(\text{poly}(\log(1/\epsilon)))$ as well.*

The choice of norm is again not very important, although the sup-norm is a reasonable choice.

The original version of the Solovay-Kitaev theorem requires that $\mathcal{G}$ is closed under inverses, but there is a later improvement that removes that requirement. The proof I will sketch is the original proof. It is also worth noting that certain gate sets (such as $\{H, R_{\pi/8}\}$) can achieve this approximation more efficiently (i.e., with a lower exponent of the logarithm) than the general case.

Also, very importantly, we are fixing the Hilbert space dimension. While the theorem works for any dimension, the number of gates needed is certainly exponential in the number of qubits. The algorithm is only efficient in the accuracy needed.

Proof sketch.

The first step of the proof is to get a very rough approximation, with a constant degree of accuracy ϵ_0. We can discover how to do this by simply trying out all the possible circuits up to a certain (constant) size. There are a constant number of such circuits. Because the gate set is universal, *eventually* we will get ϵ_0-close to every unitary. The size of circuit needed to do this will be dependent on the exact gate set, but it is independent of the eventual target accuracy ϵ, which hasn't shown up yet.

Define an η-net S for a subset $B \subseteq SU(D)$ to be a set such that for all $V \in B$, $\exists U \in S$ such that $\|U - V\| < \eta$. (Here $SU(D)$ is the special unitary group of dimension D, the set of all $D \times D$ unitary matrices of determinant 1; unitaries always have determinant with absolute value 1, so any unitary is related to an element of $SU(D)$ up to a global phase, which is physically irrelevant.) So this first stage of the proof is to create an ϵ_0-net for $SU(D)$.

Given the target V for the theorem, we can therefore pick a U_0 from the ϵ_0-net such that $\|V - U_0\| < \epsilon_0$. Let $V_1 = VU_0^{-1}$. Then

$$\|V_1 - I\| = \|(V - U_0)U_0^{-1}\| = \|V - U_0\| < \epsilon_0 \tag{2}$$

since the distance measures we are likely to use are invariant under unitary rotations.

Our next task is to find a good approximation U_1 to V_1. Suppose $\|V_1 - U_1\| < \epsilon_1 < \epsilon_0$. Then

$$\|V - U_1U_0\| = \|V_1U_0 - U_1U_0\| = \|V_1 - U_1\| < \epsilon_1, \tag{3}$$

again using invariance of the distance under unitaries, so we have found a better approximation to V. We keep doing this with smaller and smaller ϵ_i to find the desired circuit.

So how to do we find this better approximation? Suppose we have an ϵ_i-net S_i in a ball of radius δ_i around the identity. We want to use this to find a ϵ_{i+1}-net S_{i+1} in a ball of radius δ_{i+1} around the identity, with $\epsilon_{i+1} < \epsilon_i$ and $\delta_{i+1} < \delta_i$.

In the case of unitaries, we write $U = e^{-i\eta H} \approx I - i\eta H$, where H is Hermitian (i.e., $H = H^\dagger$) and traceless ($\text{Tr}\, H = 0$). Suppose we have $U_1 = e^{-i\eta H_1}$ and $U_2 = e^{-i\eta H_2}$. By the Baker-Campbell-Hausdorff lemma,

$$U_1U_2 = e^{-i\eta(H_1+H_2)-\eta^2[H_1,H_2]/2+O(\eta^3)}, \tag{4}$$

with $[H_1, H_2] = H_1H_2 - H_2H_1$. In particular, when η is small, we can reasonably approximate the product $U_1U_2 = e^{-i\eta(H_1+H_2)+O(\eta^2)}$, whereas the group commutator

$$U_1U_2U_1^\dagger U_2^\dagger = e^{-\eta^2[H_1,H_2]+O(\eta^3)}. \tag{5}$$

Lemma 1. *The set of unitaries $U_1U_2U_1^\dagger U_2^\dagger$ where U_1 and U_2 are arbitrary elements of S_i form an $O(\epsilon_i\delta_i)$-net in a ball of radius $O(\delta_i^2)$ provided $\epsilon_i < \delta_i$ and $\epsilon_i = \Omega(\delta_i^2)$.*

Proof of lemma. Suppose we have an element W in a ball of radius $O(\delta_i^2)$. Then we can write $W = e^{-i\delta_i^2 H}$ and pick traceless Hermitian H_1, H_2 such that $\delta_i^2 H = -i[\delta_i H_1, \delta_i H_2] + O(\delta_i^3)$. (This is not obvious, but follows from properties of the Lie algebra for $SU(D)$.). Since $e^{-i\delta_i H_1}$ and $e^{-i\delta_i H_2}$ are in a ball of radius δ_i around the identity, we can approximate them with elements of S_i: U_1 and U_2. We have $U_j = e^{-i\delta_i H_j'}$, with $H_j' = H_j + O(\epsilon_i)$ $(j = 1, 2)$. Then

$$U_1U_2U_1^\dagger U_2^\dagger = e^{-\delta_i^2[H_1', H_2'] + O(\delta_i^3)} \tag{6}$$

$$= e^{-\delta_i^2[H_1, H_2] + O(\epsilon_i\delta_i) + O(\delta_i^3)} \tag{7}$$

$$= e^{-i\delta_i^2 H + O(\epsilon_i\delta_i) + O(\delta_i^3)} \tag{8}$$

$$= W + O(\epsilon_i\delta_i) + O(\delta_i^3). \tag{9}$$

When $\epsilon_i = \Omega(\delta_i^2)$, we therefore have an approximation of W to an accuracy $O(\epsilon_i\delta_i)$. □

It's actually possible to get a somewhat tighter error bound for ϵ_i than in this lemma due to cancellations in the commutator. Unfortunately, the net does not cover enough area for our purposes, so we need to expand it to cover a larger ball using the same approximation technique we are planning to use for V (shift to near I and then approximate).

Putting all of these components together, we can get finer and finer nets with $\epsilon_i = O(\epsilon_{i-1}^c)$ for some constant c, so $\epsilon_i = O(\epsilon_0^{c^i})$. Each net uses more gates, however, by a constant factor d. To achieve the desired approximation ϵ, then, we need to have $c^i = \log\epsilon / \log\epsilon_0 = O(\log(1/\epsilon))$ (since $\epsilon_0 < 1$) and a total number of gates

$$d^i = O(\log(1/\epsilon)^{\log d/\log c}) = O(\text{poly}(\log(1/\epsilon))). \tag{10}$$

The standard approach gives $c = 3/2$ and $d = 5$, so the exponent is about 4, but this can be tightened.

□

One immediate consequence of the Solovay-Kitaev theorem is that if we have two different universal gates, we can convert a circuit written using one such gate set to the other one with minimal overhead. In particular, suppose $C_{\mathcal{G}}$ is a circuit of size T using gates from gate set $\mathcal{G}$. In order to rewrite this circuit using gates from gate set $\mathcal{H}$, we should replace each gate in $C_{\mathcal{G}}$ with an approximation from $\mathcal{H}$. We need an accuracy $\epsilon = O(1/T)$ so that the approximate circuit is close enough to the original. Therefore, each gate from $\mathcal{G}$ gets replaced with $O(\text{poly}\log(T))$ gates from $\mathcal{H}$. Since the gates from $\mathcal{G}$ act on a bounded number of qubits, we don't have to worry about the dimension factors in this approximation. The new circuit thus has $O(T\text{poly}\log T)$ gates, which is certainly polynomial in T. While the polylogarithmic scaling of Solovay-Kitaev is not necessary for defining BQP (polynomial in $1/\epsilon$ would have sufficed), it *is* necessary for getting meaningful polynomial speedups, such as Grover's algorithm, from a quantum computer.

2 Strong and Weak Simulations

What if instead of a universal gate set, we take some non-universal gate set? There are a lot of different choices, and now they are not all equivalent. So let us think about polynomial-size quantum circuits built from some different sets $\mathcal{G}$ which are not universal. It turns out that there are a lot of different variations and even small changes in the rules can produce different results. For the moment, let us assume that we always initialize the circuit with states in the standard basis, measure at the end of the computation in the standard basis, and do not allow intermediate measurements.

As a first example, consider the gate set $\mathcal{G} = \{X, CNOT, Tof\}$. What complexity class do we get from polynomial circuits with this gate set?

Answer: P! These are all classical gates and indeed, they form a universal set of reversible classical gates. Since every classical computation can be converted to a reversible computation, this allows everything in P.

Note that we don't even have any way to introduce randomness, so we have P, not BPP. But if we allowed some qubits to be initialized in the state $|+\rangle = |0\rangle + |1\rangle$, then we could use those as random bits and we would get BPP. (This is a simple example of what I mean when I say that small rule changes can give different results.)

This is also a simple example showing that non-universal quantum gate sets can have a computational power of P or less. But what exactly does it mean to have a computational power of "P or less"? One way to be unambiguously sure of this is through the ethos of reduction — if polynomial-size classical circuits can decide any problem that can be decided using the gate set, then P is stronger. But decision problems are perhaps too restrictive here. Even restricting to classical inputs and classical outputs, it might be true that when you measure many qubits at the end of the circuit, there are correlations between the output bits that reveal some computationally hard question. So it makes sense to look instead at sampling problems.

For sampling problems, we capture the notion that a gate set is weaker than P through the idea of *simulation*. However, there are a variety of different kinds of simulation. The two most common are known as *strong simulation* and *weak simulation*. A strong simulation calculates the marginal probabilities of outcomes of the quantum circuit, whereas a weak simulation produces a sample from the output probability distribution.

Definition 2. *Given a specification of a random variable X (for instance as the outcome distribution of a given quantum circuit), we say a sampling algorithm provides a* weak simulation *of X if it outputs a random variable Y which is close to X in one of the following senses:*

a. Exact weak simulation: $Y = X$.

b. Weak simulation with multiplicative error ϵ: for all outcomes x,

$$(1-\epsilon)\text{Prob}(X=x) \leq \text{Prob}(Y=x) \leq (1+\epsilon)\text{Prob}(X=x). \tag{11}$$

c. Weak simulation with additive error ϵ:

$$\frac{1}{2}\sum_x |\text{Prob}(X=x) - \text{Prob}(Y=x)| \leq \epsilon. \tag{12}$$

A strong simulation *is a function problem. The input is a specification of a random variable X, a set S of bit labels, and a value x for the bits in S. The output is* $\text{Prob}(X|_S = x)$ *(i.e., the marginal probability that variable X is equal to x on the bits in S) for an exact strong simulation, or a value close to this (for instance in either the multiplicative or additive sense above) for an approximate strong simulation.*

Note that a quantum circuit provides an exact weak simulation of itself. It doesn't provide a strong simulation and it is not generally the case that a quantum circuit can exactly strongly simulate itself without exponentially many repetitions to fully map out the probability distribution.

On the other hand, suppose we have a single sample from the random variable Y that provides a weak simulation with additive error ϵ (the weakest notion of simulation we will consider here) for small ϵ. It is difficult to distinguish that from a single sample from X. If we have to guess, without any prior knowledge, we can only bias our guess by ϵ, and if the quantum circuit is computing some function or solving a decision problem, a sample from Y will have the same chance of success as X does up to ϵ. Thus, weak simulation with additive error is the most physically sensible notion of simulation. For this reason, fault-tolerant quantum circuits only attempt to provide a weak simulation of the ideal quantum circuit with additive error ϵ for sufficiently small ϵ.

Still, it can be helpful to talk about other kinds of simulations for a variety of reasons. A simulation with small multiplicative error is rather powerful because it must get approximately right the probability of even very rare outcomes. For this reason, the potential existence of a weak simulation with small multiplicative error can be useful for proving complexity-theoretic statements, as we will see when talk about quantum advantage. The existence of exact simulations and strong simulations is similarly indicative of something

about the power of the circuit class. Exact (or multiplicative) strong simulations can also be practically useful since they give us additional information about a quantum system beyond even what would be available from running the quantum circuit.

One thing that may not be obvious is that a strong simulation is definitely more powerful than a weak simulation.

Theorem 2. *Suppose we have an exact strong simulation of X. Then (given a source of randomness) we can efficiently produce an exact weak simulation of X.*

Proof. First, note that computing the marginal probabilities lets us also compute conditional probabilities conditioned on some of the bits:

$$\text{Prob}(X = x|X|_S = x_s) = \frac{\text{Prob}(X = x)}{\text{Prob}(X|_S = x_s)}. \tag{13}$$

Now let S_i be the set of bits $\{0, \ldots, i\}$. First compute $p_0 = \text{Prob}(X|_{S_0} = 0)$ and generate a random bit x_0 which is 0 with probability p_0 to be the first bit of random variable Y. (One technicality that I am ignoring in the theorem statement is that p_0 and the subsequent probabilities should be efficiently computable so that these random bits can be generated efficiently given a uniform source of randomness.) With this assignment, $\text{Prob}(Y|_{S_0} = x_0) = \text{Prob}(X|_{S_0} = x_0)$.

Next, given the outcome x_0 generated for the zeroth bit, compute $p_1 = \text{Prob}(X|_{S_1} = x_0 0|X|_{S_0} = x_0)$. That is, p_1 is equal the conditional probability that the first bit of X is 0 given that the zeroth bit was x_0. Then generate the second bit of Y according to p_1, so the conditional probability that the first bit of Y is 0 is the same as that of X. Let x_1 be x_0 followed by the newly generated bit. This means that

$$\begin{aligned}
\text{Prob}(Y|_{S_1} = x_1) =& \text{Prob}(Y|_{S_0} = x_0)\text{Prob}(Y|_{S_1} = x_1|Y|_{S_0} = x_0) && (14)\\
=& \text{Prob}(X|_{S_0} = x_0)\text{Prob}(X|_{S_1} = x_1|X|_{S_0} = x_0) && (15)\\
=& \text{Prob}(X|_{S_1} = x_1). && (16)
\end{aligned}$$

Continue this process, letting $p_i = \text{Prob}(X|_{S_i} = x_{i-1}0|X|_{S_{i-1}} = x_{i-1})$ and generating a new bit of Y according to p_i. Then let x_i be x_{i-1} with the new bit appended. Repeating the above argument and using induction, we find that $\text{Prob}(Y|_{S_i} = x_i = \text{Prob}(X|_{S_i} = x_i)$ for all i, so Y is an exact weak simulation of X. □

In the literature, sometimes you may encounter the term "strong simulation" as referring to computing not the marginal probability of outcomes, but instead the full probability of single outcomes (i.e., $\text{Prob}(X = x)$ for arbitrary n-bit strings x). Such a simulation is in many cases strictly weaker than a strong simulation as I have defined it here, and it is not even enough to guarantee being able to do a weak simulation without some additional assumptions about the distribution.

3 Clifford Group

OK, what about this set: $\mathcal{G} = \{H, CNOT, R_{\pi/4}\}$. If we replace $R_{\pi/4}$ by $R_{\pi/8}$ or $CNOT$ by Tof, this is approximately universal. But what about this gate set itself? What is its computational power?

The group generated by these gates is a finite group known as the *Clifford group*. Note that these gates can generate entangled states such as a Bell state $|00\rangle + |11\rangle$ or a GHZ state $|000\rangle + |111\rangle$. The group is also of practical importance since it is all that is needed to do encoding and error correction on the large class of stabilizer quantum error-correcting codes. Nevertheless, this gate set is not just not universal, but can actually be efficiently simulated on a classical computer:

Theorem 3. *There is a polynomial time classical algorithm that provides an exact strong simulation for any quantum circuit consisting of qubits initialized in the state $|0\rangle$, gates from the Clifford group, and ending with standard basis measurements of all qubits.*

Proof. The main insight needed for this theorem is that the gates in the Clifford group are exactly those which conjugate the Pauli group into itself. The Pauli group consists of tensor products of the Pauli matrices

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\ Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},\ Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{17}$$

with overall phase $\pm 1, \pm i$. For instance, $HXH^\dagger = Z$, $HZH^\dagger = X$ and $CNOT(X \otimes I)CNOT^\dagger = X \otimes X$.

We then note that the initial state of all n qubits in the state $|0\rangle$ is the $+1$ eigenstate of the Paulis $M_1 = Z_1$, $M_2 = Z_2$, ..., $M_n = Z_n$ (here Z_i means "Z acting on qubit i"), and is the unique state (up to global phase) with that property. We could thus equally well describe the initial state by listing the n operators $M_1, \ldots, M_n$ (generators of the *stabilizer* of the state). When we perform a gate U from the Clifford group, the state changes from $|\psi\rangle$ to $U|\psi\rangle$, and if it was a $+1$ eigenstate of M before, then

$$(UMU^\dagger)U|\psi\rangle = UM|\psi\rangle = U|\psi\rangle. \tag{18}$$

That is, the state $U|\psi\rangle$ is a $+1$ eigenstate of $UMU^\dagger$. Thus, if the state of the n-qubits is a $+1$ eigenstate of $M_1, \ldots, M_n$ before the gate, after the gate it is the $+1$ eigenstate of $UM_1U^\dagger, \ldots, UM_nU^\dagger$ and vice-versa. If it is the unique eigenstate before the gate, it is also the unique eigenstate after the gate.

The upshot is that we can uniquely specify the state throughout the circuit by updating the stabilizer: Whenever we perform a gate U, replace M_i with $UM_iU^\dagger$. When U is a general unitary, this might be a complicated thing, but if U is in the Clifford group and M_i is in the Pauli group, then $UM_iU^\dagger$ is also in the Pauli group, and therefore can be specified using just $2n+2$ bits: 2 bits for each of the n Paulis in the tensor product and 2 more for the overall phase. The full description of the state thus requires only $O(n^2)$ bits. Updating each M_i takes a constant time, since the only bits that need to be changed are those specifying the Paulis on the qubits acted on by the gate plus the bits specifying the global phase. Thus, simulating a single Clifford group gate takes time $O(n)$.

The measurement at the end is a little trickier. Measuring qubit i in the standard basis corresponds to measuring the eigenvalue of Z_i. If Z_i or $-Z_i$ is one of our generators M_j, then this is straightforward to compute, since if $M_j = Z_i$, the outcome for measuring qubit i will always be 0 (since the $+1$ eigenstate of Z_i is $|0\rangle$) and if $M_j = -Z_i$, the outcome for measuring qubit i will always be 1 (since the state is a $+1$ eigenstate of $-Z_i$, which means it is a -1 eigenstate of Z_i, namely $|1\rangle$). Also note that it is not possible that $\pm iZ_i$ is one of the generators M_j, since the state is a $+1$ eigenstate of M_j and $\pm iZ_i$ has eigenvalues $\pm i$.

But what if Z_i is not equal to a generator? One possibility is that $\pm Z_i$ is equal to a *product* of generators

$$\pm Z_i = \prod_{j=1}^{n} M_j^{b_j}, \tag{19}$$

with each b_j a bit. If Z_i satisfies this equation, then the state is a $+1$ eigenstate of Z_i as well:

$$Z_I|\psi\rangle = \prod_{j=1}^{n} M_j^{b_j}|\psi\rangle = |\psi\rangle, \tag{20}$$

since $|\psi\rangle$ is a $+1$ eigenstate of each M_j. The measurement outcome will then be 0. SImilarly, if $-Z_i$ satisfies (19), then the state is a -1 eigenstate of Z_i and the measurement outcome will always be 1.

We can find out if (19) holds by doing linear algebra. In particular, suppose we ignore the global phase for the moment and represent each M_j by a $2n$-bit vector $(\mathbf{x}|\mathbf{z})$. If x_k is the kth bit of $\mathbf{x}$ and z_k is the kth bit of $\mathbf{z}$, then the tensor factor Pauli of M_j acting on the kth qubit is

- I if $(x_k, z_k) = (0, 0)$,
- X if $(x_k, z_k) = (1, 0)$,
- Y if $(x_k, z_k) = (1, 1)$,

- Z if $(x_k, z_k) = (0, 1)$.

Note that if $(\mathbf{x}|\mathbf{z})$ is the binary vector corresponding to M and $(\mathbf{x}'|\mathbf{z}')$ is the binary vector corresponding to M', then $(\mathbf{x}+\mathbf{x}'|\mathbf{z}+\mathbf{z}')$ is the binary vector corresponding to MM'.

This means that (19) holds iff

$$(\mathbf{0}|\mathbf{e_i}) = \sum_{j=1}^{n} b_j(\mathbf{x}_j|\mathbf{z}_j). \tag{21}$$

Here, $\mathbf{e_i}$ is the vector which is 0 except in the ith coordinate, which is 1, and $(\mathbf{x}_j|\mathbf{z}_j)$ is the binary vector corresponding to M_j. This equation can be rewritten as

$$(\mathbf{0}|\mathbf{e_i})^T = M\mathbf{b}^T, \tag{22}$$

where $\mathbf{b}$ is the row vector of the b_j's and M is the $2n \times n$ matrix with columns equal to the $(\mathbf{x}_j|\mathbf{z}_j)$ vectors.

This is a system of linear equations over the binary field and can be solved by standard techniques, such as Gaussian elimination. If it has a solution, we find the values of b_j. This procedure also tells us if (19) does *not* hold.

Note, however, that we are not quite done with this case. We have found the b_j's but we do not yet know whether the measurement outcome is 0 or 1 because we dropped the global phase for this calculation. Now we must restore it, computing $\prod M_j^{b_j}$ in the Pauli group to see if we get Z_i or $-Z_i$.

What about if (19) does not hold? Actually, there is a shortcut we can use to determine that. Note that the initial generators $M_i = Z_i$ all commute with each other under multiplication, and when we conjugate them by U that is still true:

$$(UM_iU^\dagger)(UM_jU^\dagger) = UM_iM_jU^\dagger = UM_jM_iU^\dagger = (UM_jU^\dagger)(UM_iU^\dagger). \tag{23}$$

When we take the binary vector representations of the initial $M_i = Z_i$, the vectors we get are all linearly independent. This remains true after performing Clifford group gates because the gates are invertible; if P is a product of the $UM_iU^\dagger$s, then $U^\dagger PU$ is the same product of the M_i's.

Thus, the M_i's at all times are independent, commuting Pauli operators. It turns out that we can have at most n independent commuting Pauli operators on n qubits.

Claim 1. *N commutes with every M_i iff $\pm N$ is a product of some M_i's.*

Proof of claim. Certainly, if $\pm N$ is a product of M_i's, then it commutes with all of them, since they all commute with each other.

The forward direction can again be seen as a consequence of linear algebra. Let $(\mathbf{x}|\mathbf{z})$ be the binary vector corresponding to M and let $(\mathbf{x}'|\mathbf{z}')$ be the binary vector corresponding to M'. Then we can determine by direct calculation that M and M' commute iff

$$\mathbf{x}\cdot\mathbf{z}' \oplus \mathbf{z}\cdot\mathbf{x}' = 0. \tag{24}$$

The commutation of Paulis corresponds to a *symplectic product* in the binary vector space. In particular, if M is the matrix whose columns are the binary vectors corresponding to M_i, then N with binary vector $(\mathbf{x}|\mathbf{z})$ commutes with all of the M_i's iff

$$(\mathbf{z}|\mathbf{x})M = \mathbf{0}. \tag{25}$$

(Note here that the z and x terms in the vector are switched due to the symplectic product.) This means that the vector $(\mathbf{x}|\mathbf{z})$ is again a solution to a set of linear equations. But since the M_i's are all independent, the matrix M has maximum rank n. That means that the dimension of the solution space is n (as a binary vector space). But the columns of M, the vectors corresponding to M_i, are solutions already, since the M_i's commute with each other, and there are n of them. They are linearly independent, so they span the solution space and any vector that solves (25) is a sum of the vectors corresponding to M_i. This, in turn, means that $\pm N$ is a product of the M_i's. □

So the only remaining case is when Z_i fails to commute with one or more of the M_j's. Elements of the Pauli group either commute or *anticommute*, $PQ = -QP$. Therefore, there must be some j such that $Z_i M_j = -M_j Z_i$. In this case, the measurement outcome must be a random bit.

To see this, note that the projector onto the ± 1 eigenspace of Z_i is $(I \pm Z_i)/2$. This means that the probability of getting the outcome 0 to when measuring the ith bit of the state $|\psi\rangle$ is

$$\frac{1}{2}\langle\psi|(I + Z_i)|\psi\rangle. \tag{26}$$

But if $|\psi\rangle$ is a $+1$ eigenstate of M_j and M_j anticommutes with Z_i, we have

$$\frac{1}{2}\langle\psi|(I + Z_i)|\psi\rangle = \frac{1}{2}\langle\psi|(I + Z_i)M_j|\psi\rangle = \frac{1}{2}\langle\psi|M_j(I - Z_i)|\psi\rangle = \frac{1}{2}\langle\psi|(I - Z_i)|\psi\rangle, \tag{27}$$

which is the probability of getting outcome 1. Thus, outcome 0 and outcome 1 both have probability 1/2.

If we are only measuring a single qubit, we can stop here. But that won't let us calculate conditional probabilities. To go further, we want to figure out the residual state of the remaining qubits after we measure one of them. We can do so by noting that if we measure qubit i and get outcome 0, the overall state is now $\frac{1}{\sqrt{2}}(I + Z_i)|\psi\rangle$, the projector onto the $+1$ eigenspace of Z_i, renormalized to take into account that the probability of this outcome is 1/2. We won't bother to track the normalization from now on, since it is automatic. We can also do the same thing for outcome 1 using the projector $I - Z_i$ (i.e., using $-Z_i$ instead of Z_i).

We can update the stabilizer generators to take the measurement into account. Note that if M_k commutes with Z_i, then the state is still a $+1$ eigenstate of M_k:

$$M_k(I + Z_i)|\psi\rangle = (I + Z_i)M_k|\psi\rangle = (I + Z_i)|\psi\rangle. \tag{28}$$

The state is not still an eigenstate of M_j, which anticommuted with Z_i, and any other M_k that anticommute with Z_i have a similar problem. However, note that before the measurement, if the state is a $+1$ eigenstate of M_j and M_k, then it is also a $+1$ eigenstate of $M_j M_k$, and if M_j and M_k both anticommute with Z_i, then $M_j M_k$ commutes with Z_i:

$$Z_i(M_j M_k) = -M_j Z_i M_k = +(M_j M_k)Z_i. \tag{29}$$

Therefore, after the measurement, the state is a $+1$ eigenstate of $M_j M_k$.

We therefore have the following algorithm to compute a new set $\{M_1, \ldots, M_n\}$ for which the post-measurement state is a $+1$ eigenstate:

1. Find j such that M_j anticommutes with Z_i
2. Run through all $k = 1, \ldots, n$, $k \neq j$. If M_k commutes with Z_i, leave it. If M_k anticommutes with Z_i, replace it by $M_j M_k$.
3. Replace M_j by Z_i if the measurement outcome was 0 and by $-Z_i$ if the measurement outcome was 1.

It is not hard to see that the resulting new set of M_i's all commute with each other and are independent.

We therefore have the following algorithm to determine conditional probabilities of measurements. Suppose we want to find the probability of measuring 0 on qubit d conditioned on having the outcomes $c_1, \ldots, c_{d-1}$ on qubits 1 through $d-1$. (This is WLOG since we can relabel the qubit numbers as needed.)

1. For qubit i running from 1 to $d-1$:
   (a) Determine if Z_i commutes with all M_j.
   (b) If Z_i and all M_j commute, solve (22) to find the expansion of $\pm Z_i$ as a product of the M_j's and then determine if the outcome of measuring Z_i should be 0 or 1. If the result matches c_i, then continue; otherwise, this condition is not possible, so halt and return that result.

(c) If Z_i anticommutes with some M_j, update that stabilizer as above assuming that the (random) measurement outcome is c_i.

2. Determine if Z_d commutes with all M_j

3. If Z_d and all M_j commute, solve (22) to find the expansion of $\pm Z_d$ as a product of the M_j's and then determine if the outcome of measuring Z_d is 0 or 1. If the outcome is 0, return probability 1; if the outcome is 1, return probability 0.

4. If Z_i anticommutes with some M_j, return probability 1/2.

Solving the systems of linear equations by Gaussian elimination takes time $O(n^3)$, so that is the complexity of this algorithm. By tracking some additional information, we can speed this up to an algorithm taking time $O(n^2)$. □

Note that the conditional probability of getting 0 on a qubit is always 0, 1, or 1/2 (or the conditional cannot occur). This is a consequence of the special structure of the Clifford group.

This theorem about classical simulation of the Clifford group can be strengthened in a few different ways. First, note that the proof already shows that we don't need to have measurement only at the end of the computation. We can allow measurement in the middle of the computation and condition future unitaries on the result of some classical computation.

If we don't allow conditional unitaries and wish to solve a decision problem using only Clifford group gates, the above theorem tells us that the problem is in P. We might wonder if it can do all of P, or if it is located in some smaller complexity class. Indeed, Clifford group gates by themselves are *not* as strong as P. Instead, this problem is complete for a class called $\oplus$L, which is the set of problems reducible via a logspace reduction (not a polynomial-time reduction) to a sequence of XORs (i.e., CNOTs). Indeed, most of the algorithm to simulate a Clifford group circuit just involves XORs. The only place where we need more is in calculating the overall phases, which needs a very limited amount of mod 4 arithmetic. However, this part can still be done as part of the logspace reduction.

Another practical impact of this theorem is that many algorithms for more-efficient-than-the-obvious (but still exponential) classical simulation of general quantum circuits build on it. For instance, there are algorithms that are exponential in the number of non-Clifford gates in the circuit, which can be very effective in simulating circuits which are mostly Clifford gates.

4 Matchgates

We now turn to a set of gates that has a somewhat different set of rules. We will still assume that we are starting in the state $|00\ldots0\rangle$ and measuring at the end in the computational basis, but now the qubits have an order and we can only perform gates on adjacent qubits. For the Clifford group, putting the qubits in a line like this wouldn't have mattered, because the SWAP gate is in the Clifford group, so we could reorder however we liked. The current set of gates won't quite have the SWAP gate, however.

The gates we are interested are the set of two-qubit gates of the form

$$G(A,B) = \begin{pmatrix} a & 0 & 0 & b \\ 0 & w & x & 0 \\ 0 & y & z & 0 \\ c & 0 & 0 & d \end{pmatrix}, \tag{30}$$

where

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, B = \begin{pmatrix} w & x \\ y & z \end{pmatrix} \tag{31}$$

are matrices in $SU(2)$. Recall that this means they are unitary and have determinant 1. Note that this is a case where it matters that they are in $SU(2)$ rather than $U(2)$ because it is important for the result that they have the same relative phase. So, for instance, the SWAP gate wouldn't work because it has the matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{32}$$

which would give us a B with determinant -1. Instead we can have an "iSWAP" gate

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{33}$$

which swaps the two qubits but creates a phase i if they are not the same.

These gates, the *matchgates*, rotate adjacent pairs of qubits in the $|00\rangle$ and $|11\rangle$ subspace and the $|01\rangle$ and $|10\rangle$ subspace. Thus, they preserve the total parity of the state. There is also some phase constraint to the rotation, as we saw with the SWAP/iSWAP example. They have a physical interpretation as beam splitters and pair creation/destruction for non-interacting Majorana fermions. (Note: the original paper about classical simulation of matchgates by Valiant had some additional non-unitary gates, but these are not too relevant for most quantum computing applications.)

Note that matchgates include the identity and all single-qubit phase gates (with suitable global phase), but not things like the single-qubit Hadamard or bit flip. This is an infinite set, however, unlike the Clifford group. To restrict it to a reasonable complexity class, we should require that the matrices A and B are efficiently computable.

Leaning into the interpretation of Majorana fermions (and if you don't know what they are, that's OK, but it provides motivation), we can define the Majorana operators from the Paulis via the *Jordan-Wigner* transformation:

$$c_1 = X_0,\ c_2 = Y_0,\ c_{2i+1} = \bigotimes_{j=0}^{i-1} Z_j X_i,\ c_{2i+2} = \bigotimes_{j=0}^{i-1} Z_j Y_i \tag{34}$$

for $i = 1, \ldots, n-1$. Thus, there are $2n$ operators c_μ for n qubits.

Theorem 4. *For any matchgate $G(A,B)$, for all μ*

$$G(A,B)^\dagger\, c_\mu\, G(A,B) = \sum_\nu R_{\mu\nu} c_\nu. \tag{35}$$

Proof. One way to show this is to first note that any matchgate $G(A,B) = e^{-iH}$, where H is a sum of terms quadratic in the c's. This in turn will imply the result.

However, it is fairly straightforward and more elementary to just calculate this. First note that since $G(A,B)$ is a 2-qubit gate, we only need focus attention on two adjacent qubits in c_μ, and there are only 6 possible 2-qubit Paulis that appear on adjacent qubits in any c_μ: $I \otimes I$, $Z \otimes Z$, $X \otimes I$, $Y \otimes I$, $Z \otimes X$, and $Z \otimes Y$.

Now, $I \otimes I$ commutes with anything, so $G(A,B)^\dagger (I \otimes I) G(A,B) = I \otimes I$. $Z \otimes Z$ acts like I on the $|00\rangle$, $|11\rangle$ subspace and like $-I$ on the $|01\rangle$, $|10\rangle$ subspace, so again it commutes with $G(A,B)$ (which doesn't mix those subspaces): $G(A,B)^\dagger (Z \otimes Z) G(A,B) = Z \otimes Z$. Thus, if $G(A,B)$ acts on qubits i and $i+1$, all of the c_μ with $\mu < 2i+1$ or $\mu > 2i+4$ are left unchanged by conjugation.

That leaves $X \otimes I$ (which appears in c_{2i+1} on qubits i and $i+1$), $Y \otimes I$ (which appears in c_{2i+2}), $Z \otimes X$ (c_{2i+3}), and $Z \otimes Y$ (c_{2i+4}). We will show that conjugation gives us a linear combination of these same four operators.

First, observe:

$$X \otimes I = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{36}$$

$$Y \otimes I = \begin{pmatrix} 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \\ 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \end{pmatrix} \tag{37}$$

$$Z \otimes X = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix} \tag{38}$$

$$Z \otimes Y = \begin{pmatrix} 0 & 0 & -i & 0 \\ 0 & 0 & 0 & i \\ i & 0 & 0 & 0 \\ 0 & -i & 0 & 0 \end{pmatrix}. \tag{39}$$

Then we can calculate $G(A,B)^\dagger (X \otimes I) G(A,B)$:

$$G(A,B)^\dagger (X \otimes I) G(A,B) = \begin{pmatrix} a^* & 0 & 0 & c^* \\ 0 & w^* & y^* & 0 \\ 0 & x^* & z^* & 0 \\ b^* & 0 & 0 & d^* \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} a & 0 & 0 & b \\ 0 & w & x & 0 \\ 0 & y & z & 0 \\ c & 0 & 0 & d \end{pmatrix} \tag{40}$$

$$= \begin{pmatrix} 0 & a^*w + c^*y & a^*x + c^*z & 0 \\ aw^* + cy^* & 0 & 0 & bw^* + dy^* \\ ax^* + cz^* & 0 & 0 & bx^* + dz^* \\ 0 & b^*w + d^*y & b^*x + d^*z & 0 \end{pmatrix}. \tag{41}$$

Now, this has 0s in the right place and is Hermitian, but in order for it to be correct, we need some relationship between the top left and bottom right blocks and the top right and bottom left blocks. This follows because A and B are in $SU(2)$.

In particular, for a unitary matrix, the rows and columns must be orthonormal vectors, and for a 2-D Hilbert space, that is very limiting, and even more so for an element of $SU(2)$. In particular, A must have the form

$$A = \begin{pmatrix} a & b \\ -b^* & a^* \end{pmatrix} \tag{42}$$

and

$$B = \begin{pmatrix} w & x \\ -x^* & w^* \end{pmatrix}. \tag{43}$$

This tells us

$$a^*w + c^*y = a^*w + bx^* = bx^* + dz^* \equiv r + is \tag{44}$$

$$a^*x + c^*z = a^*x - bw^* = -(bw^* + dy^*) \equiv t + iu \tag{45}$$

with real r, s, t, u. Thus,

$$G(A,B)^\dagger (X \otimes I) G(A,B) = \begin{pmatrix} 0 & r+is & t+iu & 0 \\ r-is & 0 & 0 & -t-iu \\ t-iu & 0 & 0 & r+is \\ 0 & -t+iu & r-is & 0 \end{pmatrix} \tag{46}$$

$$= rX \otimes I - sY \otimes I + tZ \otimes X - uZ \otimes Y. \tag{47}$$

Similarly for the other three cases. □

As a consequence of this theorem, we can immediately track the behavior of any matchgate circuit by following the evolution of the c_μ's. Note that since there are only $2n$ c_μ's, $R_{\mu\nu}$ is a $2n \times 2n$ matrix, so we can store it classically and multiply the R matrices corresponding to each matchgate in the circuit. This gives us an overall R matrix that determines the behavior of the circuit.

But once again, this is not enough. We need to determine measurement outcomes. To do so, note that $Z_i = -ic_{2i+1}c_{2i+2}$. Then if $|\psi\rangle = U|00\ldots0\rangle$ for some U corresponding to a matchgate circuit with overall matrix R,

$$\langle\psi|Z_i|\psi\rangle = -i\langle 00\ldots0|U^\dagger c_{2i+1}c_{2i+2}U|00\ldots0\rangle \tag{48}$$

$$= -i\langle 00\ldots0|(U^\dagger c_{2i+1}U)(U^\dagger c_{2i+2}U)|00\ldots0\rangle \tag{49}$$

$$= -i\sum_{\nu,\xi} R_{2i+1,\nu}R_{2i+2,\xi}\langle 00\ldots0|c_\nu c_\xi|00\ldots0\rangle. \tag{50}$$

Now, $c_\nu c_\xi$ can be translated back as a product of Paulis $\bigotimes P_i$, so

$$\langle 00\ldots0|c_\nu c_\xi|00\ldots0\rangle = \prod_i \langle 0|P_i|0\rangle. \tag{51}$$

Since $\langle\psi|Z_i|\psi\rangle = \text{Prob}(0) - \text{Prob}(1)$ when qubit i is measured in the standard basis, we can therefore efficiently calculate the measurement distribution on qubit i. Given a circuit of T matchgates, the procedure involves multiplying T $2n \times 2n$ matrices to find R and then summing up potentially $O(n^2)$ terms, each of which can be computed in time $O(n)$.

The marginal probability of getting outcome v on some subset of qubits is

$$\text{Prob}(v) = \text{Tr}(|v\rangle\langle v|)|\psi\rangle\langle\psi| \tag{52}$$

We can write the projector on $|v\rangle$ in terms of the c_μ as well, and then calculate this using the same procedure as above. However, the time to do this calculation is exponential in the number of c's, which is proportional to the number of qubits whose joint measurement we are simulating, so this approach is inherently limited.

Nevertheless, it is possible to do so using a more sophisticated argument (which I will not reproduce). The answer turns out to be the Pfaffian of some $n \times n$ matrix, and the Pfaffian is the square root of the determinant, and so can be computed in polynomial time (for instance by diagonalizing the matrix).

Joint measurement of subsets then gives us conditional probability distributions:

$$\text{Prob}(0|v) = \text{Prob}(0v)/\text{Prob}(v). \tag{53}$$

Therefore, we have the following result:

Theorem 5. *There is an efficient classical algorithm to perform an exact strong simulation of any circuit of matchgates specified with constant precision.*

There are also various strengthenings of this result. For instance, the exact power of matchgates (at least for decision problems) is known: The class of decision problems solvable with bounded probability with matchgates is equivalent (under classical logspace reductions) to logspace quantum computation.

5 Computational Universality

5.1 Real gates and computational universality

Now let us consider the following set of gates: $\mathcal{G} = \{\text{all 3-qubit unitary gates with real matrix elements}\}$. Clearly this set of gates, even though it includes many-qubit gates, is not universal or even approximately universal: We can't get close to implementing any gate with complex matrix elements no matter how many

real gates we multiply together. Nevertheless, this is clearly a very large subgroup of the unitary group. (It is actually $SO(2^n)$, the real orthogonal matrices.) That seems unlikely to be classically simulatable.

In fact, it is just as computationally powerful as a universal set. For instance, if we define BQP using efficiently computable gates in $\mathcal{G}$, we get the same complexity class BQP as if we define it using an approximately universal set of gates.

The most straightforward way to see this is to realize that while we can't *approximate* an arbitrary unitary operation, we can *simulate* an arbitrary unitary:

Theorem 6. *For any initial quantum state followed by any unitary operation and measurement of every qubit, there exists an exact weak simulation of the circuit using an initial quantum state with real amplitudes and a circuit consisting of gates from $\mathcal{G}$, followed by measurement of every qubit and a constant-depth classical computation.*

Proof. An n-qubit system with complex amplitudes will be simulated in an $(n+1)$-qubit system with *real* amplitudes. When the extra qubit is $|0\rangle$, that gives us the real part of the original state and when the extra qubit is $|1\rangle$, that gives us the imaginary part. In particular, the n-qubit complex state $|\psi\rangle = \sum_x \alpha_x |x\rangle$ corresponds to the $(n+1)$-qubit real state

$$|\tilde{\psi}\rangle = \sum_x (\mathrm{Re}(\alpha_x)|0\rangle + \mathrm{Im}(\alpha_x)|1\rangle) \otimes |x\rangle. \tag{54}$$

Note that the notion of a state or unitary with real coefficients is inherently a basis-dependent concept. And also note that the map $|\psi\rangle \mapsto |\tilde{\psi}\rangle$ is *not* unitary.

We can then simulate a unitary U as follows:

$$\tilde{U}|0\rangle \otimes |x\rangle = |0\rangle \otimes \mathrm{Re}(U)|x\rangle + |1\rangle \otimes \mathrm{Im}(U)|x\rangle \tag{55}$$

$$\tilde{U}|1\rangle \otimes |x\rangle = -|0\rangle \otimes \mathrm{Im}(U)|x\rangle + |1\rangle \otimes \mathrm{Re}(U)|x\rangle. \tag{56}$$

Lemma 2. *For all $|\psi\rangle$,*

$$\tilde{U}|\tilde{\psi}\rangle = \widetilde{U|\psi\rangle}. \tag{57}$$

Proof of lemma.

$$\tilde{U}|\tilde{\psi}\rangle = \sum_x \mathrm{Re}(\alpha_x)(|0\rangle \otimes \mathrm{Re}(U)|x\rangle + |1\rangle \otimes \mathrm{Im}(U)|x\rangle) + \mathrm{Im}(\alpha_x)(-|0\rangle \otimes \mathrm{Im}(U)|x\rangle + |1\rangle \otimes \mathrm{Re}(U)|x\rangle) \tag{58}$$

$$= \sum_x (\mathrm{Re}(\alpha_x)|0\rangle + \mathrm{Im}(\alpha_x)|1\rangle) \otimes \mathrm{Re}(U)|x\rangle + (\mathrm{Re}(\alpha_x)|1\rangle - \mathrm{Im}(\alpha_x)|0\rangle) \otimes \mathrm{Im}(U)|x\rangle. \tag{59}$$

Now,

$$U|\psi\rangle = (\mathrm{Re}(U) + i\mathrm{Im}(U))(\sum_x (\mathrm{Re}(\alpha_x) + i\mathrm{Im}(\alpha_x))|x\rangle) \tag{60}$$

$$= \sum_x (\mathrm{Re}(\alpha_x)\mathrm{Re}(U)|x\rangle - \mathrm{Im}(\alpha_x)\mathrm{Im}(U)|x\rangle) + i(\mathrm{Im}(\alpha_x)\mathrm{Re}(U)|x\rangle + \mathrm{Re}(\alpha_x)\mathrm{Im}(U)|x\rangle) \tag{61}$$

$$= \sum_x (\mathrm{Re}(\alpha_x) + i\mathrm{Im}(\alpha_x))\mathrm{Re}(U)|x\rangle + (-\mathrm{Im}(\alpha_x) + i\mathrm{Re}(\alpha_x))\mathrm{Im}(U)|x\rangle. \tag{62}$$

Thus,

$$\widetilde{U|\psi\rangle} = \sum_x (\mathrm{Re}(\alpha_x)|0\rangle + \mathrm{Im}(\alpha_x)|1\rangle) \otimes \mathrm{Re}(U)|x\rangle + (-\mathrm{Im}(\alpha_x)|0\rangle + \mathrm{Re}(\alpha_x)|1\rangle) \otimes \mathrm{Im}(U)|x\rangle, \tag{63}$$

which is equal to $\tilde{U}|\tilde{\psi}\rangle$, as claimed.

□

This lemma means that if we start with a state encoded in this way, we can maintain the encoding and evolve the state according to the appropriate unitary. Note that if U is a t-qubit gate, then $\tilde{U}$ will be a $(t+1)$-qubit gate since it involves the extra qubit. In particular, since $\mathcal{G}$ includes all 3-qubit real gates, it can simulate all 2-qubit complex gates, and since those are universal, circuits generated by $\mathcal{G}$ can simulate all unitaries in this encoding.

So far, we have seen how to simulate an arbitrary initial state and maintain the encoding through arbitrary unitary circuits. But what about the final measurement? If we measure every qubit of $|\tilde{\psi}\rangle$, we get the outcome $0x$ with probability $|\mathrm{Re}(\alpha_x)|^2$ and the outcome $1x$ with probability $|\mathrm{Im}(\alpha_x)|^2$. If we discard the extra qubit, we get that the probability of outcome x is

$$|\mathrm{Re}(\alpha_x)|^2 + |\mathrm{Im}(\alpha_x)|^2 = |\alpha_x|^2, \tag{64}$$

which is exactly the probability of outcome x in the original circuit. □

This is an interesting example, which shows that we can have the full power of a quantum computer without having a universal set of gates. What we have is a weaker notion of universality, *computational universality*. I don't think there is a single accepted definition of computational universality, but here is a sensible one:

Definition 3. *A set of quantum resources (e.g., a set of quantum gates) is* computationally universal *if there exists a polynomial classical algorithm which, given any quantum circuit Q with n qubits and T gates starting with standard initial states and ending with standard basis measurements, produces an algorithm using classical computation and subroutine calls using the quantum resources which produces an approximate weak simulation of Q with additive error ϵ which uses $O(\mathrm{poly}(n, T, \log 1/\epsilon))$ classical and quantum gates or other resources.*

There are alternatives that are both weaker and stronger than this definition. For instance, this doesn't say anything about the ability to simulate situations where you want to keep a quantum state around at the end of the computation. Such a definition would need to make explicit mention of a map between the state in the original circuit and in the simulation; but it would exclude other simulations that perform some sort of transformation on the whole circuit and don't explicitly simulate the instantaneous state. We could also insist on a stricter degree of approximation, but we don't want to be too strict, since we would like regular approximate universality to be a special case of computational universality. On the other hand, rather than asking for a weak simulation, we could instead say that the computationally universal set of resources can decide any language in BQP while using only polynomial resources. I am not aware of any interesting examples which achieve the BQP definition and not the weak simulation one (although I believe you can construct somewhat artificial examples, for instance by saying that you can only measure one qubit of the final state of a quantum circuit).

5.2 Tof and H

Now let us look at the set of gates $\mathcal{G} = \{Tof, H\}$. These are real gates, so we can't hope for more than computational universality, but it turns out we do indeed have computational universality.

To show this, we use the same simulation given by the general real simulation. We can start from a somewhat standard approximately universal set $\{H, C - R_{\pi/4}\}$ and see what real gates we need to simulate that. The Hadamard is a real gate already, so $\mathrm{Im}(H) = 0$ and

$$\tilde{H}|0\rangle \otimes |x\rangle = |0\rangle \otimes H|x\rangle \tag{65}$$

$$\tilde{H}|1\rangle \otimes |x\rangle = |1\rangle \otimes H|x\rangle, \tag{66}$$

so $\tilde{H}$ is just H acting on the same qubit and doing nothing to the extra one.

The $C - R_{\pi/4}$ is the controlled-$\pi/4$ rotation, $C - R_{\pi/4} = \mathrm{diag}(1, 1, 1, i)$. Note that the phase in $R_{\pi/4}$ is an arbitrary choice, but in $C - R_{\pi/4}$, because of the control, the choice of phase gives different 2-qubit gates. We will choose this particular phase, which is standard. Then

$$\mathrm{Re}(C - R_{\pi/4}) = \mathrm{diag}(1, 1, 1, 0) \tag{67}$$

$$\mathrm{Im}(C - R_{\pi/4}) = \mathrm{diag}(0, 0, 0, 1), \tag{68}$$

which means

$$\widetilde{C - R_{\pi/4}}|000\rangle = |000\rangle \tag{69}$$

$$\widetilde{C - R_{\pi/4}}|001\rangle = |001\rangle \tag{70}$$

$$\widetilde{C - R_{\pi/4}}|010\rangle = |010\rangle \tag{71}$$

$$\widetilde{C - R_{\pi/4}}|011\rangle = |111\rangle \tag{72}$$

$$\widetilde{C - R_{\pi/4}}|100\rangle = |100\rangle \tag{73}$$

$$\widetilde{C - R_{\pi/4}}|101\rangle = |101\rangle \tag{74}$$

$$\widetilde{C - R_{\pi/4}}|110\rangle = |110\rangle \tag{75}$$

$$\widetilde{C - R_{\pi/4}}|111\rangle = -|011\rangle \tag{76}$$

We can recognize this as $CC - Z$, a twice-controlled Z gate on the three qubits (which gives a phase -1 if all three qubits are 1), followed by a Toffoli gate with the last two qubits as controls and the first one as target. The $CC - Z$ gate is equal to $(I \otimes I \otimes H)Tof(I \otimes I \otimes H)$. Therefore, to perform the real simulation of H and $C - R_{\pi/4}$, it suffices to have H and Tof, proving these gates are computationally universal.

Because Tof is universal for classical reversible computation, this result means we can think of the extra computational power of quantum computing as coming just from the Hadamard gate. But don't read too much into that: the field has lots of results like this, with quantum coming from something classical plus just one more thing.

5.3 Encoded Universality

Another important type of computational universality is *encoded universality*. Like the example of real gates, encoded universality gives a simulation of a general quantum circuit, but unlike real gates, the encoding is unitary. A system with encoded universality is genuinely universal (exactly or approximately), but only on a subspace of the whole Hilbert space.

One particularly striking example of encoded universality is the *exchange interaction*. Specifically, we will look at the gate set $\mathcal{G} = \{e^{-itH_E}\}$ for arbitrary t (or at least computable t), and

$$H_E = \frac{1}{2}(I \otimes I + X \otimes X + Y \otimes Y + Z \otimes Z). \tag{77}$$

This is called the exchange interaction because it implements the SWAP gate between the two qubits. In some sense, therefore, it seems trivial since all it does is switch the order of qubits, but with the ability to run this Hamiltonian for varying amounts of time, it is possible to do non-trivial interactions which lead to computational universality. However, note that on the initial state $|00\rangle$ it does nothing, so we need a starting state other than that.

The exchange interaction permutes qubits, so in particular, it commutes with any unitary of the form $U \otimes U$ on the two qubits. It turns out that with enough qubits, there are large subspaces that are invariant under all operations $U^{\otimes n}$ on n qubits. When $n = 4$, we can use the subspace spanned by

$$|\bar{0}\rangle = \frac{1}{2}(|01\rangle - |10\rangle)(|01\rangle - |10\rangle) \tag{78}$$

$$|\bar{1}\rangle = \frac{1}{12}(2|0011\rangle + 2|1100\rangle - |0101\rangle - |1010\rangle - |1001\rangle - |0110\rangle). \tag{79}$$

(You can check by hand that these states have the claimed properties.) Because the Hamiltonian commutes with $U \otimes U \otimes U \otimes U$ for all U, the state remains invariant under such operations when we act on it with a gate from $\mathcal{G}$, so those gates keep us in this two-dimensional subspace. An analysis of the Lie algebra shows that by applying the Hamiltonian on different pairs of qubits in succession, we can generate arbitrary elements of $SU(2)$ on the subspace. (You can reduce this to a subspace of a 3-qubit system to encode on qubit, but the encoding is a bit more complicated because each logical basis state corresponds to a 2-dimensional subspace rather than a single state.)

When we have 8 qubits, which is enough to encode two logical qubits, the subspace of states invariant under $U^{\otimes n}$ is larger: It is 14-dimensional, so it includes much more than the tensor product of the two encoded qubit Hilbert spaces. The exchange Hamiltonian acting on different pairs of qubits keeps us in this 14-dimensional Hilbert space but not in the two-qubit logical Hilbert space. It generates the full $SU(14)$ unitary group on this larger Hilbert space. When we have all gates of the form e^{-itH_E}, we can exactly get all such gates and we can restrict ourselves to only using those gates within the $SU(4)$ subgroup corresponding to two-qubit logical operations. That will give us a universal set of logical gates.

In the case where we instead have only a limited set of times t that we can use (and then rely on the Solovay-Kitaev theorem to generate a dense set of gates in these subgroups), we might not get any exact two-qubit gates. However, we can approximate them to high accuracy. This acts as some additional "leakage" errors taking us out of the computational Hilbert space, but by making them small enough, we can make them negligible over the course of the computation. In this case, we can approximate an arbitrary unitary in the computational subspace.

In either case, the ability to exactly or approximately perform arbitrary elements of $SU(2^n)$ using $O(n)$ qubits lets us accurately simulate any quantum algorithm. Thus, this set of gates is also computationally universal.

6 References

For a proof of the Solovay-Kitaev theorem, see Ch. 8 of Kitaev, Shen, and Vyalyi, *Classical and Quantum Computation* or Appendix 3 of Nielsen and Chuang, *Quantum Computation and Quantum Information*. The inverse-free Solovay-Kitaev theorem is Bouland and Giurgica-Tiron, "Efficient Universal Quantum Compilation: An Inverse-free Solovay-Kitaev Algorithm," arXiv:2112.02040 [quant-ph].

For the Clifford group, see D. Gottesman, "The Heisenberg Representation of Quantum Computers," quant-ph/9807006 and Aaronson and Gottesman, "Improved Simulation of Stabilizer Circuits," quant-ph/0406196.

The analysis of matchgates is based on Terhal and DiVincenzo, "Classical simulation of noninteracting-fermion quantum circuits," quant-ph/0108010 and Jozsa and Miyake, "Matchgates and classical simulation of quantum circuits," arXiv:0804.4050 [quant-ph]. The result pinpointing the exact strength of matchgates is Jozsa, Kraus, Miyake, and Watrous, "Matchgate and space-bounded quantum computations are equivalent," arXiv:0908.1467 [quant-ph].

The computational universality of real gates is from Bernstein and Vazirani, "Quantum Complexity Theory," SIAM Journal on Computing, vol. 26, 1411-1473 (1997), which also contains a lot about quantum Turing machines and foundational results on quantum complexity. The proof that Toffoli plus Hadamard is computationally universal is from Shi, "Both Toffoli and Controlled-NOT need little help to do universal quantum computation," quant-ph/0205115. My presentation of both results is from Aharonov, "A Simple Proof that Toffoli and Hadamard are Quantum Universal," quant-ph/0301040. For a discussion of encoded universality, see Kempe, Bacon, Lidar, and Whaley, "Theory of Decoherence-Free Fault-Tolerant Universal Quantum Computation," quant-ph/0004064.