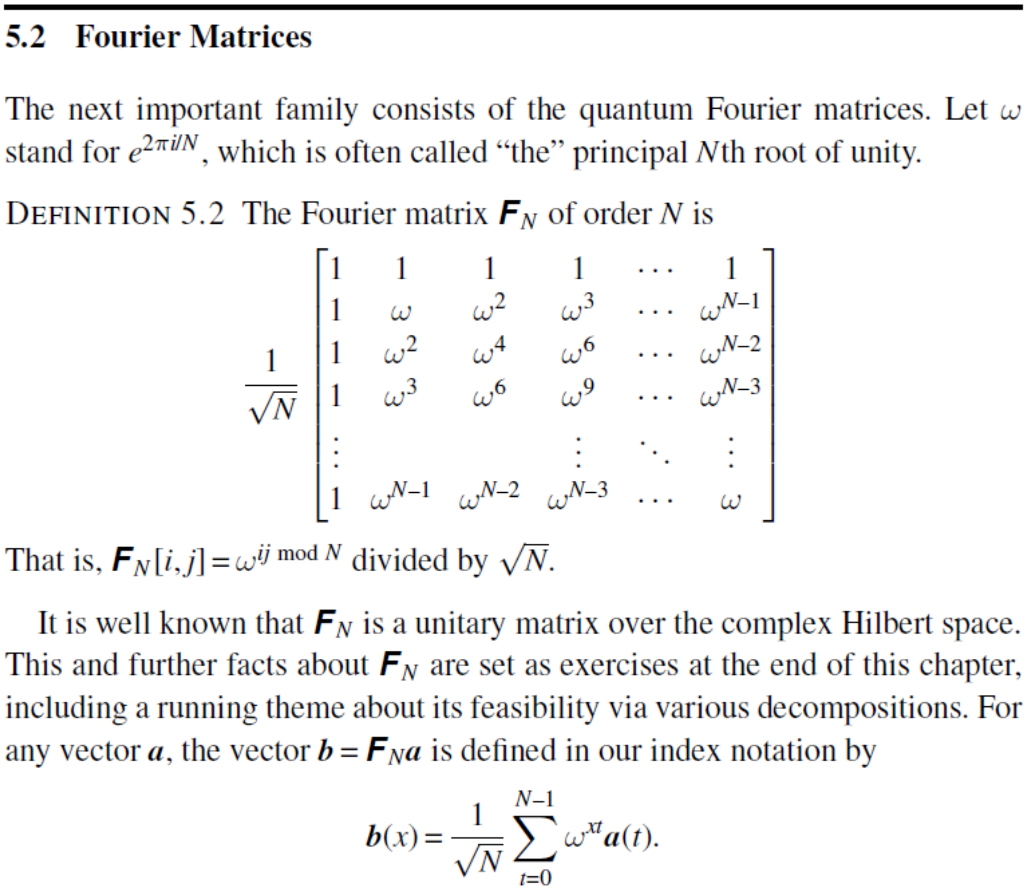
CSE610 Week 5: Quantum Synthesis

Having covered quantum gates and more about the underlying physics, we now start the design of quantum circuits for larger-scale purposes.

The main new piece is the Quantum Fourier Transform, which is just the Discrete Fourier Transform with exponential scaling:



5.2 Fourier Matrices

The next important family consists of the quantum Fourier matrices. Let ω stand for $e^{2\pi i/N}$, which is often called “the” principal Nth root of unity.

DEFINITION 5.2 The Fourier matrix $\boldsymbol{F}_N$ of order N is

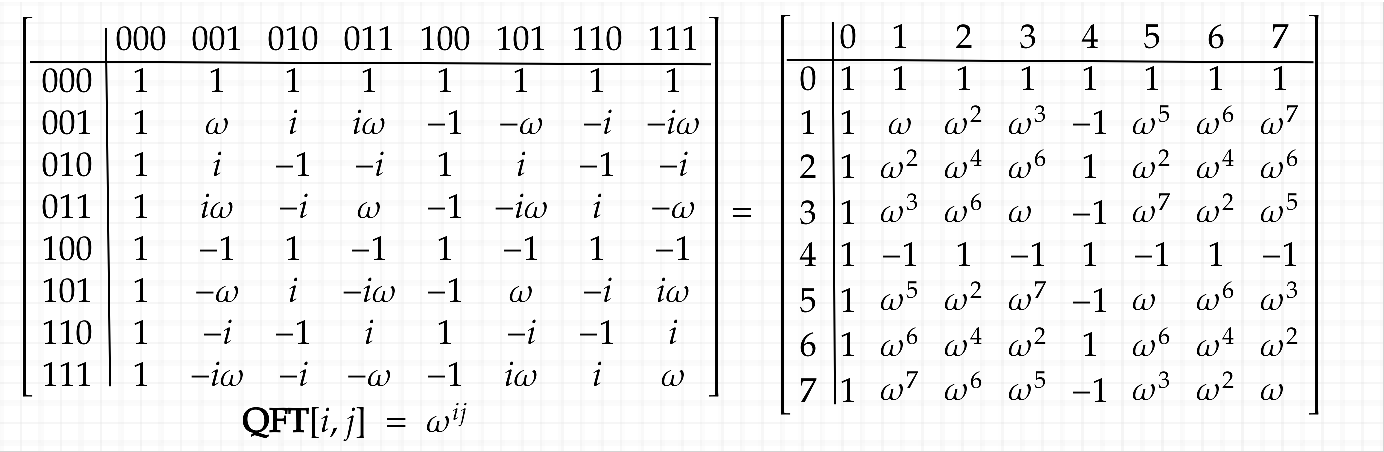
$$\frac{1}{\sqrt{N}}\begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{N-2} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{N-3} \\ \vdots & & & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{N-2} & \omega^{N-3} & \cdots & \omega \end{bmatrix}$$

That is, $\boldsymbol{F}_N[i,j] = \omega^{ij \bmod N}$ divided by $\sqrt{N}$.

It is well known that $\boldsymbol{F}_N$ is a unitary matrix over the complex Hilbert space. This and further facts about $\boldsymbol{F}_N$ are set as exercises at the end of this chapter, including a running theme about its feasibility via various decompositions. For any vector $\boldsymbol{a}$, the vector $\boldsymbol{b} = \boldsymbol{F}_N\boldsymbol{a}$ is defined in our index notation by

$$\boldsymbol{b}(x) = \frac{1}{\sqrt{N}}\sum_{t=0}^{N-1} \omega^{xt}\boldsymbol{a}(t).$$

For any n, it takes $\omega_n = e^{2\pi i/N}$ where $N = 2^n$. With $n = 3$, the matrix together with its quantum coordinates is:



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

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

Compare-contrast with the Hadamard Transform---here illustrated on 4 qubits:

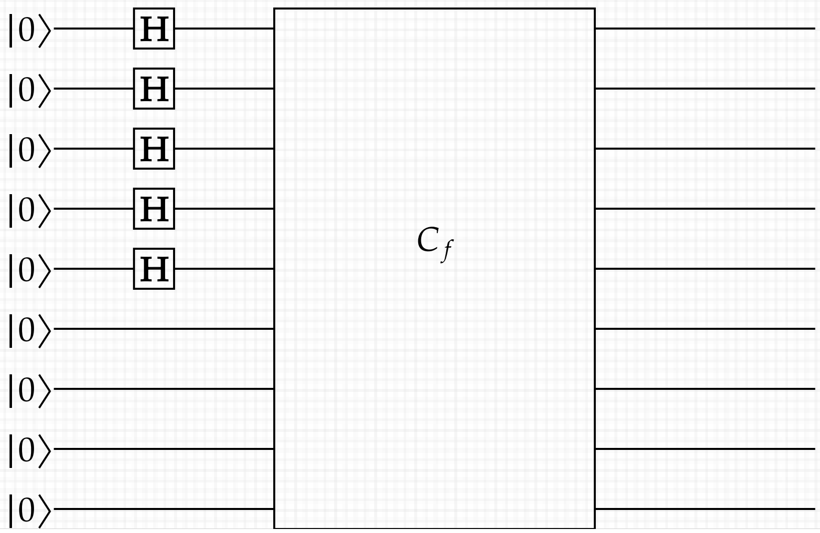
$$\left[\begin{array}{ccccccccccccccccc}
\mathbf{H} & 0000 & 0001 & 0010 & 0011 & 0100 & 0101 & 0110 & 0111 & 1000 & 1001 & 1010 & 1011 & 1100 & 1101 & 1110 & 1111 \\
0000 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0001 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1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

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

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

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

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. This sum has exponentially many terms. It seems to preserve an exponential amount of information: the entire truth table of the Boolean function $f(x)$ over all arguments $x \in \{0,1\}^n$. However:

- f is not an arbitrary or "random" function: it is computed by a small circuit of s NAND gates.
- We cannot actually extract an exponential amount of information from $|\Phi_f\rangle$. If we measure it using the standard basis, we get our argument x back again plus r bits of some sampled function value. Measuring it in a different basis does not increase the information yield (this is part of **Holevo's Theorem**).

Nevertheless, the question remains of whether some exponential amount of "effort" must go in to the creation of $|\Phi_f\rangle$. We will "table" this question and consider the effort to be just $O(n)$ for the Hadamard transform plus $O(s)$ for the circuit.

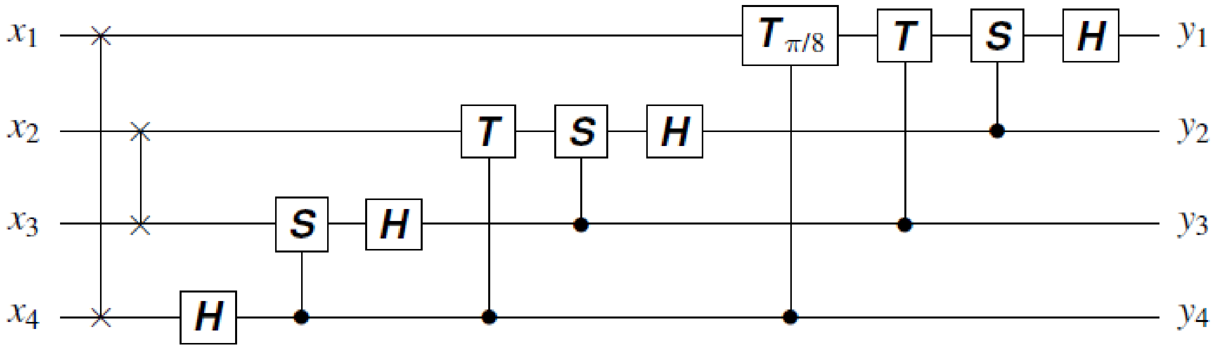
The Fourier transform can produce the same functional superposition, since it gives the same result on the all-$|0\rangle$ initialization. However, its body---which comes into play on other arguments---involves a different exponential element: the fineness of phase angles, starting with taking

$$\omega \;=\; e^{2\pi i/2^n}.$$

The two arguments that justify this are:

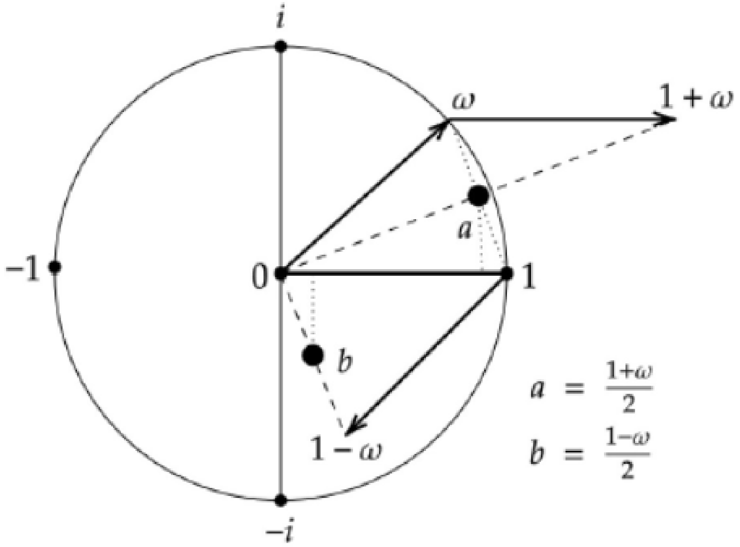
- The n-qubit **quantum Fourier transform** (QFT) can be built up out of $O\left(n^2\right)$ smaller gates.
- Some of those gates are controlled fine-angle rotations (about the $\mathbf{z}$ axis of the Bloch Sphere), but they in turn can be built up from a small basic **universal set** of gates by what I'll call "stretching and halving".

To illustrate the former first, here is how to create the $\mathbf{QFT}$ on four qubits:

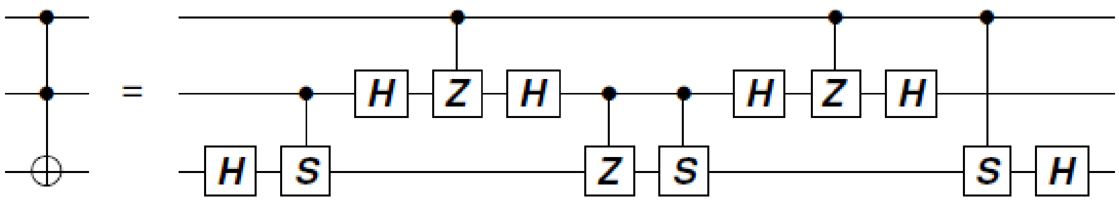


Here $T_{\pi/8} = \begin{bmatrix} 1 & 0 \\ 0 & \omega' \end{bmatrix}$ with $\omega' = e^{i\pi/8}$ not $\omega = e^{i\pi/4}$ as with the T-gate. So ω' has a phase angle one-sixteenth of a circle. For $n = 5$ the next bank uses $1/32$, then $1/64$, and soon the angles would be physically impossible so the gates could never be engineered. Those super-tiny angles are in the definition of the QFT itself.

A prior example already shows what's meant by "stretching and halving":



- Basic gates can fabricate quantum states having finer phases. This is already hinted by the diagram in the case of $\mathbf{HTH}$. Try composing $\mathbf{HTHT^{*}H}$ and $\mathbf{HTHT^{*}HTHT^{*}H}$. The *Solovay-Kitaev theorem* enables approximating operators with exponentially fine angles by polynomially many gates of phases that are multiples of ω (using **CNOT** to extend this to multiple-qubit operators).
- A supplementary point is that the Toffoli and Hadamard gates by themselves, which have phases only $+1$ and -1, can simulate the real parts and imaginary parts of quantum computations separately via binary code, in a way that allows re-creating all measurement probabilities. (This is undertaken in exercises 7.8--7.14 with a preview in the solved exercise 3.8.)
- The **CNOT** and Hadamard gates do not suffice for this, even when the so-called "phase gate" $\mathbf{S} = \mathbf{T}^2 = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$ is added. The Pauli $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ gates and also $\mathbf{CZ}$ can be built from these, but quantum circuits of these gates can be simulated in deterministic ("classical") polynomial time. However, $\mathbf{CS}$ suffices to build the Toffoli gate, per the diagram below (which is also a presentation option). So Hadamard + $\mathbf{CS}$ is a universal set using only quarter phases.



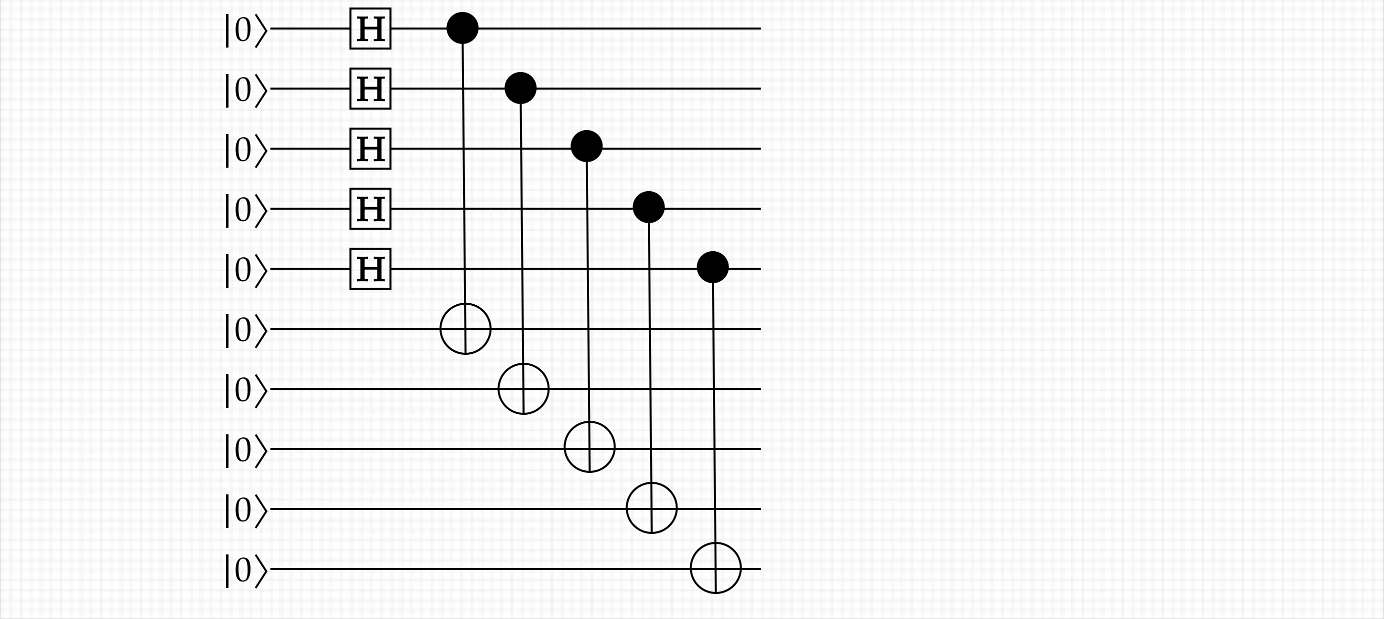
- The ultimate reason may be that the signature application of the QFT, which is *Shor's algorithm* showing that factoring belongs to BQP, may only require coarsed-grained approximations to $\mathbf{QFT}_N$.

For these reasons, $\mathbf{QFT}_N$ is considered feasible even though $N = 2^n$ is exponential. Not every $N \times N$ unitary matrix U is feasible---the Solovay-Kitaev theorem relies on U having a small exact formulation to begin with. But if we fix a finite **universal gate set** (such as $\mathbf{H+T+CNOT}$, $\mathbf{H+Tof}$, or $\mathbf{H+CS}$ above) and use only matrices that are compositions and tensor products of these gates, then we can use the simple gate-counting metric as the main complexity measure.

[The Tue. 9/28 lecture ended by demo'ing the above circuits on both Davy Wybiral's Quantum Circuit Simulator and the richer-but-majorly-busier simulator Quirk. The latter illustrates Bloch spheres both for qubits and for single-qubit operators, and also shows mixed states as being local "handles" of entangled states. We will pick the latter theme up in section 14.6 after first covering up through the applications in Chapter 8 with "Alice" and "Bob" in communication.]

Note About Functional Superpositions (cf. sections 6.2 and 6.4)

We've seen (on homework) that when f is the Boolean identity function on $n = 1$ bit, then C_f consists of just one $\mathbf{CNOT}$ gate. This generalizes for $n > 1$ using one $\mathbf{CNOT}$ gate per argument. Thus



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 sdaw 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 $D\big(|x\rangle\big)$ have the single term $|xx\rangle$ with coefficient $1 = 1^2$.

The Copy-Uncompute Trick (section 6.3)

The Deferred Measurement Principle (section 6.6)

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

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

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.

Theorem: 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.

The Phase Flip Trick (section 6.5)

Reckoning and Visualizing Circuits and Measurements (chapter 7)

There are basically three ways to "reckon" a quantum circuit computation:

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

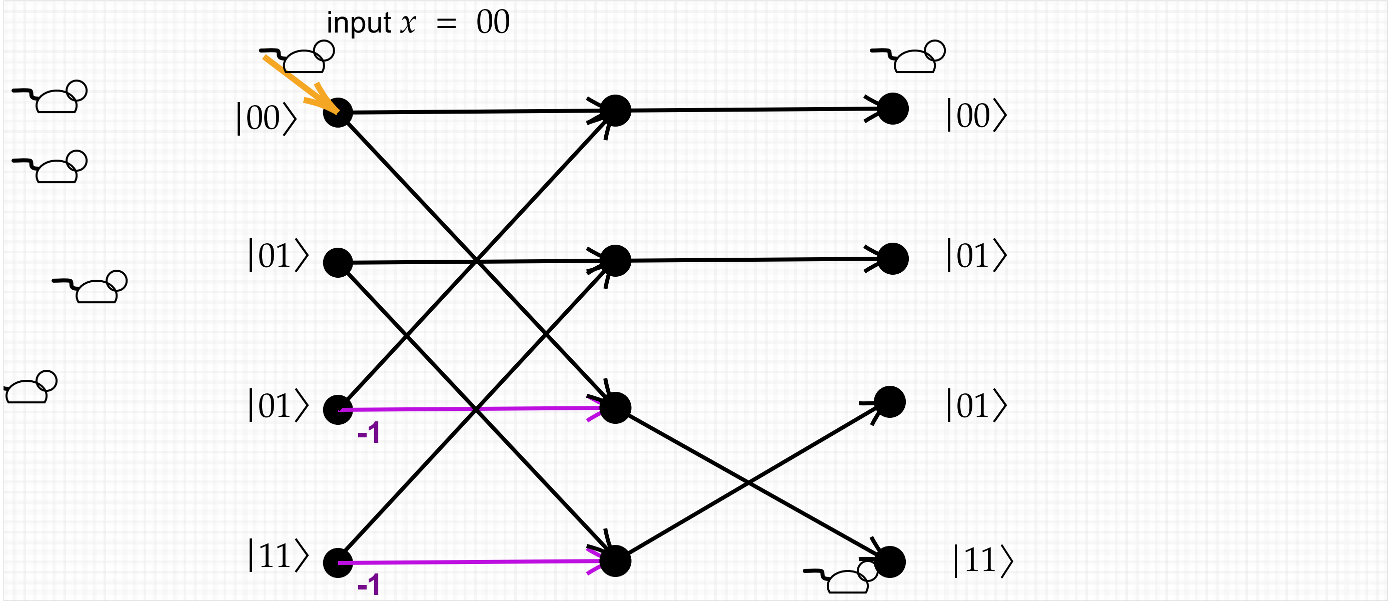
$$ABCDu[i] \ = \sum_{j,k,l,m=1}^{Q} A[i,j] \cdot B[j,k] \cdot C[k,l] \cdot D[l,m] \cdot u[m]\ .$$

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.

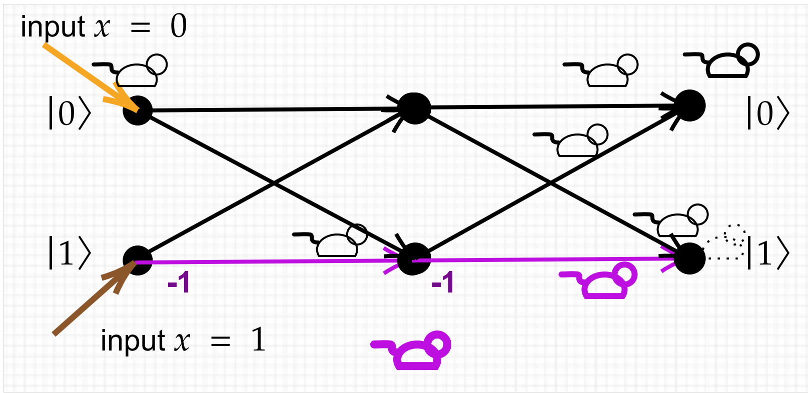
3. Find a way to formulate the matrix product so that the answer comes out of symbolic linear algebra---if possible!

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 $\mathbf{H} + \mathbf{CNOT}$ entangling circuit example:



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:



$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{array}{c} \begin{array}{cc} |0\rangle & |1\rangle \end{array} \\ \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \end{array}$$

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.