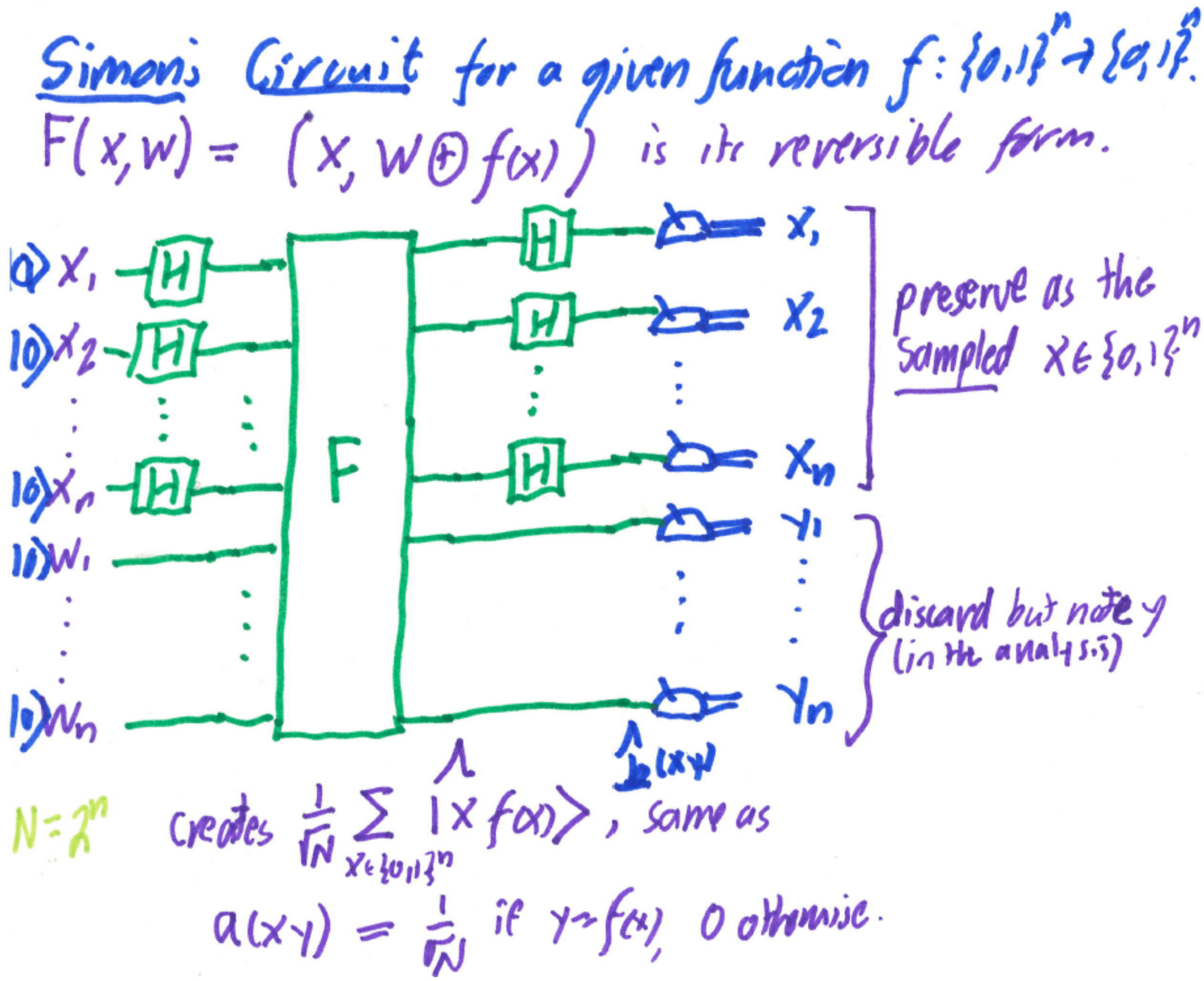
CSE491/596 Lecture 12/08/23: Simon's Algorithm, Concluded

Lecture began with pen on paper sketching the whole circuit:



To do the analysis, we represent the quantum state $\mathbf{b}$ just before the measurement, to see which possible outcomes $|xy\rangle$ from the measurement have nonzero amplitude. If y is not in the range of f, then the terms $\mathbf{a}(ty)$ in the expression for the amplitude $\mathbf{b}(xy)$ are all zero, so $\mathbf{b}(xy)$ is zero regardless of x. For y in the range of f, in the case where f is 2-to-1, this means there are unique strings z_1 and z_2 such that: $f(z_1) = f(z_2) = y$ and $z_1 \oplus z_2 = s$. Then the body of $\mathbf{b}(xy)$ simplifies as shown below:

Input $a(xy) = \frac{1}{\sqrt{N}}$ if $y = f(x)$, 0 otherwise.

Output $b(xy) = \frac{1}{\sqrt{N}} \sum_{t \in \{0,1\}^n} (-1)^{x \bullet t} a(ty)$

Now suppose f is 2-1 with "period" $s \neq 0$. Given $y \in Ran(f)$, take z_1, z_2 s.t. $f(z_1) = y = f(z_2)$ where $z_1 = z_2 \oplus s$

If $z_1 = z_2 \oplus s$ give y

$$b(x,y) = \frac{1}{N}\left((-1)^{x \bullet z_1} + (-1)^{x \bullet z_2}\right) = \frac{1}{N}\left((-1)^{x \bullet z_1} + (-1)^{x \bullet (z_1 \oplus s)}\right)$$

$$= \frac{1}{N}(-1)^{x \bullet z_1}\left(1 + (-1)^{x \bullet s}\right)$$

zero if $x \bullet s = 1$, else $\pm\frac{2}{N}$ where $x \bullet s = 0$

∴ The sampled x ALWAYS gives $x \bullet s = 0$.

This is either 0 or $\pm\frac{2}{N}$, depending on whether or not $x \bullet s = 0$. Thus, in this case, we have

$$\boldsymbol{b}(xy) = \begin{cases} \pm\frac{2}{N} & \text{if } y \in R \text{ and } x \bullet s = 0; \\ 0 & \text{otherwise.} \end{cases}$$

The case where $\boldsymbol{b}(xy)$ is nonzero occurs exactly for half the xs and for half the ys. This is as it should be—otherwise, the norm of $\boldsymbol{b}$ would not be 1.

Finally, it follows that any measurement yields xy with x a random Boolean string, so $x \bullet s = 0$ as claimed. □

Proof of Theorem 10.1. By lemma 10.2, we accumulate random x so that $x \cdot s = 0$. Because a random vector avoids even an $(n-1)$-dimensional subspace with probability at least $\frac{1}{2}$, the expected number of trials to obtain a full-rank system is below $2n$, and the probability of eventual success is overwhelming. If we are in the $s = 0$ case, then we will quickly find that out as well. The last step, on solving for a nonzero s, is to generate and verify the witness for f not being one to one. □

Note, incidentally, that the classical part of the algorithm gives a vector-space structure to $\{0,1\}^n$, with bitwise XOR serving as vector addition modulo 2. This contrasts with the quantum part of the algorithm using N-dimensional space for its own reckonings.

LEMMA 10.2 Suppose that f is periodic with nonzero s. Then the measured xs are random Boolean strings in $\{0, 1\}^n$ such that $x \bullet s = 0$.

Proof. In this case, f is two to one. Define R to be the set of y such that there is an x with $f(x) = y$; that is, R is the range of the function f. Note that R contains exactly half of the possible y values.

If y is not in R, then $\boldsymbol{b}(xy) = 0$, because no t makes $\boldsymbol{a}(ty)$ nonzero. If y is in R, then there are two values z_1 and z_2 so that $f(z_i) = y$ for each i. Further,

$$z_1 = z_2 \oplus s.$$

In this case,

$$\begin{aligned}\boldsymbol{b}(xy) &= \frac{1}{N}\left((-1)^{x \bullet z_1} + (-1)^{x \bullet (z_1 \oplus s)}\right) \\ &= \frac{1}{N}(-1)^{x \bullet z_1}\left(1 + (-1)^{x \bullet s}\right).\end{aligned}$$

To finish the whole argument: If f is 1-to-1, so that $s = 0^n$, then every x makes $x \bullet s = 0$. The analysis kicks in to say that if we currently have at most $n - 1$ linearly independent equations, then with at least 50-50 probability we get one more from the measurement, which gives a random vector $x \in \{0, 1\}^n$. Once we know we have n linearly independent equations---which we can tell in deterministic polynomial time by Gaussian elimination---then we know we must be in the 1-to-1 case. The only possible error is if we kepy on unluckily getting "tails" meaning a dependent equation.

If f is 2-to-1, then we will never get n independent equations. We want to get $n - 1$ of them, so that we can deterministically solve for s uniquely. By similar reasoning, the worst case is when one has $n - 2$ independent equations, whereupon the chance of getting a new one from re-running the circuit and re-sampling the measurement is 50-50. Doing $3n$ or so trials gives only an exponentially small chance of never getting the $(n - 1)$st equation. And when you get it, there is only an exponentially small chace of being wrongly stuck on $(n - 1)$ when the truth is f being 1-to-1. Thus, with high likelihood, you will efficiently reach the answer ``2-to-1'' in this case---and compute s as well.

The final plank in Simon's theorem is that a *classical* polynomial-time randomized algorithm cannot achieve anywhere near the same level of confidence in the answer. This is rigorously proved when the algorithm is only allowed to query the function f in its Boolean form. If f is given as a numerical function (such as under the reductions to polynomial and linear functions on assignments 4 and 5), then classical impossibility is unclear. This is the import of my article

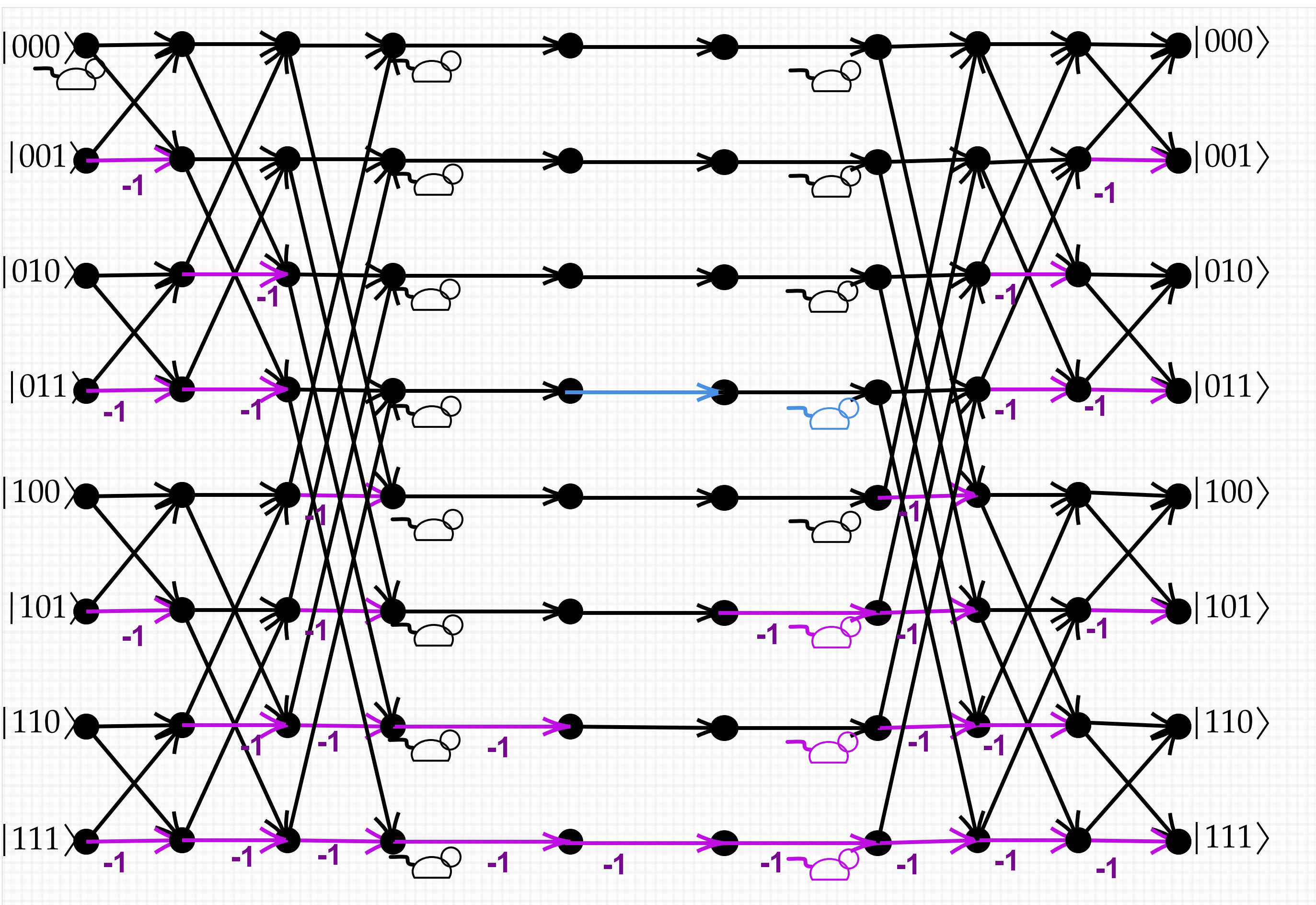
https://rjlipton.wpcomstaging.com/2011/11/14/more-quantum-chocolate-boxes/

from November 2011. This objection notwithstanding, Peter Shor was inspired by Simon's algorithm to find an efficient quantum algorithm for a standard (i.e., non-oracle, non-learning) problem that much of humanity believes in---and depends on---its not being efficiently classicially solvable. This problem is our old friend **factoring**, whose decision version we saw belongs to $\mathrm{NP} \cap \mathrm{co\text{-}NP}$.

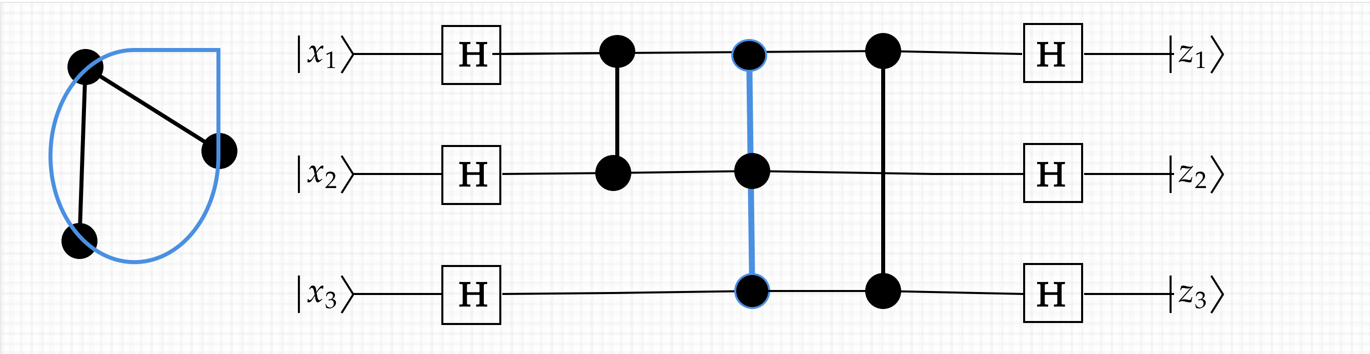
[Lecture then took Qs on homeworks. I mainly wanted to emphasize the similarity of the use of front-and-back Hadamard transform between *graph states* and the week's application examples. The main difference is that the graph states have only the initial n qubit lines for their vertices, whereas the applications have 1 or n more qubits beneath. I closed by inserting the following snippet---here with a better diagram than the one I drew in lecture.]

When Graph States Go "Hyper"

Let us revisit the example at the end of the Friday 12/01 lecture of the graph state circuit for the triangle graph on three nodes. Suppose we change it by rubbing out the first -1 from the middle column, which was previously on the arrow shown in blue:



That is, we removed 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 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 $\mathbf{CCZ}$ gate could likewise be in any order---the gates commute so there is no element of time sequencing until the final bank of $\mathbf{H}$ gates. The middle part is "instantaneous."

This little illustration of wildness sits over a more general point. When you translate the action fo the $\mathbf{CCZ}$ from Boolean logic to a numerical equation, you get one that is cubic---just like from general 3SAT on the homeworks. Counting solutions to this kind of cubic equation is NP-**hard**. In fact, sandwiching the $\mathbf{CCZ}$ gate between two $\mathbf{H}$ gates (on any one qubit line) gives the Toffoli gate (with target on that line). So $\mathbf{CCZ}$ likewise gives a universal gate set. There is a general theorem:

Gottesmann-Knill Theorem: There is a deterministic polynomial-time classical algorithm that, given any n-qubit quantum circuit C composed of the gates $\mathbf{H}$, $\mathbf{CNOT}$, $\mathbf{S}$, $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$, and $\mathbf{CZ}$ only, computes $\langle 0^n | C | 0^n \rangle$ exactly in $s^{O(1)}$ time, where $s \geq n$ is the number of gates in C.

As soon as we add $\mathbf{Tof}$, $\mathbf{T}$, or $\mathbf{CS}$, the theorem goes away---and we have to deal with the full power of quantum circuits. That this power goes beyond classical randomized algorithms is argued by **Shor's Theorem**, to come next.