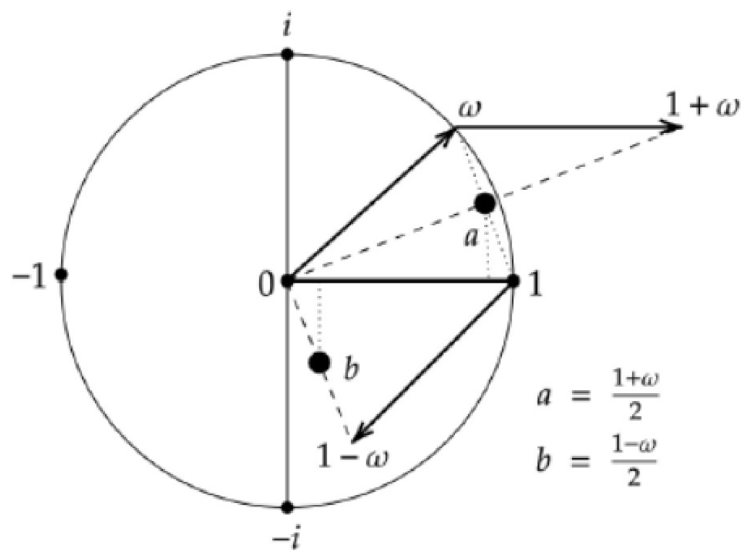
CSE439 Fall 2026 Week 5: Building and Visualizing Quantum Circuits

The $\mathbf{Z}$ and $\mathbf{CZ}$ gates are the heads of an important family of basic gates having to do with rotations of **phase**, which is a curious but definitely physical property. When a complex number $x + iy$ is rewritten in polar form as $re^{i\theta}$, the angle θ is the phase. The magnitude is r, so when $r = 1$ we have a unit complex number. Note that i itself is the same as $e^{i\pi/2}$ since $\frac{\pi}{2}$ means 90° phase. Then $i^2 = e^{i\pi} = -1$ and if we put $\omega = e^{i\pi/4}$ then $\omega^2 = i$. In Cartesian coordinates, $\omega = \frac{1+i}{\sqrt{2}}$. Here is some more geometry:



The vector $\mathbf{u} = [a, b]^T$ is a funky unit vector. To see that it is a unit vector, note that

$$||\mathbf{u}||^2 = \langle \mathbf{u}, \mathbf{u} \rangle = \mathbf{u}^* \cdot \mathbf{u} = a^*a + b^*b = \left(\frac{1+\overline{\omega}}{2}\right)\left(\frac{1+\omega}{2}\right) + \left(\frac{1-\overline{\omega}}{2}\right)\left(\frac{1-\omega}{2}\right).$$

In polar form, the complex conjugate of $e^{i\theta}$ is always $e^{-i\theta} = e^{i(2\pi-\theta)}$, so $\overline{\omega} = e^{i7\pi/4} = \omega^7$. In Cartesian coordinates,

$$\frac{1+\omega}{2} = \frac{1}{2}\left(1 + \frac{1+i}{\sqrt{2}}\right) = \frac{\sqrt{2}+1+i}{2\sqrt{2}} \quad\text{and}\quad \frac{1-\omega}{2} = \frac{1}{2}\left(1 - \frac{1+i}{\sqrt{2}}\right) = \frac{\sqrt{2}-1-i}{2\sqrt{2}}$$

So

$$\frac{1+\overline{\omega}}{2} = \frac{1}{2}\left(1 + \frac{1-i}{\sqrt{2}}\right) = \frac{\sqrt{2}+1-i}{2\sqrt{2}} \quad\text{and}\quad \frac{1-\overline{\omega}}{2} = \frac{1}{2}\left(1 - \frac{1-i}{\sqrt{2}}\right) = \frac{\sqrt{2}-1+i}{2\sqrt{2}}.$$

Then

$$\left(\frac{1+\overline{\omega}}{2}\right)\left(\frac{1+\omega}{2}\right) = \frac{1}{8}\left(\sqrt{2}+1+i\right)\left(\sqrt{2}+1-i\right) = \frac{1}{8}\left[\left(\sqrt{2}+1\right)^2 + 1\right] = \frac{1}{8}\left(2+1+2\sqrt{2}+1\right) = \frac{2+\sqrt{2}}{4},$$

$$\left(\frac{1-\overline{\omega}}{2}\right)\left(\frac{1-\omega}{2}\right) = \frac{1}{8}\left(\sqrt{2}-1-i\right)\left(\sqrt{2}-1+i\right) = \frac{1}{8}\left[\left(\sqrt{2}-1\right)^2+1\right] = \frac{1}{8}\left(2+1-2\sqrt{2}+1\right) = \frac{2-\sqrt{2}}{4}.$$

These squared values add to 1 as promised, so $\mathbf{u} = [a,b]^T$ is a unit vector. How do we get it? Here is the start of an infinite family of gates:

$$\mathbf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \ \mathbf{S} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \ \mathbf{T} = \begin{bmatrix} 1 & 0 \\ 0 & \omega \end{bmatrix}, \ \mathbf{T}_{\pi/8} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/8} \end{bmatrix}.$$

The controlled versions to go with $\mathbf{CZ}$ are $\mathbf{CS}$, $\mathbf{CT}$, etc. They, too, are symmetric---indeed, all of these gates are controlled phase shifts conditioned on the basis-state 1 of all of the (one or two) qubits involved.

Now here is an important distinction. The **group commutator** of any two operators A, B is ABA^*B^*. We can repeat the pattern: $ABA^*B^*ABA^*B^*ABA^*B^*$... Let's take $A = \mathbf{H}$ and start with $B = \mathbf{Z}$. Then A and B are self-adjoint, so

$$ABA^*B^* = \mathbf{HZHZ} = \mathbf{XZ} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = -i\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = -i\mathbf{Y}.$$

Just as with quantum states, two unitary operators are **equivalent** if one is a unit scalar multiple of each other. The Pauli matrices $\mathbf{I}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}$ are closed under multiplication up to this equivalence and so form a mathematical **group**. (The constant factors are always $1, i, -1, -i$ so you can also say they form a larger group without this equivalence.) Since $\mathbf{Y}^2 = \mathbf{I}$, repeating the commutator never gets us out of the group.

Next try $B = \mathbf{S}$. The matrix $\mathbf{V} = \mathbf{HSH}$ is $\frac{1}{2}\begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix}$. Note this equals $\frac{1}{\sqrt{2}}\begin{bmatrix} e^{i\pi/4} & e^{-i\pi/4} \\ e^{-i\pi/4} & e^{i\pi/4} \end{bmatrix}$, and if you multiply it by the unit scalar $e^{i\pi/4}$ you get the nicer-looking matrix $\mathbf{V}' = \frac{1}{\sqrt{2}}\begin{bmatrix} i & 1 \\ 1 & i \end{bmatrix}$. Note: $\mathbf{V}^2 = \mathbf{HSHHSH} = \mathbf{HSSH} = \mathbf{HZH} = \mathbf{X}$. So $\mathbf{V}$ is called the "square root of NOT" and is also written as SRN or as SRNOT or as $\mathbf{X}^{1/2}$ in various sources. Now (using $\mathbf{V}'$) we have:

$$\mathbf{HSHS}^* \equiv \frac{1}{\sqrt{2}}\begin{bmatrix} i & 1 \\ 1 & i \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} i & -i \\ 1 & 1 \end{bmatrix}.$$

Call that matrix W for now. Then $W^2 = \frac{1}{2}\begin{bmatrix} -1-i & 1-i \\ 1+i & 1-i \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} -e^{i\pi/4} & e^{-i\pi/4} \\ e^{i\pi/4} & e^{-i\pi/4} \end{bmatrix} = \frac{e^{-i\pi/4}}{\sqrt{2}}\begin{bmatrix} -i & 1 \\ i & 1 \end{bmatrix}$. Noe ignore the $e^{-i\pi/4}$ part and do $WW^2 = \frac{1}{2}\begin{bmatrix} i & -i \\ 1 & 1 \end{bmatrix}\begin{bmatrix} -i & 1 \\ i & 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \mathbf{I}$. Thus three iterations of the group commutator get you back to the identity. It does not escape from the larger group generated by the Pauli matrices and $\mathbf{S}$. This group (also when extended via tensor products to any number of qubits) is called the **Clifford group**. [Note: in lecture I used a different $\mathbf{V}'$, so $\mathbf{W}$ differed as well---but the end was equivalent. I also illustrated the correspondence of $\mathbf{S}$ to the permutation cycle (0123) and $\mathbf{H}$ to (01), under which W cycles $0, 1, 3$ but leaves 2 fixed, so it is a permutation of order 3 not 4.]

Like Hadamard, this is a source of quantum nondeterminism. Multiplying a whole unitary matrix by a unit scalar, even by -1, is not considered to change the quantum operation it represents.

Thus, using $\mathbf{S}$ does not really help us "break out" from the realm of the Pauli gates $\mathbf{I}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}$. Using $\mathbf{T}$ does, however. We can get a taste by composing $\mathbf{HTHT^*H}$ and $\mathbf{HTHT^*HTHT^*H}$. In fact, the group generated by $\mathbf{H}$ and $\mathbf{T}$ is *infinite*. It never cycles back.

Outputs and Measurements

There are various conventions about what it means for a family $[C_n]$ of quantum circuits to compute a function f on $\{0,1\}^*$, where f is an ensemble of functions f_n on $\{0,1\}^n$ and each C_n computes f_n. I like supposing that $f(x)$ is coded in $\{0,1\}^r$ where r depends only on n and giving C_n r-many output qubits separate from the n input qubits, plus some number m of ancilla qubits. (It is traditional, IMHO weirdly, to consider that the primordial input is always $\mathbf{0}^n$ and that for any other x, **NOT** gates are prepended onto the circuit for those lines i where $x_i = 1$.)

For *languages*, this means that the yes/no verdict comes on qubit $n+1$. Many references say to measure line 1 instead. (Using a swap gate between lines 1 and $n+1$ can show these conventions to be equivalent, but I prefer reserving lines 1 to n for *potential* use of the "copy-uncompute" trick, which is covered in section 6.3.) Even for languages, however, one evidently cannot get the most power if you need always to rig the circuit so that on any input $x \in \{0,1\}^n$, the output line always has a (standard-)basis value, i.e., is $\mathbf{0}$ with certainty or is $\mathbf{1}$ with certainty. Instead, one must **measure** it, whereupon the value $\mathbf{0}$ is given with some probability p, $\mathbf{1}$ with probability $1 - p$.

The math of measurements (at least of the kind of *pure states* we get in completely-specified circuits) is simple. At the end we have a quantum state Ψ of $n + r + m$ qubits, counting the output and any ancilla lines. It "is" a vector $(v_1, v_2, \dots v_Q) \in \mathbb{C}^Q$ where $Q = 2^{n+r+m}$. Numbering $\{0,1\}^{n+r+m}$ in canonical order as $z_1, \dots, z_S$, an **all-qubits measurement** gives any z_j with probability $|v_j|^2$. If we focus on just the r output lines, then any $y \in \{0,1\}^r$ occurs with probability

$$\sum_{j:\, z_j \text{ agrees with } y \text{ on the } r \text{ output lines}} |v_j|^2.$$

When $r = 1$ and $y = \mathbf{0}$ the sum is over all binary strings z_j that have a 0 in the corresponding places. To simplify the notartion, let p_x denote the probability of measuring 1 on the output qubit line.

The notion of uniformity is similar to that for ordinary Boolean circuits: it means that C_n can be written down in $n^{O(1)}$ (classical) time. We can finally define:

Definition: A language L belongs to BQP if there is a uniform family $[C_n]$ of polynomial-sized quantum circuits such that for all n and inputs $x \in \{0,1\}^n$,

$$\begin{array}{rcl} x \in L & \Longrightarrow & p_x \geq 3/4; \\ x \notin L & \Longrightarrow & p_x \leq 1/4. \end{array}$$

Aspects of Computing Functions

Let us view the 4-qubit Hadamard transform as a big matrix:

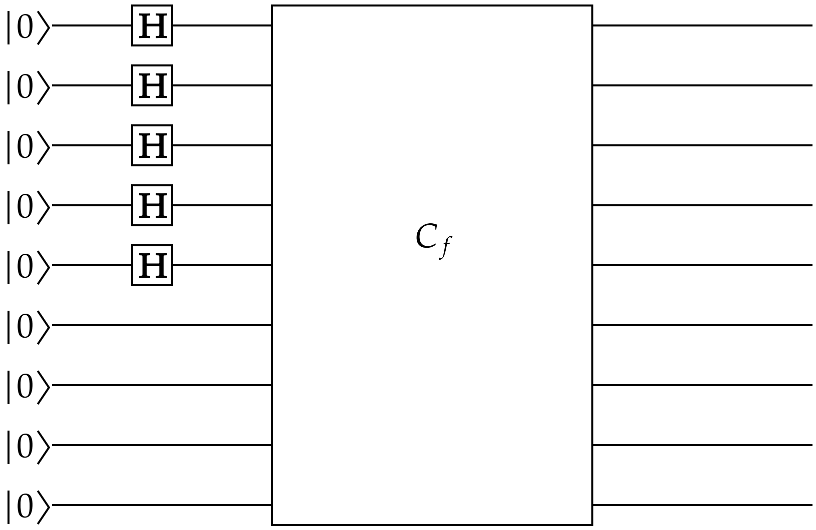
$$\frac{1}{4}\left[\begin{array}{ccccccccccccccccc}
\mathbf{H}^{\otimes 4} & 0000 & 0001 & 0010 & 0011 & 0100 & 0101 & 0110 & 0111 & 1000 & 1001 & 1010 & 1011 & 1100 & 1101 & 1110 & 1111 \\
0000 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0001 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
0010 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
0011 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 \\
0100 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
0101 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
0110 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
0111 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\
1000 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1001 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\
1010 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\
1011 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\
1100 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\
1101 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\
1110 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\
1111 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1
\end{array}\right]$$

$$\mathbf{H}^{\otimes n}[u,v] = (-1)^{u \bullet v}$$

We have argued that the Hadamard transform is feasible: it is just a column of n Hadamard gates, one on each qubit line. There is, however, one consequence that can be questioned. We observed that a network of Toffoli gates suffices to simulate any Boolean circuit C (of NAND gates etc.) that computes a function $f:\{0,1\}^n \to \{0,1\}^r$. The Toffoli network C_f actually computes the reversible form

$$F(x_1, \ldots, x_n, a_1, \ldots, a_r) = (x_1, \ldots, x_n, a_1 \oplus f(x)_1, \ldots, a_r \oplus f(x)_r).$$

The matrix $\mathbf{U_f}$ of C_f is a giant permutation martrix in the 2^{n+r} underlying coordinates. Yet if the Boolean circuit C has s gates, then we reckon that C_f costs $O(s)$ to build and operate. Now build the following circuit, which is illustrated with $n = 5$ and $r = 4$:



What this circuit piece computes is the **functional superposition** of f, defined as

$$|\Phi_f\rangle \;=\; \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle|f(x)\rangle.$$

The juxtaposition of two kets really is a tensor product, $|x\rangle \otimes |f(x)\rangle$. The abbreviated form above is "okayyy..." because $|x\rangle$ and $|f(x)\rangle$ individually belong to the standard basis. The whole state $|\Phi_f\rangle$, however, is far from belonging to the standard basis, and it (IMHO) has several issues.

One of them is highlighted by Holevo's Theorem, which is not covered *per se* but can be given the following informal statement:

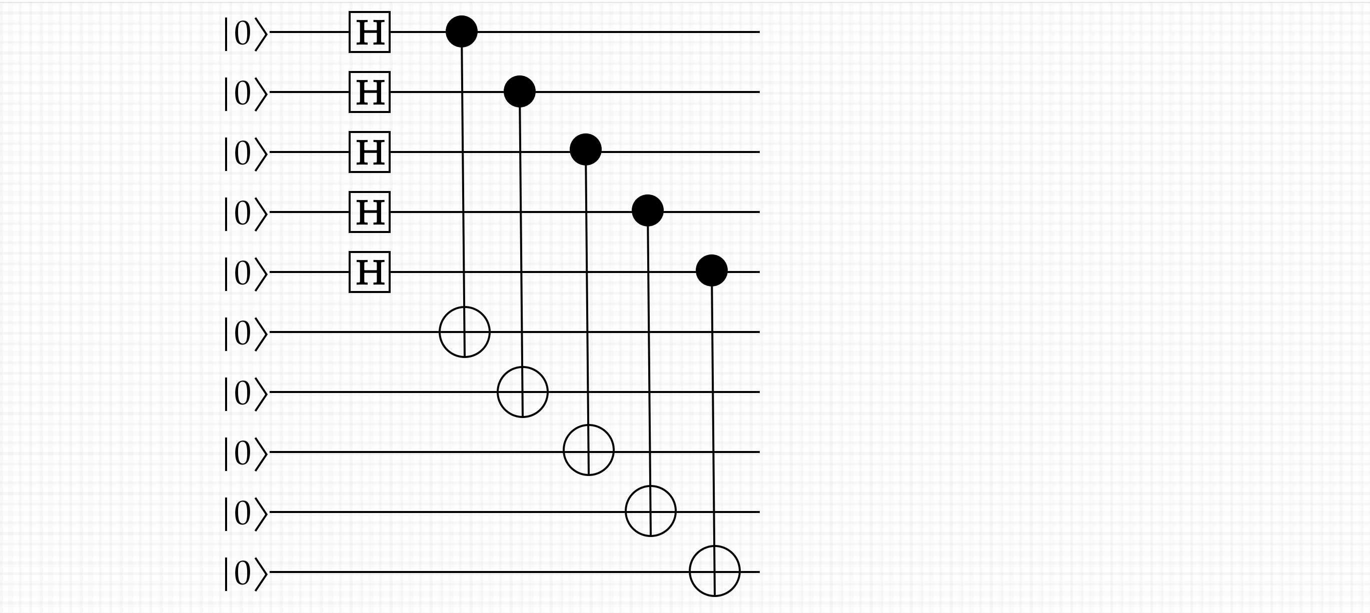
> **A quantum state on n qubits can store at most n bits of classical information.**

Let's think of this first about our n-qubit graph states Φ_G for n-node graphs G. (NB: Writing $|\Phi_G\rangle$ is OK but redundant since "Φ_G" is not a basic attribute---likewise the ket in "$|\Phi_f\rangle$" above just looks "quantum-y".) G can have up to $\binom{n}{2} \sim 0.5n^2$ edges. Thus G itself can encode $\Theta(n^2)$ bits of information, especially when the vertices are explicitly numbered $1 \ldots n$. However, the graph state holds only $n = o(n^2)$ bits. It follows that graph states are "lossy" for general graphs. They give full fidelity only for special classes of *sparse* or highly-regular/symmetrical graphs.

With Φ_f, however, the state looks like attempting to store exponentially many bits of information about the function f---as defined by its 2^n values $f(x)$ on inputs $x \in \{0, 1\}^n$. The sum has exponentially many terms. We can, however, get at most n distinguishable bits out of the state from any measurement. This is commensurate with the fact that it is produced by a circuit of $O(s + n)$ gates, especially when s itself is $O(n)$.

Nevertheless, the question remains of whether some exponential amount of "effort" must go in to the creation of $|\Phi_f\rangle$, instead of just $O(n)$ for the Hadamard transform plus $O(s)$ for the circuit. Or does the fact of only $O(s+n)$ gates mean that $|\Phi_f\rangle$ doesn't meaningfully reflect the exponentially many values taken by the function $f(x)$?

Let's ask this where the circuit C_f is just a bunch of $\mathbf{CNOT}$ gates. On five qubits,



computes the functional superposition

$$\frac{1}{\sqrt{32}} \sum_{x \in \{0,1\}^5} |x\rangle|x\rangle.$$

This is not the same as $|{+}{+}{+}{+}{+}\rangle \otimes |{+}{+}{+}{+}{+}\rangle$, because that is the equal superposition over all basis states for 10-bit binary strings, including all the cases of $|xy\rangle$ where the binary strings x and y of length 5 are different. An analogy is that for any set A of two or more elements, the Cartesian product of A with itself includes ordered pairs (x,y) with $x,y \in A$ but $x \neq y$, whereas the functional superposition is like the diagonal of the Cartesian product, namely $\{(x,x) : x \in A\}$. The functional superposition is entangled, just as we first saw in the case $n = 1$.

If we replace the five $\mathbf{H}$ gates by a subcircuit that prepares a general 5-qubit state

$$|\phi\rangle \;=\; a_0|00000\rangle + a_1|00001\rangle + \;\cdots\; + a_{30}|11110\rangle + a_{31}|11111\rangle,$$

then the five $\mathbf{CNOT}$ gates produce

$$D\big(|\phi\rangle\big) \;=\; a_0|0000000000\rangle + a_1|0000100001\rangle + \;\cdots\; + a_{30}|1111011110\rangle + a_{31}|1111111111\rangle.$$

This is not the same as $|\phi\rangle \otimes |\phi\rangle$, whose terms have coefficients $a_i a_j$ for all i and j. IMHO the notation $|\phi\rangle|\phi\rangle$ or $|\phi\phi\rangle$ can be unclear about what is meant, though I've freely used $|{+}{+}\rangle$ etc. as above. When $|x\rangle$ is a basis element in the basis used for notation, then there is no difference: both $|x\rangle \otimes |x\rangle$ and $D(|x\rangle)$ have the single term $|xx\rangle$ with coefficient $1 = 1^2$.

Feasible Diagonal Matrices (section 5.4)

We can continue the progression $\mathbf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$, $\mathbf{CZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$, by

$$\mathbf{CCZ} = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & -1 \end{bmatrix}, \quad \mathbf{CCCZ} = diag([1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,-1]),$$

and so forth. These are examples of a different kind of conversion of a Boolean function f besides the reversible form called F or C_f above. This is the matrix G_f defined for all indices u, v by

$$G_f[u,v] \;=\; \begin{cases} 0 & \text{if } u \neq v \\ -1 & \text{if } u = v \;\wedge f(u) = 1 \\ 1 & \text{if } u = v \;\wedge f(u) = 0 \end{cases}.$$

The above are G_{AND} for the n-ary AND function. The G stands for "Grover Oracle", though here I would rather emphasize that it is a concretely feasible operation. This ultimately leads to a theorem whose statement doesn't appear until chapter 6:

Theorem (6.2): If f is computable by a Boolean circuit with s gates, thgen G_f can be computed by a quantum circuit of $O(s)$ gates.

When $s \;=\; s(n)$ is polynomial in n, this makes a big contrast to G_f being a 2^n-sized diagonal matrix. We can also summarize a relationship to the previous language-based definition of BQP:

Theorem (not stated as such): If the language $L_f = \{x, y : \; f(x) \leq y\}$ belongs to BQP, then for every $\epsilon > 0$ and all n there are circuits $C_{n,\epsilon}$ of size $s(n) = n^{O(1)}$ (with as many output gates needed to write values $f(x)$ for $x \in \{0,1\}^n$), the probability that $C_{n,\epsilon}(x)$ correctly outputs $f(x)$ after measurement of its output gates is at least $1 - \epsilon$. This is true for both the "F_f" and "G_f" representations of f.

The nub of the proof is the---completely classical---fact that **binary search** using the language L_f works in polynomial time even though there are exponentially many values $f(x)$ to sift through.

Universal Gate Sets

The above theorems allow us to solidify our intuition about the power of quantum gates.

Definition. A set S of basic quantum gates is **universal** if every language/function in BQP can be computed by polynomial-sized circuits that use only gates in S.

Definition. The set S is **metrically universal** if for every unitary operation $\mathbf{U}$ on some number m of qubits, and $\epsilon > 0$, there is a circuit C on m qubits using finitely many gates from S such that for all m-qubit quantum states $\boldsymbol{\Phi}$, $||C\boldsymbol{\Phi} - \mathbf{U}\boldsymbol{\Phi}|| < \epsilon$. (The norm is the sup-norm, aka. ∞-norm.)

Theorem. The following gate sets are universal:

1. Hadamard, CNOT, and $\mathbf{T}$.
2. Hadamard and $\mathbf{CS}$.
3. Hadamard and Toffoli.

The first two sets are metrically universal. The third is not---simply because it doesn't use any complex numbers at all.

These facts are stated but not proved in the text; a key idea of the third is in the solved exercise 3.8 in chapter 3. But given the third fact, universality of $\mathbf{H} + \mathbf{CS}$ follows by the circuit equation for Toffoli gates given before, because $\mathbf{CZ}$ can be written as $\mathbf{CS} \cdot \mathbf{CS}$. And the simulation of $\mathbf{CS}$ by $\mathbf{H} + \mathbf{CNOT} + \mathbf{T}$ could be homework... This doesn't prove metric universality, however. Indeed, the only source I know for gate set 2 being metrically universal is the exercise section of lecture notes by John Preskill: https://www.preskill.caltech.edu/ph219/chap5_13.pdf (start on page 47). Those of you who are sharp on logic may not be convinced that "metrically universal" implies "universal" the way I worded it, because ϵ-errors on single gates might compound themselves when the gates are composed in a circuit---and also, how large is that "finitely many gates from S" part when m can grow with n rather than be fixed? The connection is enforced by the Solovay-Kitaev theorem and its efficient underlying algorithm, which shows that only $(\log n)^{O(1)}$ extra overhead in gates is needed---not even linear or polynomial overhead.

Another important fact to bear in mind is that the gate set $\{\mathbf{H}, \mathbf{CNOT}, \mathbf{CZ}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{S}\}$ is not metrically universal. Every circuit of these gates is simulatable in classical polynomial time. This is called the Gottesman-Knill theorem. My graduated PhD student Chaowen Guan and I improved the running time of this theorem in 2019 using a new analysis of (essentially) graph-state circuits. These gates and their ordinary and tensor products generate the **Clifford gate set**. One other notable member is $\mathbf{V} = \mathbf{HSH}$.

The most particular takeaway for (philosophical issues in) this course, however, concerns the extended series of gates we've mentioned before:

$$\mathbf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \mathbf{S} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \mathbf{T} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}, \mathbf{T}_{\frac{\pi}{8}} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/8} \end{bmatrix}, \mathbf{T}_{\frac{\pi}{16}} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/16} \end{bmatrix}, \mathbf{T}_{\frac{\pi}{32}} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/32} \end{bmatrix} \dots$$

Can we really engineer these super-fine angles? By (metric) universality, we don't have to: we can combine $\mathbf{T}$ with $\mathbf{H}$ (and $\mathbf{CNOT}$, but only $\mathbf{X}$ is needed for these particular gates) to emulate them. There is a web app for this. A useful technote: if you multiply $\mathbf{T}$ by the unit scalar $e^{-i\pi/8}$ you get

$$\begin{bmatrix} e^{-i\pi/8} & 0 \\ 0 & e^{i\pi/8} \end{bmatrix}.$$

This sets up some confusing nomenclature: $\mathbf{T}$ itself, not what I've called $\mathbf{T}_{\pi/8}$, is often called "the $\pi/8$ gate". The web app calls this "$R_z(\theta)$" with $\theta = \pi/4$. The Wybiral applet has "R2" as a redundant name for $\mathbf{S}$, "R4" for $\mathbf{T}$, but at least gives "R8" for the gate $\mathbf{T}_{\pi/8}$.

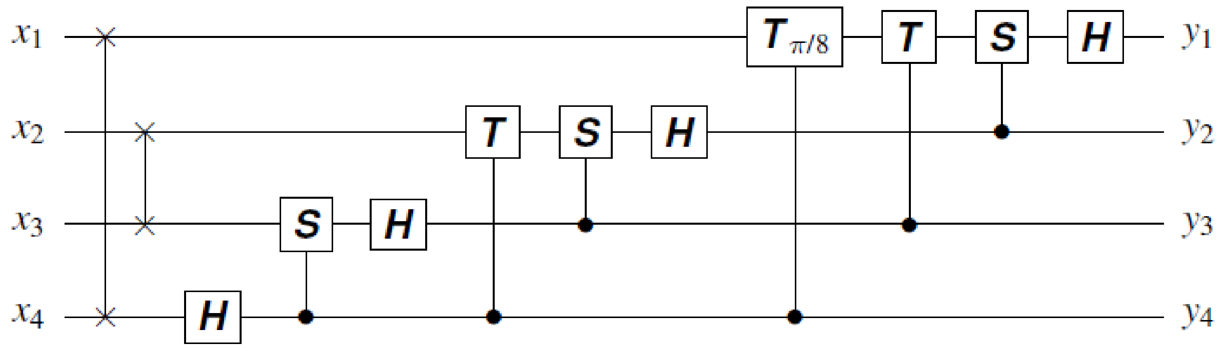
The Quantum Fourier Transform

Super-tiny angles are in the definition of the **QFT** itself. For any n, it takes $\omega_n = e^{2\pi i/N}$ where $N = 2^n$. With $n = 3$ and $\omega = e^{i\pi/4}$, the matrix together with its quantum coordinates is:

$$\begin{array}{c|cccccccc}
 & 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\ \hline
000 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
001 & 1 & \omega & i & i\omega & -1 & -\omega & -i & -i\omega \\
010 & 1 & i & -1 & -i & 1 & i & -1 & -i \\
011 & 1 & i\omega & -i & \omega & -1 & -i\omega & i & -\omega \\
100 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
101 & 1 & -\omega & i & -i\omega & -1 & \omega & -i & i\omega \\
110 & 1 & -i & -1 & i & 1 & -i & -1 & i \\
111 & 1 & -i\omega & -i & -\omega & -1 & i\omega & i & \omega
\end{array}
=
\begin{array}{c|cccccccc}
 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & \omega & \omega^2 & \omega^3 & -1 & \omega^5 & \omega^6 & \omega^7 \\
2 & 1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\
3 & 1 & \omega^3 & \omega^6 & \omega & -1 & \omega^7 & \omega^2 & \omega^5 \\
4 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
5 & 1 & \omega^5 & \omega^2 & \omega^7 & -1 & \omega & \omega^6 & \omega^3 \\
6 & 1 & \omega^6 & \omega^4 & \omega^2 & 1 & \omega^6 & \omega^4 & \omega^2 \\
7 & 1 & \omega^7 & \omega^6 & \omega^5 & -1 & \omega^3 & \omega^2 & \omega
\end{array}$$

$$\mathbf{QFT}[i,j] = \omega^{ij}$$

The above "R_z series"---and their controlled versions $\mathbf{CS}, \mathbf{CT}, \mathbf{CT}_{\pi/8}, \dots$, gives us a recursive way to build the n-qubit QFT using only $O(n^2)$ unary and binary gates. This is already evident from the four-qubit illustration in the textbook (where the two gates on the left are swap gates):



For $n = 5$ the next bank uses $1/32$, then $n = 6$ uses angles of $1/64$ of a circle, and so on. Soon the angles would be physically impossible so the gates could never be engineered. But:

- The metric universality of $\mathbf{H} + \mathbf{CNOT} + \mathbf{T}$ says you only need to engineer "pieces of eight" for angles---and the Solovay-Kitaev algorithm shows you how to build the approximating circuits with only $(\log n)^{O(1)}$ extra multiplicative overhead. Such "polylog" factors are often ignored under the notation of saying the whole simulation of the n-qubit QFT needs only $\widetilde{O}(n^2)$ gates.
- Doing this with $\mathbf{H} + \mathbf{CS}$ instead needs only quarter-circle angles---that is, i and $-i$.
- With Hadamard + Toffoli the only angles involved are 0 and π. You wind up simulating the real and imaginary parts of QFT computations under two separate binary encodings.

I retain, however, a "meta-physical" objection that the inherent instability in tiny angles still infects these circuits when attempts are made to engineer them physically and keep them free of noise. One can cite LIGO as a supreme success case where tiny physical displacements are magnified and detected in a roughly analogous manner. But that has a fixed physical limit of resolution, whereas the Shor's algorithm application of $\mathbf{QFT}_m$ wants m to grow at least linearly with the overall problem instance size n. Well, if the obstacle is actually *physical*, not just "meta-", it will entail the discovery of a new physical law that modulates quantum mechanics.

Maybe $\mathbf{QFT}_n$ isn't impossible. We can show, however, that a simpler-looking task---one we take for granted in classical computing---is really impossible in the quantum realm. This also exemplifies how interpreting quantum circuits can be tricky unless you apply the principle of linearity strictly.

The No-Cloning Theorem

It's good enough to prove this in the case of copying one qubit in a two-qubit circuit.

Theorem: There is no 4×4 unitary operation U such that for any single-qubit quantum state $\phi = ae_0 + be_1$, $U(\phi \otimes e_0) = \phi \otimes \phi$.

Proof: Suppose U existed. Then $U(e_0 \otimes e_0) = e_0 \otimes e_0$ and $U(e_1 \otimes e_0) = e_1 \otimes e_1$. So by linearity,

$$U(\phi \otimes e_0) = U((ae_0 + be_1) \otimes e_0) = U(a(e_0 \otimes e_0) + b(e_1 \otimes e_0))$$

$$= aU(e_0 \otimes e_0) + bU(e_1 \otimes e_0) = a(e_0 \otimes e_0) + b(e_1 \otimes e_1) = ae_{00} + be_{11}.$$

But $U(\phi \otimes e_0)$ is supposed to equal $\phi \otimes \phi$, which

$$= (ae_0 + be_1) \otimes (ae_0 + be_1) = a^2 e_{00} + abe_{01} + abe_{10} + b^2 e_{11}.$$

The only way these quantities can be equal is if $ab = 0$. That boils down to saying that *the only single-qubit states that can be copied are the two standard basis states.* (Note that this is a much stronger conclusion than the theorem stated.) ☒

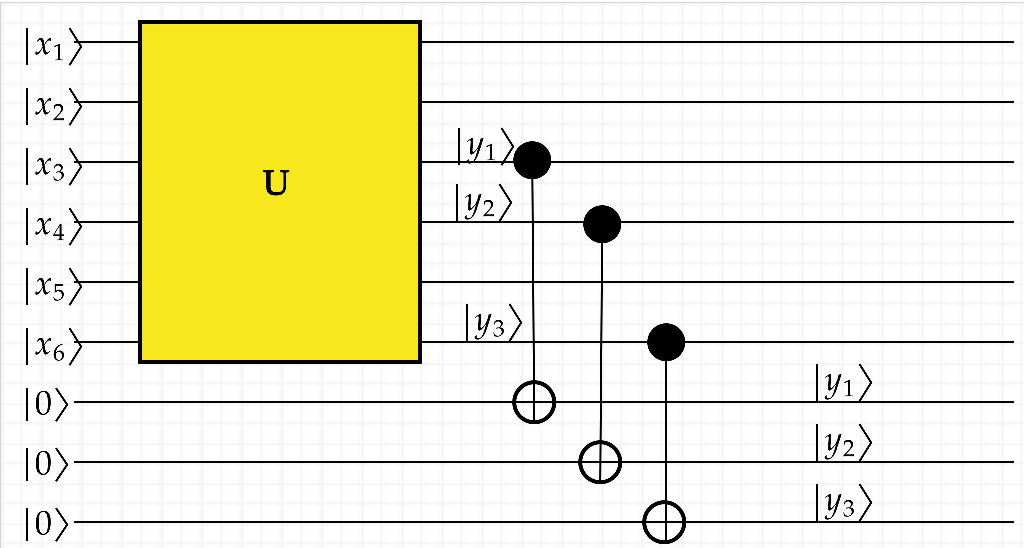
Summarizing, we have learned four fundamental theorems:

1. **Holevo's Theorem**: n qubits can hold only n bits of classical information. This means that individual graph states can be "quadratically lossy"---and functional superpositions using $2n$ qubits can be "exponentially lossy". They will be effective only in *special* purposes.
2. **Solovay-Kitaev Theorem** (informal statement): For any unitary operation $\mathbf{U}$ on n qubits, and $\epsilon > 0$, we can construct a quantum circuit $\mathbf{C}$ using just $\mathbf{H}$ and $\mathbf{CS}$ gates, or $\mathbf{H}$, $\mathbf{CNOT}$,and $\mathbf{T}$ gates (or any finite metrically universal set of gates) such that for every unit vector $\mathbf{x}$, $||\mathbf{Ux} - \mathbf{U_C x}||_2 \leq \epsilon$. Under pertinent notions of the "specification size" s of $\mathbf{U}$, both the size of $\mathbf{C}$ and the time needed to build it are on the order of $s \cdot \log(1/\epsilon)^{O(1)}$. An improvement for single-qubit circuits is GridSynth (app).
3. **Gottesman-Knill Theorem** (informal statement): Quantum cicuits using only Clifford gates can be "de-quantized" (i.e., simulated in deterministic polynomial time).
4. **No-Cloning Theorem**: No unitary operation U on $2n$ qubits gives $U(\mathbf{x} \otimes |0^n\rangle) = \mathbf{x} \otimes \mathbf{x}$ for all (or even a few) n-qubit quantum states $\mathbf{x}$, unless $\mathbf{x}$ is a standard-basis vector. (Quasi-exception: you can make multiple copies $\mathbf{y}_i$ of a state $\mathbf{y}$, any one of which can be "de-crypted" back to $\mathbf{x}$, but the copies are entangled so that doing this for any one copy destroys the ability to recover $\mathbf{x}$ from all the others.)

You *can* copy all of the standard basis states at once, provided the input is exactly one of them, not a superposition. This needs only a bunch of $\mathbf{CNOT}$ gates, leading in to the next topic.

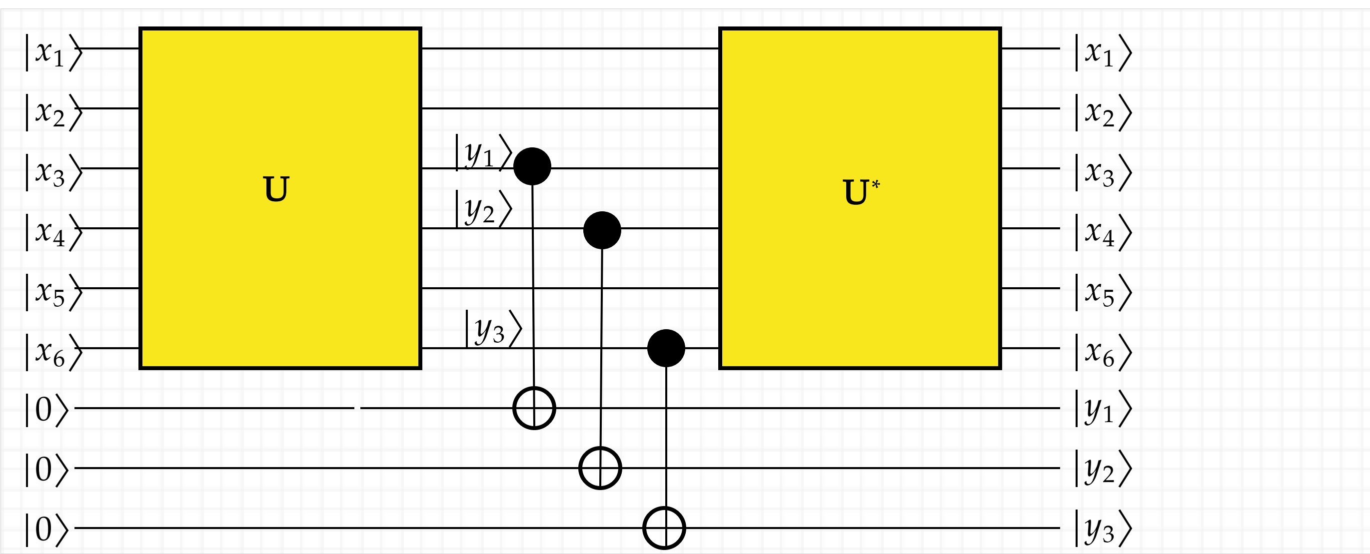
The Copy Uncompute Trick

Suppose we *know in advance* that at a certain point in a quantum circuit C on a particular input x (that is to say, e_x), some set of r qubit lines will be in a standard basis state e_y. Then we can insert $\mathbf{CNOT}$ gates between each of those lines and one of r fresh qubit lines to make a copy of e_y:



(We will later mirror the gates except the last one giving the function value d in order to reset the ancilla qubits 7--9 to $|0\rangle$, so that all qubits except the last keep their given basis values.)

If we then follow up with the inverse $\mathbf{U}^*$ of $\mathbf{U}$, then we also restore the input lines $x_1 \cdots x_n$ to what they were:



Note: this works only when it really is true that the selected lines have separated basis state values at that juncture. An example where it fails is with $n = 1$ and $r = 1$, the circuit `H 1 CNOT 1 2 H 1` which creates the operation we called E. Also recall above that when U does the Hadamard transform of $|0^n\rangle$ to create $\mathbf{y} = |+^n\rangle$, the $\mathbf{CNOT}$ gates then create the functional superposition summing $|x\rangle|x\rangle$ over all $x \in \{0,1\}^n$, which is not the same as $\mathbf{y} \otimes \mathbf{y} \; = \; |+^{2n}\rangle$.

[Show H CNOT H example in Quirk (little-endian) (flipped).]

On input e_{00}, that is, $x_1 = x_2 = 0$, the first Hadamard gate gives the control qubit a value that is a superposition. Hence, the second Hadamard gate does *not* "uncompute" the first Hadamard to restore $z_1 = 0$. The action can be worked out by the following matrix multiplication (with an initial factor of $\frac{1}{2}$):

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

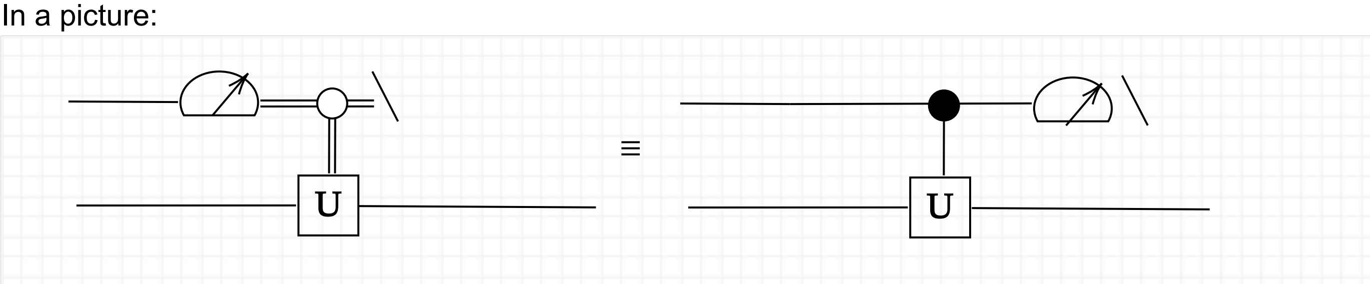
This maps e_{00} to $\frac{1}{2}[1, 1, 1, -1]$, thus giving equal probability to getting 0 or 1 on the first qubit line.

The Deferred Measurement Principle (section 6.6)

THEOREM 6.3 If the result b of a one-place measurement is used only as the test in one or more operations of the form "if b then $\boldsymbol{U}$," then exactly the same outputs are obtained upon replacing $\boldsymbol{U}$ by the quantum controlled operation $\boldsymbol{CU}$ with control index the same as the index place being measured and measuring that place later without using the output for control.

Proof. Suppose in the new circuit the result of the measurement is 0. Then the $\boldsymbol{CU}$ acted as the identity, so on the control index, the same measurement in the old circuit would yield 0, thus failing the test to apply $\boldsymbol{U}$ and so yielding the identity action on the remainder as well. If the new circuit measures 1, then because $\boldsymbol{CU}$ does not affect the index, the old circuit measured 1 as well, and in both cases the action of $\boldsymbol{U}$ is applied on the remainder. $\square$

In a picture:



What this does is legitimize the policy of having measurements only at the end of a circuit.

An Interesting Unitary Operation

Let J_n stand for the all-1s matrix of n qubits. J_n itself is $2^n \times 2^n$. As an example with $n = 2$,

$$J_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

This is Hermitian but not unitary---far from it. Actually, it equals the outerproduct $|{++}\rangle\langle{++}|$ but multiplied by 4. If we write in boldface $\mathbf{J}_n = |+^n\rangle\langle +^n|$, then $\mathbf{J}_n = \frac{1}{N} J_n$ where $N = 2^n$. With this normalization, we have (ordinary matrix multiplication, not tensoring)

$$\mathbf{J}_n^2 = |+^n\rangle\langle +^n| \cdot |+^n\rangle\langle +^n| = |+^n\rangle\big(\langle +^n||+^n\rangle\big)\langle +^n| = |+^n\rangle \cdot 1 \cdot \langle +^n| = \mathbf{J}_n.$$

(Math Jargon: this means $\mathbf{J}_n$ is **idempotent**.) Now define

$$\mathbf{R}_n = 2\mathbf{J}_n - \mathbf{I}_n,$$

where $\mathbf{I}_n$ is the $N \times N$ identity matrix, same as the 2×2 identity matrix tensored with itself n times. For $n = 2$ we get:

$$\mathbf{R}_2 = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix}.$$

Now we can verify that the matrix on the right is unitary. It resembles the matrix we earlier called $\mathbf{E}$ but that had the -1 entries going southwest to northeast instead. Now let's apply to a generic vector $\mathbf{u} = [a_1, a_2, a_3, a_4]^T$:

$$\mathbf{R}_2\, u = 2\mathbf{J}_2\, u - \mathbf{I}_2 u = \frac{a_1 + a_2 + a_3 + a_4}{2}[1, 1, 1, 1]^T - u$$

Is this unitary? Note: $\mathbf{R}_n^2 = (2\mathbf{J}_n - \mathbf{I}_n)(2\mathbf{J}_n - \mathbf{I}_n) = 4\mathbf{J}_n^2 - 2\mathbf{J}_n - 2\mathbf{J}_n + \mathbf{I}_n = \mathbf{I}_n.$

So $\mathbf{R}_n^2$ is a square root of the identity operator, and this is enough to make it unitary. Thus if we apply $\mathbf{R}_2$ a second time (and generally with $\mathbf{R}_n$), we get u back again. Thus $\mathbf{R}_n$ gives a reflection of u around the all-1s vector (that is, around $|+^n\rangle$). We will use a version of this in Grover's algorithm later.