CSE439/510 Week 5: Building and Visualizing Quantum Circuits

Computing Functions

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

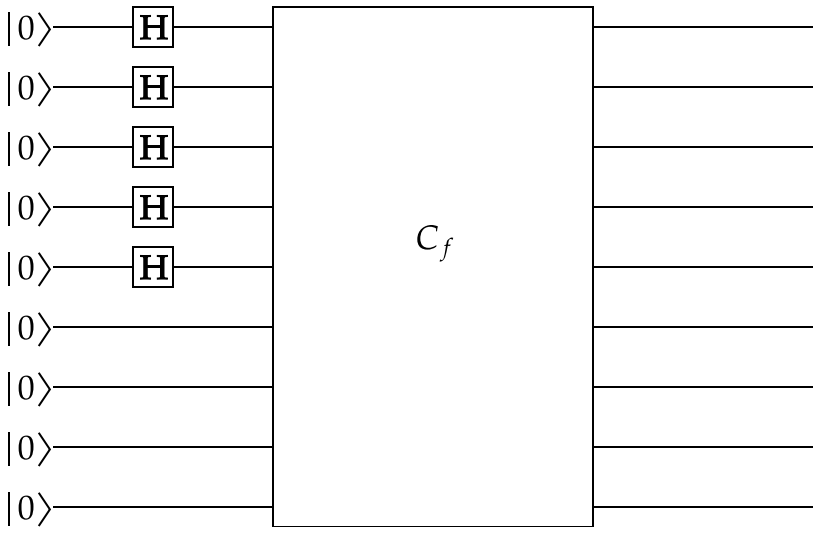
$$\frac{1}{4}\begin{bmatrix}
\mathbf{H}^{\otimes 4} & 0000 & 0001 & 0010 & 0011 & 0100 & 0101 & 0110 & 0111 & 1000 & 1001 & 1010 & 1011 & 1100 & 1101 & 1110 & 1111 \\
0000 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0001 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1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

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

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:

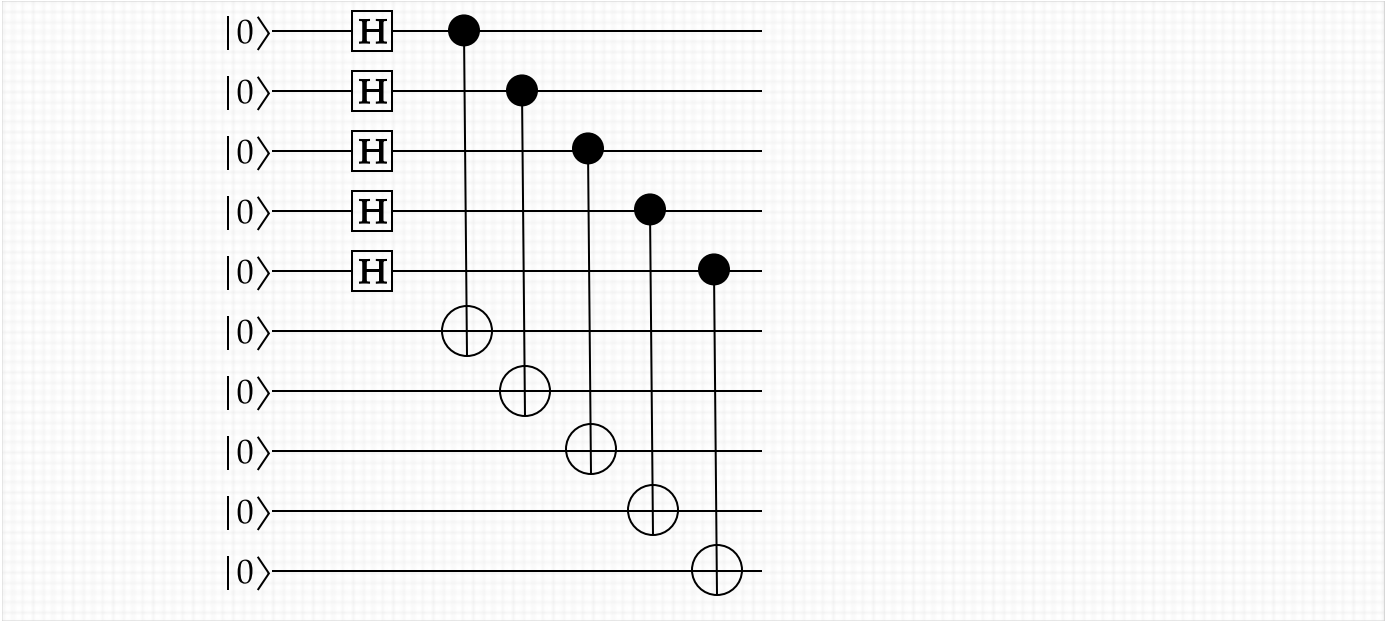
A quantum state on n qubits can store at most n bits of classical information.

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 **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 **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 **CNOT** gates produce

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

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 definition of BQP which was based on languages, i.e., on yes/no decision problems.

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 Φ, $||C\Phi - \mathbf{U}\Phi|| < \epsilon$. (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 $\mathbf{V} = \mathbf{HSH}$. 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. Its matrix 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 $\frac{1}{\sqrt{2}}\begin{bmatrix} i & 1 \\ 1 & i \end{bmatrix}$. 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 $\mathbf{I, X, Y, Z}$. Using $\mathbf{T}$ does, however. We can get a taste by composing $\mathbf{HTHT^*H}$ and $\mathbf{HTHT^*HTHT^*H}$.

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

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

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 $\mathbf{T}_{\pi/8}$.

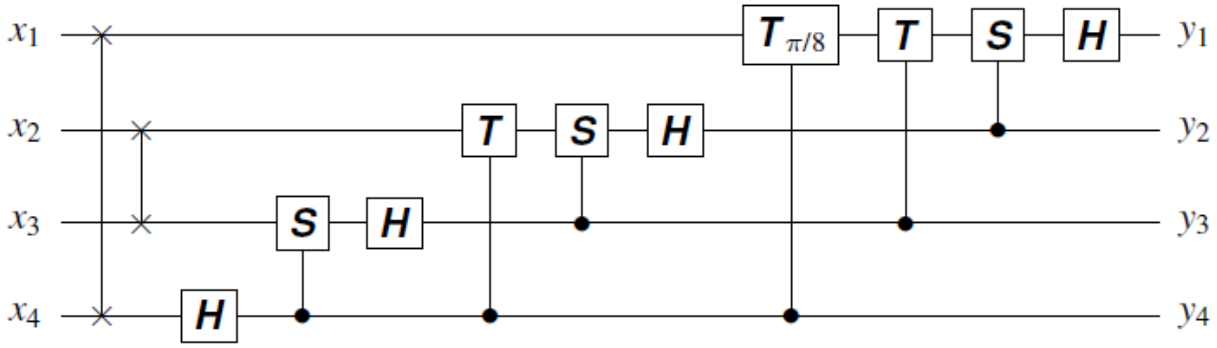
The Quantum Fourier Transform

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 $\mathbf{CS}, \mathbf{CT}, \mathbf{CT}_{\pi/8}, \ldots$, 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

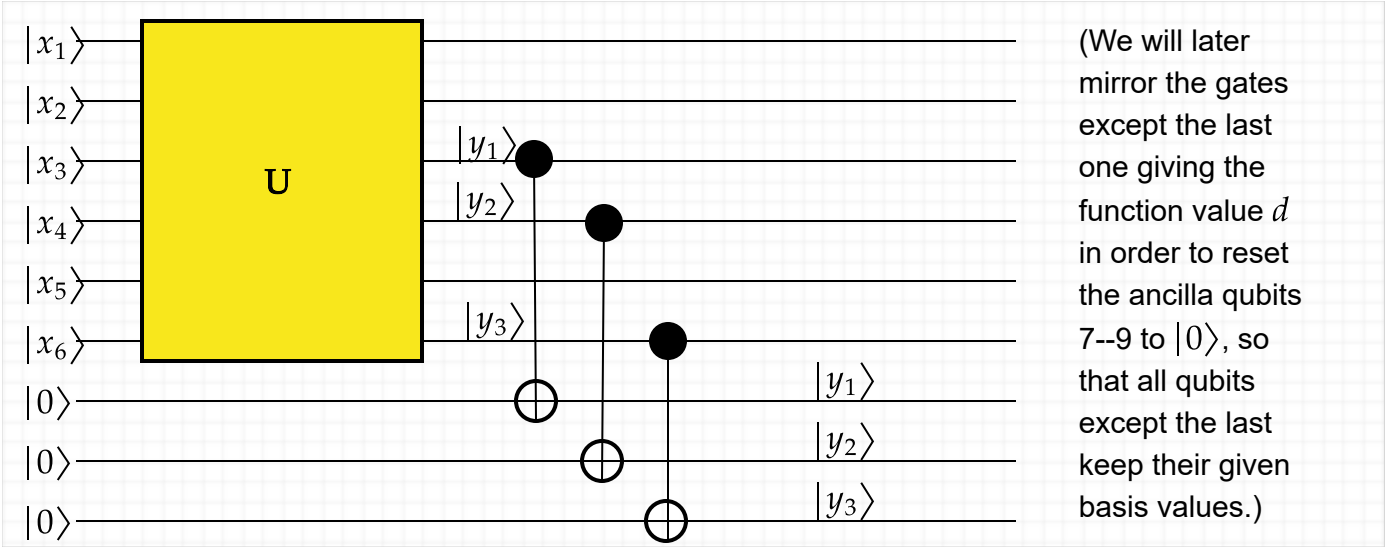
$$= (ae_0 + be_1) \otimes (ae_0 + be_1) = a^2e_{00} + abe_{01} + abe_{10} + b^2e_{11}.$$

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.) ☒

And indeed there is a 4×4 unitary matrix that can do this, namely $\mathbf{CNOT}$. This leads to the next topic.

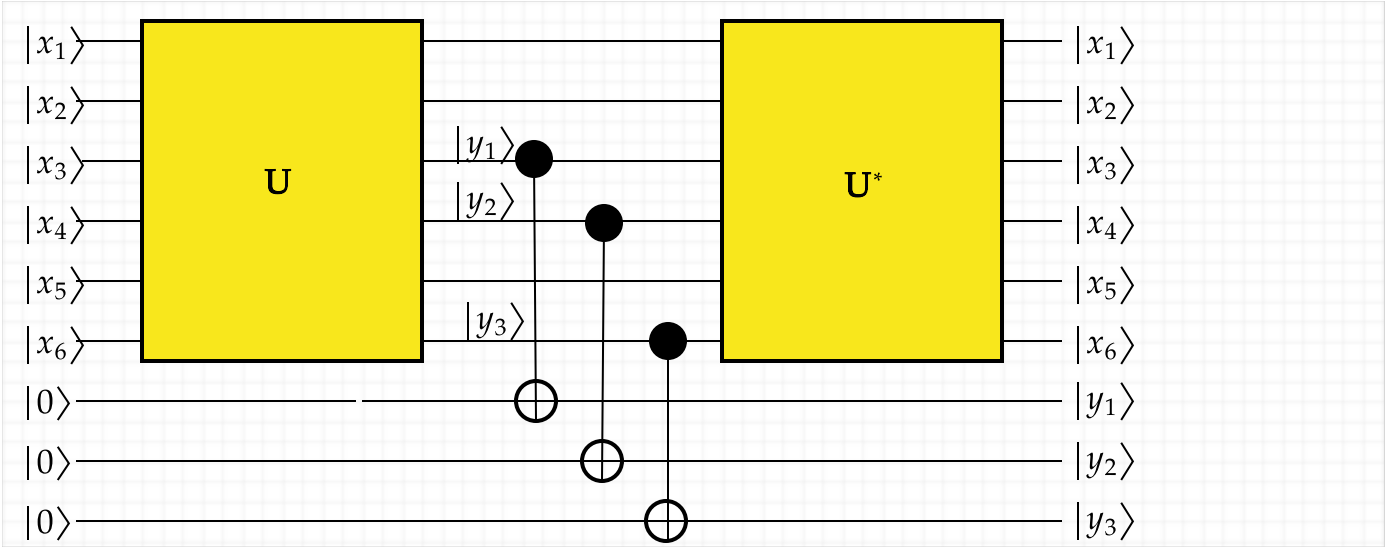
The Copy Uncompute Trick

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.

[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 $\frac{1}{2}$):

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

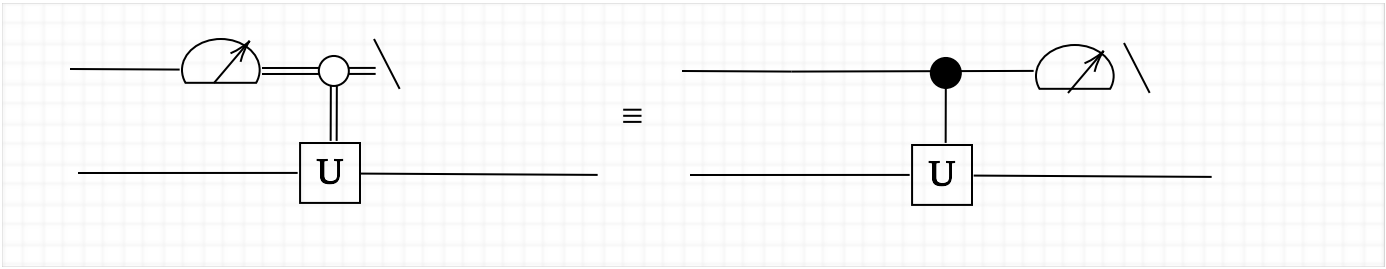
This maps e_{00} to $\frac{1}{2}[1, 1, 1, -1]$, thus giving equal probability to getting 0 or 1 on the first qubit line.

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. □

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 $\mathbf{u} = [a_1, a_2, a_3, a_4]^T$:

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

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 $\mathbf{R}_n$ 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 this is 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.

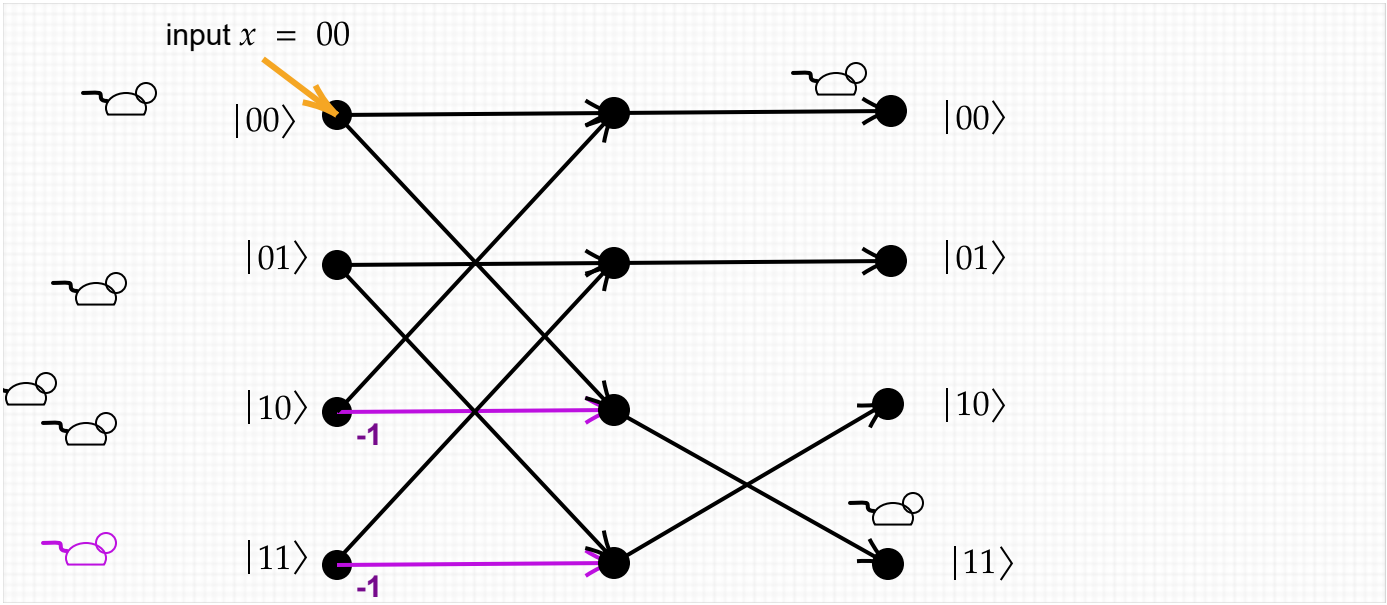
Reckoning and Visualizing Circuits and Measurements (chapter 7)

There are basically three ways to "reckon" a quantum circuit computation on q total qubits, $Q = 2^q$:

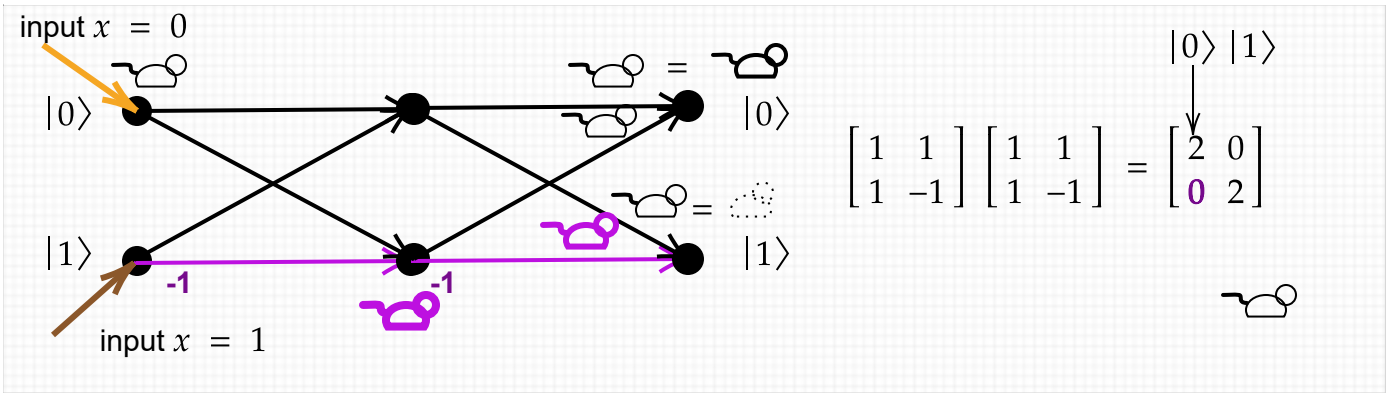
1. Multiply the $Q \times Q$ matrices together---using *sparse-matrix techniques* as far as possible. If BQP $\neq$ P and you try this on a problem in the difference then the sparse-matrix techniques must blow up at some (early) point. The downside is that the exponential blowup is paid early; the upside is that once you pay it, the matrix multiplications don't get any worse, no matter how more complex the gates become. This is often called a "Schrödinger-style" simulation.
2. Any product of s-many $Q \times Q$ matrices can be written as a single big sum of s-fold products. For instance, if A, B, C, D are four such matrices and u is a length-Q vector, then
$$ABCDu[i] = \sum_{j,k,l,m=0}^{Q-1} A[i,j] \cdot B[j,k] \cdot C[k,l] \cdot D[l,m] \cdot u[m].$$
Every (*nonzero*) product of this form can be called a (*legal*) **path** through the system. [As hinted before, in a quantum circuit, u will be at left---on an input x, it will be the basis vector $\mathbf{e}_{x0^{r+m}} = |x0^{r+m}\rangle$ under the convention that 0s are used to initialize the output and ancilla lines---and D will be the first matrix from gate(s) in the circuit as you read left-to-right. Thus the output will come out of A, which is why it is best to visualize the path as coming in from the top of the column vector u, going out at some row m (where u_m is nonzero---for a standard basis vector, there is only one such m), then coming in at column m of D, choosing some row l to exit (where the entry $D[l, m]$ is nonzero), then coming in at column l of C, and so on until exiting at the designated row i of A. This is the discrete form of Richard Feynman's **sum-over-paths** formalism which he originally used to represent integrals over quantum fields (often with respect to infinite-dimensional Hilbert spaces). The upside is that each individual path has size $O(s)$ which is linear not exponential in the circuit size. The downside is that the number of nonzero terms in the sum can be far worse than Q and doubles each time a Hadamard gate (or other nondeterministic gate) is added to the circuit.
3. Find a way to formulate the matrix product so that the answer comes out of symbolic linear algebra---if possible!

For the textbook, I devised a way to combine the *downsides* of 1 and 2 by making an exponential-sized "maze diagram" up-front but evaluating it Feynman-style. Well, the book only uses it for $1 \leq Q \leq 3$ and I found that the brilliant Dorit Aharonov had the same idea. All the basic gate matrices have the property that all nonzero entries have the same magnitude---and when normalizing factors like $\frac{1}{\sqrt{2}}$ are collected and set aside, the Hadamard, **CNOT**, Toffoli, and Pauli gates (ignoring the global i factor in $\mathbf{Y}$) give just entries $+1$ or -1, which become the only possible values of any path. That makes it easier to sum the results of paths in a way that highlights the properties of **amplification** and **interference** in the "wave" view of what's going on. The index values $m, l, k, j, i, \ldots$ become "locations" in the wavefront as it flows for time s, and since it is discrete, the text pictures packs of...well...spectral lab mice running through the maze.

One nice thing is that you can read the mazes left-to-right, same as the circuits. Here is the **H + CNOT** entangling circuit example: [Note: The mice are sometimes left in final positions, sometimes in a startup or midway position, for what I demonstrated in lecture.]

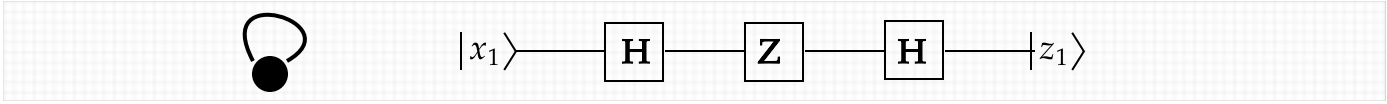


No interference or amplification is involved here---the point is that if you enter at $|00\rangle$, then $|00\rangle$ and $|11\rangle$ are the only places you can come out---and they have equal weight. To see interference, you can string the "maze gadgets" for two Hadamard gates together:

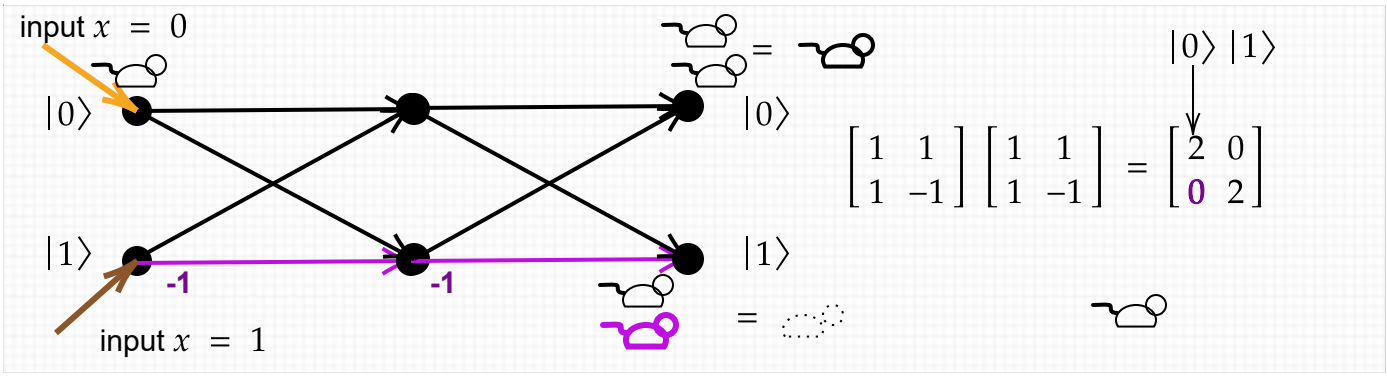


In linear-algebra terms, all that happened at lower right was $1 \cdot 1 + -1 \cdot 1$ giving 0. But the wave interference being described that way is a real physical phenomenon. Even more, according to Deutsch the two serial Hadamard gates branch into 4 universes, each with its own "Phil the mouse" (which can be a photon after going through a beam-splitter). One of those universes has "Anti-Phil", who attacks a "Phil" that tries to occupy the same location (coming from a different universe) and they fight to mutual annihilation.

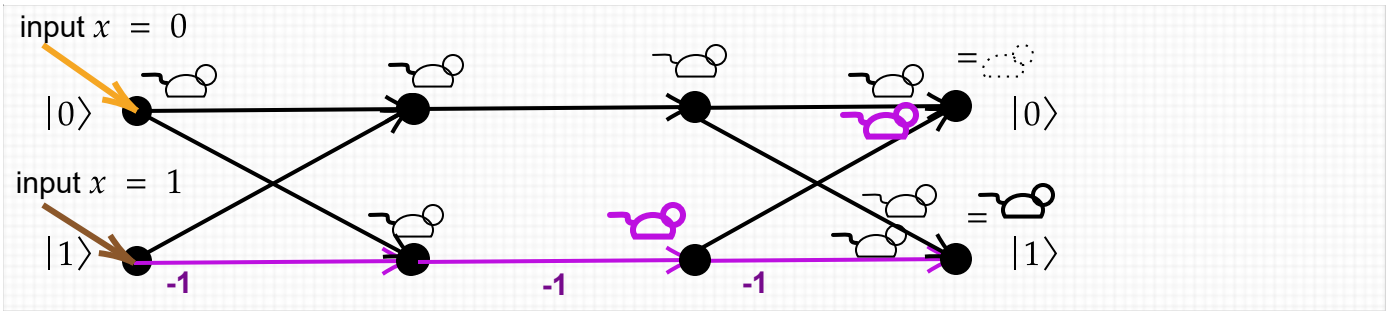
Can we build any interesting things with just a few qubits? Yes, in fact. Even the simplest **graph state circuit**---for a graph of just one node with a self-loop---is instructive to visualize.



We have seen the equation $\mathbf{HZH} = \mathbf{X}$. How is this reflected when we visualize the quantum properties? There is only one change from the "maze" for two $\mathbf{H}$-gates canceling, which was:



The change is to insert a stage that again has a -1 on the $|1\rangle$ basis value but no "crossover":



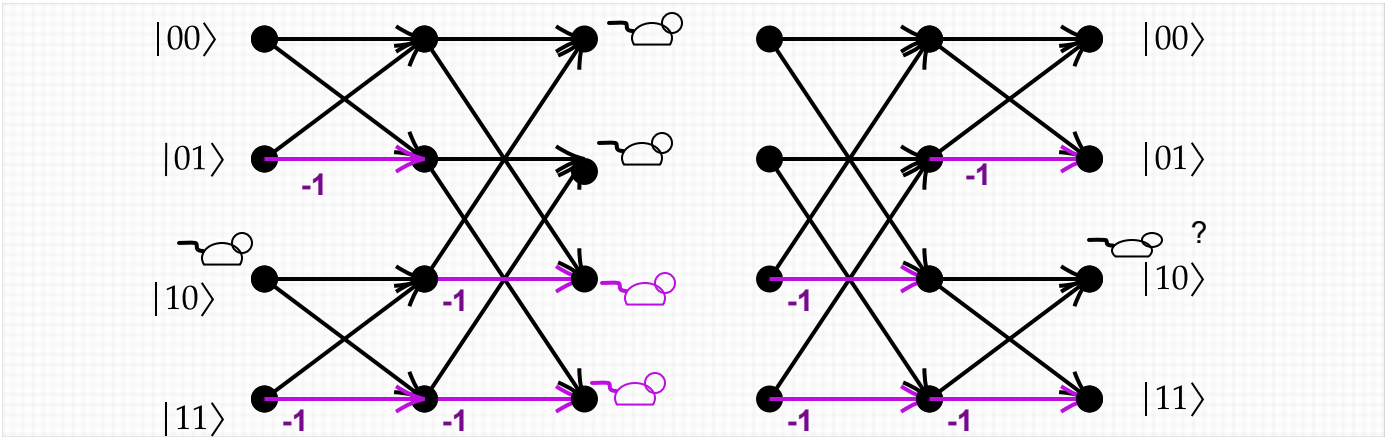
This time, when "Phil" starts running from $|0\rangle$ at left, the "mice" cancel at $z = |0\rangle$ and amplify at $z = |1\rangle$. And on input $x = |1\rangle$ they output the basis state $|0\rangle$. The result is Boolean NOT, i.e., $\mathbf{X}$.

[Footnote: A basic outcome $|z\rangle$ for the circuit C on input x has amplitude $\langle z|U_C|x\rangle$, not $\langle x|U_C|z\rangle$ as I've once been guilty of writing. Perhaps the diagrams should write the bra-form, $\langle 0|$ and $\langle 1|$ and so on, for z at right to emphasize this. But we've identified the ket-form with the notion of "outcome"; this is the form that would be given as input to a further piece of the circuit. This dilemma is another reason why Lipton and I first tried for a "handedness-free" approach.]

Phenomena of interest (tracing the "mice" is analogous to propagating a waveform):

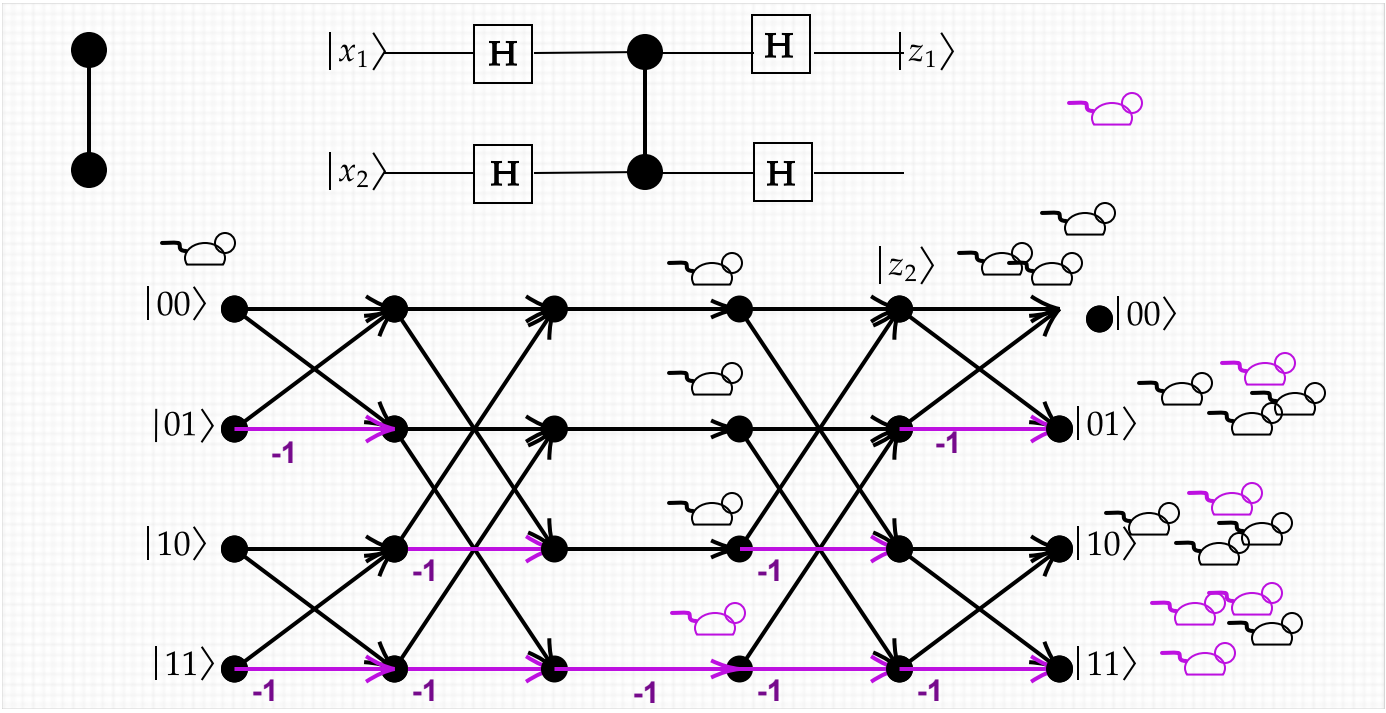
1. Superposition
2. Amplification
3. Phase changes
4. Interference.

For graph state circuits of 2 nodes we need 2 qubits. The Hadamard transform of two qubits is diagrammed as at left and right. It does not matter what order the two $\mathbf{H}$ gates go in.



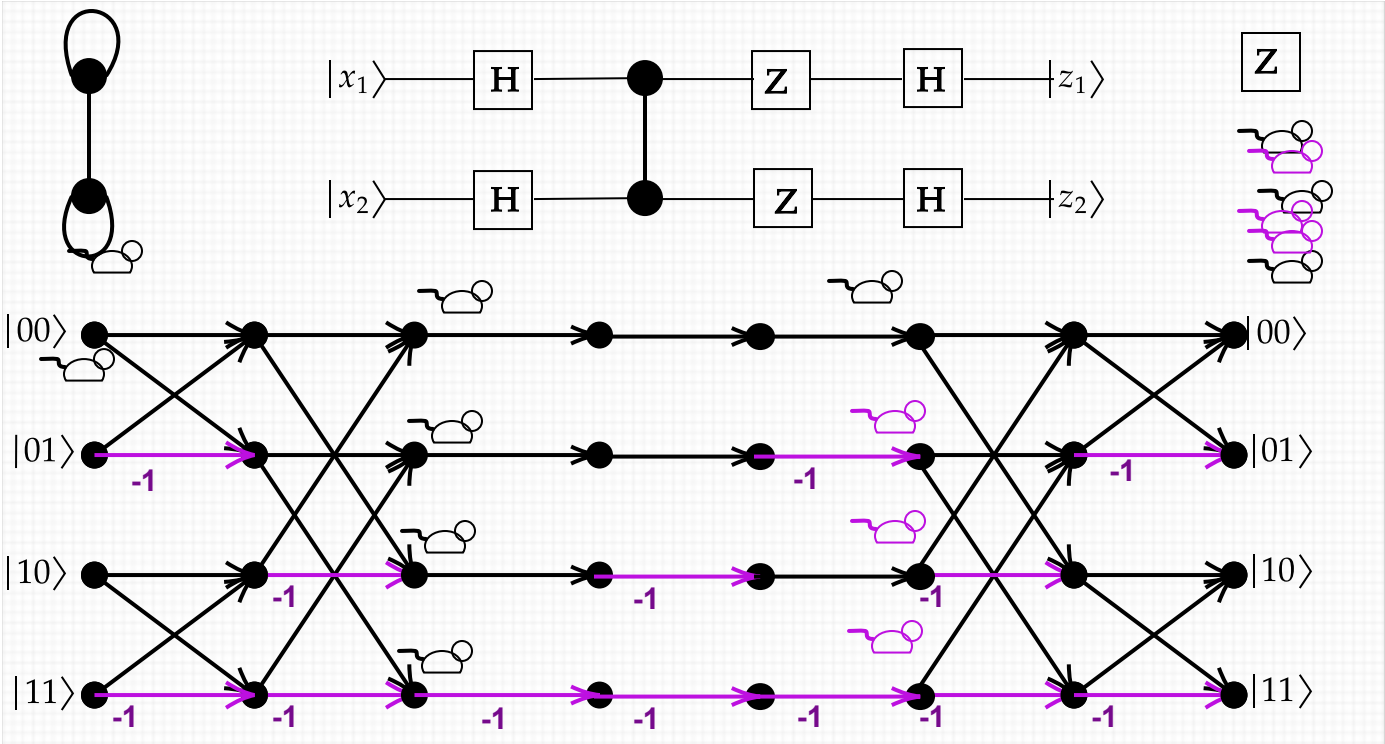
Note that the mouse running from $|00\rangle$ encounters no phase change, nor mice ending at $|00\rangle$ regardless of origin. This simply expresses that the Hadamard transform (and the QFT too) have every entry $+1$ (divided by the normalizing constant $R = \sqrt{2^n}$) in the row and column for $|00\rangle$. We will focus on the amplitude of getting $|00\rangle$ as output given $|00\rangle$ as input. If G is the graph, C_G the graph-state circuit, and U_G the unitary operator it computes, then the amplitude we want is $\langle 00|U_G|00\rangle$.

The simplest two-node G has a single edge connecting the two nodes. This introduces a single **CZ** gate between the qubits standing for the nodes.



If we take the two Hadamard gates away from line 1, then we have **H** 2 **CZ** 1 2 **H** 2, which is equivalent to **CNOT**. But with them, we get equal superpositions once again. Most in particular, the amplitude of $\langle 00|U_G|11\rangle$ (= $\langle 11|U_G|00\rangle$) is nonzero. [The lecture also noted how $\frac{1}{2}[1,1,1,-1]^T$ is a fixed point of $\mathbf{H}^{\otimes 2}$, ignoring multiplication by the unit scalar -1.]

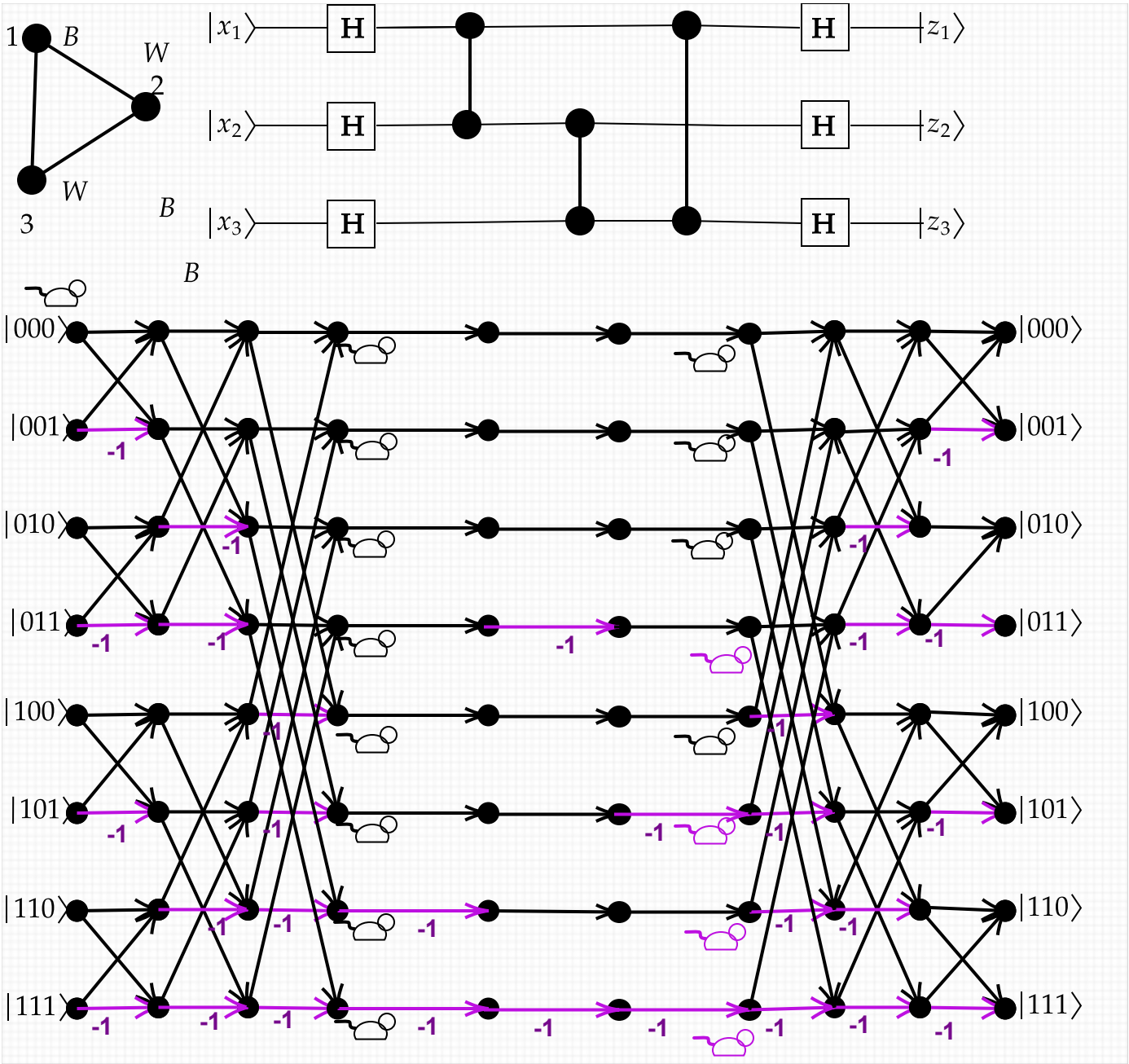
Now let's try a graph that adds a loop at each node. We can call it the "Q-Tip" graph:



The -1 phase shifts for the $\mathbf{Z}$ gates go on the basis states that have a 1 on line 1 or 2, respectively. Now the amplitude value $\langle 00|U_G|00\rangle$ is negative. Its sign does not affect the probability and the state still gives an equal superposition.

It does not matter whether we put the $\mathbf{Z}$ gates "before" or "after" the $\mathbf{CZ}$. The diagonal matrices all commute, and this is clear from how the paths go straight across without branching. We could simply make the whole graph into one diagonal gate with phase shifts that multiply the -1 factors along each row. A related thing to note is that if we repeat an edge or loop, then the two cancel completely. It's as if we have a graph with edges defined by even-odd parity rather than number.

Now let's try a three-node graph, the triangle:



For computing the amplitude $\langle 000|U_G|000\rangle$ it is not necessary to follow the "mice" through the Hadamard parts of the "maze". The mice entering the graph part from $x = |000\rangle$ are all positive, and the mice going to $z = |000\rangle$ will not change color once they leave the graph. So we need only track the middle portion and count how many mice are + and how many are −. For the triangle graph, the answer is: four of each. They **cancel**. So $\langle 000|U_G|000\rangle = 0$.

This leads us to more insight and a strategy for determining this amplitude for a general n-node graph $G = (V, E)$:

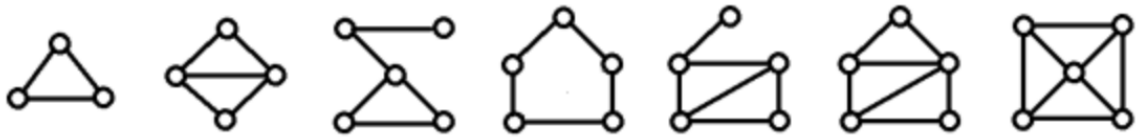
- Every basis state $|x\rangle$ with $x \in \{0,1\}^n$ corresponds to a **2-coloring** χ_x of the vertices. Say a node u is **black (B)** if $x_u = 1$, white (W) if $x_u = 0$. (The Greek letter χ (*chi*) looks like an X and

indeed X is its capital form, but the Greek letter that sounds like English X is ξ (*xi*) with capital Ξ. The χ gives the *ch* in *chromatic*. Well, we can say that the binary string x "is" the coloring χ.)

- For any edge $(u, v) \in E$, the edge contributes a -1 in its $\mathbf{CZ}$ gate if both u and v are colored **B**. Call it a **B-B** edge.
- Therefore, a coloring gives a $\mathbf{-1}$ net contribution if it gives G an odd number of B-B edges.
- The amplitude value $\langle 0^n | U_G | 0^n \rangle$ is positive if fewer than 2^{n-1} (i.e., half) the colorings create an odd number of B-B edges, zero if exactly half do, negative if more.

Whether *one* amplitude is positive or negative does not matter so much in quantum up to equivalence under scalar multiplication. (My lecture demo'ed some examples.) But patterns of signs between different amplitudes $a_z|z\rangle$ of possible outcomes z may have further significance.

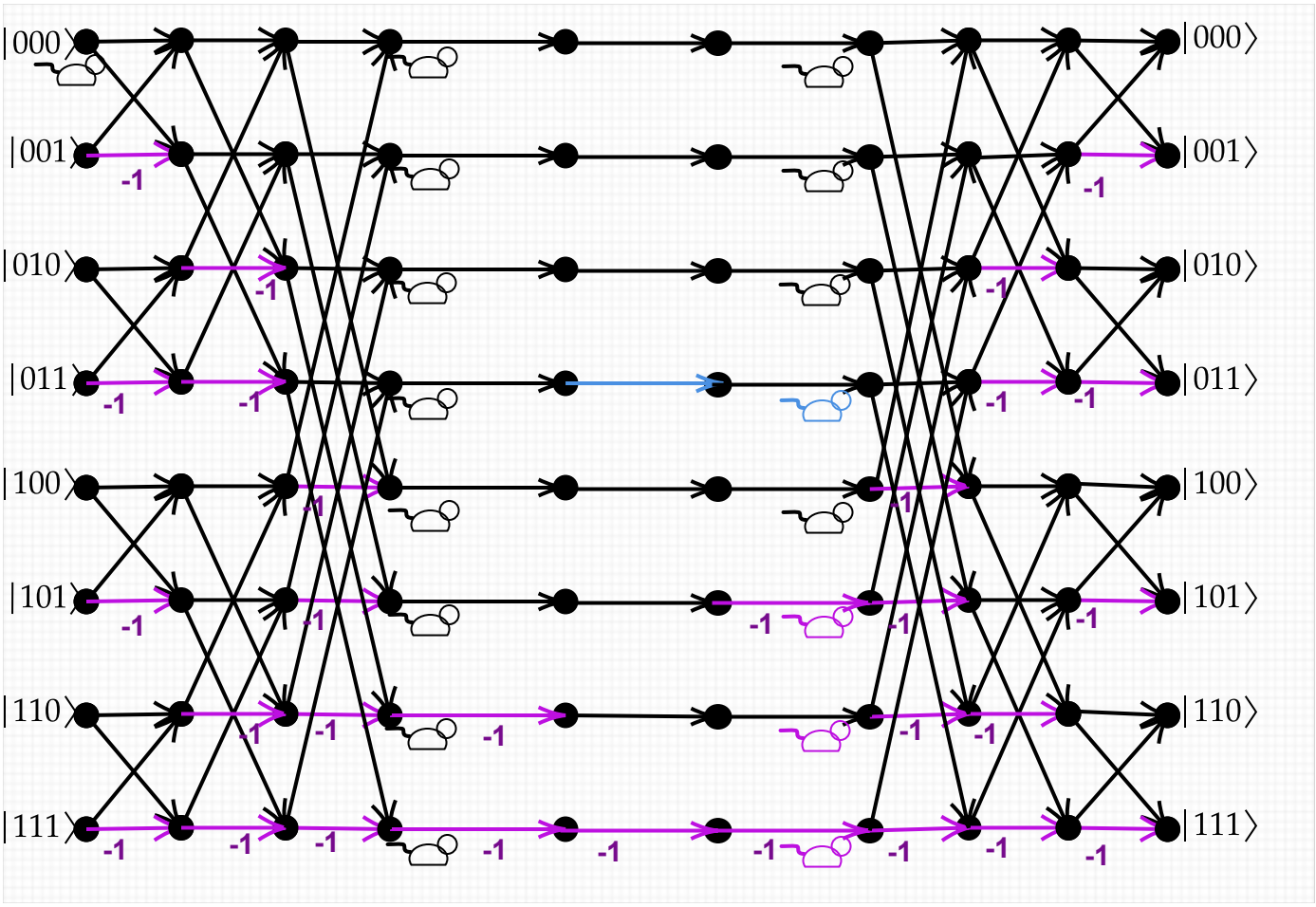
Whether the amplitude is *zero*, however, is absolute. I call a graph G "**net-zero**" if $\langle 0^n | U_G | 0^n \rangle = 0$. Above we first observed that the single-node loop graph is net-zero. The smallest simple undirected graph (meaning no loops or multiple edges) that is net-zero is the triangle. Here are all such graphs up to five vertices:



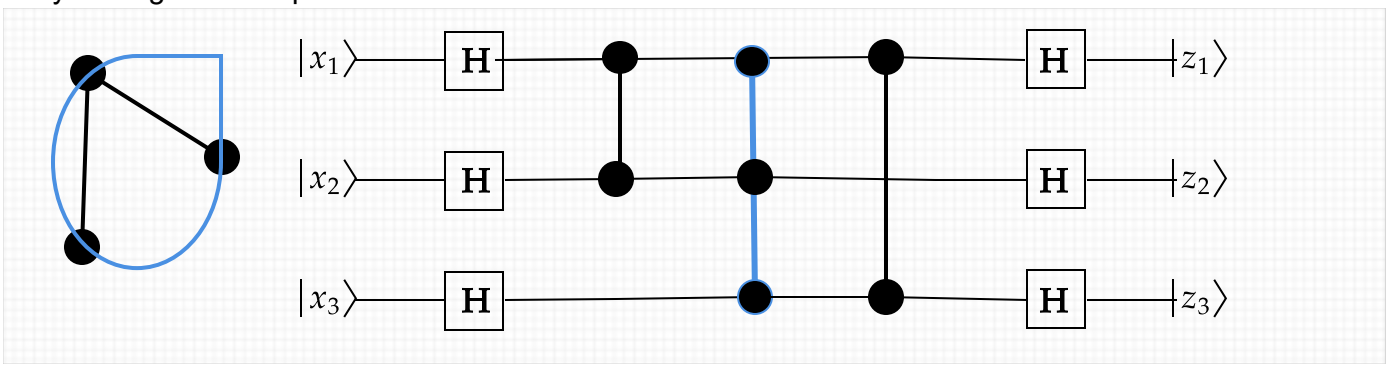
I do not see any simple way to tell "visually" whether a graph is net-zero. My recent PhD graduate Chaowen Guan and I improved the known running time to decide this algorithmically from $O(n^3)$ to whatever the time to multiply two $n \times n$ matrices is (currently $< O(n^{2.37286})$). The algorithm works by converting the graph-state circuit into a quadratic equation of a kind that converts into a linear equation in $O(n)$ variables, whose solutions can be counted in yea-much time. But a simple, more-direct criterion for a graph to be net-zero could give a practically much better algorithm. Guan and I wrote about this on the GLL blog at

https://rjlipton.wpcomstaging.com/2019/06/10/net-zero-graphs/

Some generalizations of graph-state circuits can be handled with equal efficiency. We can simulate $\mathbf{CNOT}$ gates since $\mathbf{CNOT}\ i\ j$ is equivalent to $\mathbf{H}\ j\ \mathbf{CZ}\ i\ j\ \mathbf{H}\ j$. The extra $\mathbf{H}$ gates take things outside the realm of graph-state circuits as strictly defined, but keeps them within the class of so-called **stabilizer circuits**, or equivalently, **Clifford circuits**, to which the same $< O(n^{2.37286})$ runtime applies (for getting any one amplitude, that is). The gates allowed in these circuits are $\mathbf{H}$, $\mathbf{CNOT}$, $\mathbf{S}$, $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$, $\mathbf{CZ}$, but notably not $\mathbf{Tof}$, $\mathbf{T}$, or $\mathbf{CS}$. Or $\mathbf{CCZ}$ for that matter. But there are other tweaks that seem to be easy to bring within our framework, yet yield hard problems. Consider:



The only change was in the middle column, removing the -1 from the row for $|011\rangle$. The middle column now "fires" only when all 3 bits are 1, i.e., for the component of $|111\rangle$ in any state. This is the action of the double-controlled Z-gate, **CCZ** (which is really a triple control of a 180° phase shift). It is easy to diagram in a quantum circuit:



In graph-theoretic terms, this has replaced the edge $(2, 3)$ by the **hyper-edge** $(1, 2, 3)$, thus creating a **hypergraph**. The effect of changing only the color of the mouse in row 4 (for $|011\rangle$) may seem small, but it has a wild effect on the state vector. Now $z = |000\rangle$ has 5 positive paths from $x = |000\rangle$ instead of 4, so its amplitude is $\frac{5-3}{8} = \frac{1}{4}$. Six other components have amplitude $\frac{1}{4}$, and they collectively have $\frac{7}{16}$ of the probability. The other one, for $|100\rangle$, has 7 positive paths to 1 negative, and so amplitude $\frac{7-1}{8} = \frac{3}{4}$ which squares to $\frac{9}{16}$. Note that the previous amplitude was $\frac{6-2}{8} = \frac{1}{2}$ which squares to just $\frac{1}{4}$,

so flipping just one path of eight made a $\frac{5}{16}$ difference to the probability, more than one might expect. The CCZ gate could likewise be in any order---the gates commute so there is no element of time sequencing until the final bank of H gates. The middle part is "instantaneous."

This little illustration of wildness sits over a more general point. The equation resulting from having the CCZ gate changes from quadratic to cubic. Counting solutions to this kind of cubic equation is **NP-hard**. In fact, sandwiching the CCZ gate between two H gates (on any one qubit line) gives the Toffoli gate (with target on that line). So CCZ goes outside the Clifford ambit and gives a universal gate set.

What About Measurement?

Let's say we measure qubit 1 (big-endian). There is a 1/4 chance of getting the result 0 and 3/4 chance of getting 1. If we measured all the qubits, we would see a 9/16 chance of getting 100, 1/16 each for 101, 110, and 111. But when we measure just one qubit, the rest of the state stays superposed. Which part is "the rest of the state" depends on the outcome of the measurement. In this case:

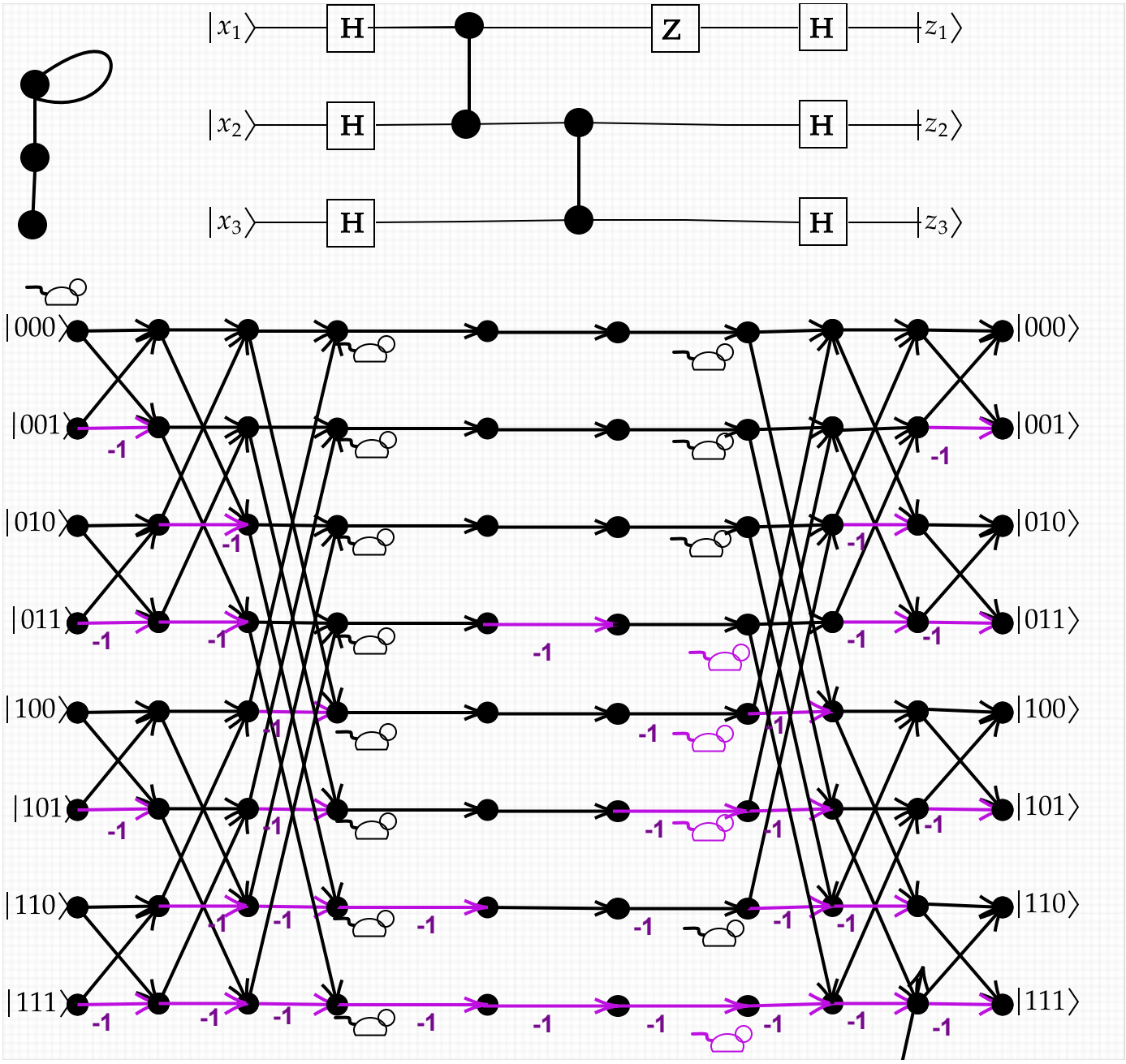
- If the outcome is 0, the new state on qubits 2 and 3 is $\frac{1}{2}[1,1,1,1]^T$. Equal weight superposition with positive signs
- If the outcome is 1, then preserving the relative amplitudes the gives $[3,-1,-1,-1]^T$. (Or $[-3,1,1,1]^T$, which has the same ratios of amplitudes .) To renormalize this, divide by the square root of 12, which is twice the square root of 3. The state also equals $\frac{1}{\sqrt{3}}[1.5,0.5,0.5,0.5]^T$.

Heres's a **challenge** : Can we get this state using just the graph-state gates on two qubits? We will also allow you CNOT and Pauli X and Y and even the **phase gate** S, but not T or CS . And not CCZ or Toffoli since only two qubits *without ancillae*. If not, can we prove not?

There is a more exact rule for computing the new state, predicated on the result of the measurement. Since we have adopted the principle of deferred measurement, we can defer it to chapter 14 in November . But we can see the results in *Quirk* by applying its postselection operators. Note that they are outerproducts.

In any event, this shows special effects one can do with non-Clifford gates like CCZ .

Another Graph State Circuit Example:


$|x_1\rangle$
$|x_2\rangle$
$|x_3\rangle$
H
Z
H
$|z_1\rangle$
$|z_2\rangle$
$|z_3\rangle$
|000⟩
|001⟩
|010⟩
|011⟩
|100⟩
|101⟩
|110⟩
|111⟩
-1