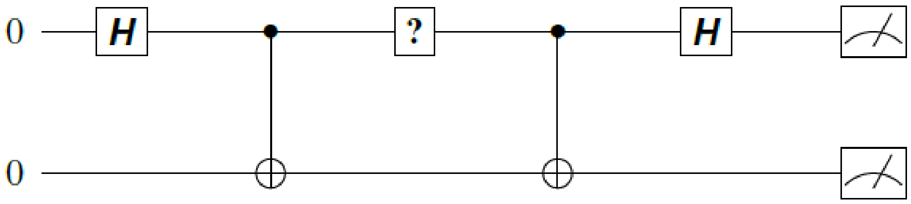
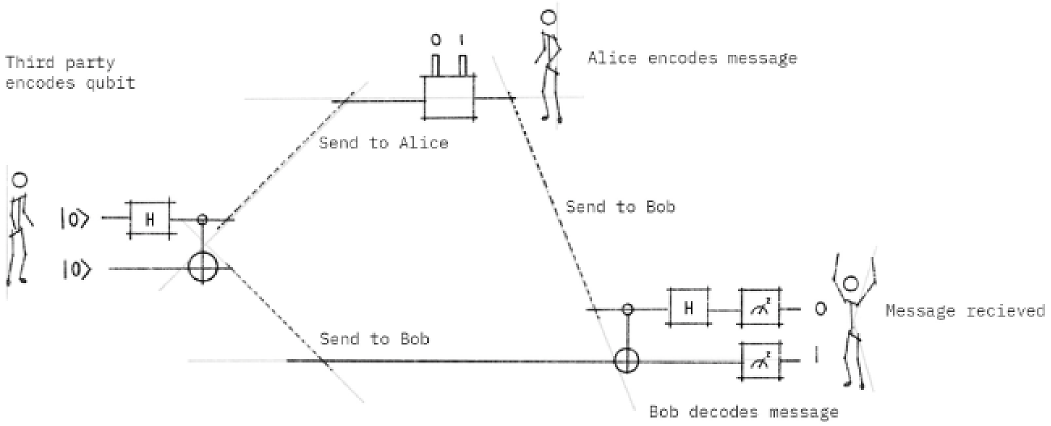
CSE491/596 Lecture Wed. 12/06/32: Quantum Applications, Continued

First, let's revisit the "superdense coding" application. Here is the circuit again in simplest form:



[Added to notes:] Here is the setting as described by IBM's **Qiskit** webpage:



They picture the input $|00\rangle$ coming from a third party, Charlie, who creates an entangled pair via the familiar Hadamard+CNOT method and sends one qubit to each of Alice and Bob. Alice then encodes a 2-bit message by applying one of the four Pauli matrices to her qubit, which is the top line of the quantum circuit. She then sends her *qubit* to Bob, who in this rendition applies CNOT and Hadamard on his own turf.

From Bob's ability to tell exactly which of the four matrices she used, he seems to be getting 2 bits of classical information from the send of *one qubit*. But the catch is that Bob already got a prior qubit from Charlie that, by virtue of entanglement, counts as a prior conduit from Alice. So he got 2 qubits of information from Alice in total after all. One was in the past, before Alice made her elective choice among 4 options, but the conduit stayed in effect after Alice committed her choice. From the standpoint of information, theory, the essence is this:

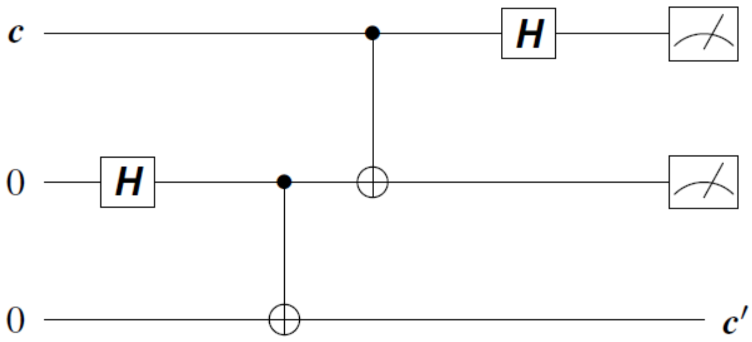
> **The channel between Alice and Bob---which Charlie perched on---carried two qubits of information after all.**

Example: Quantum Teleportation

This one is not a "cheat" but an application with real uses. Here it is conveyed by diagrams from the textbook:

Quantum teleportation involves three qubits, two initially owned by Alice and one by Bob. Alice and Bob share entangled qubits as before, whereas Alice's other qubit is in an arbitrary (pure) state $\boldsymbol{c} = a\boldsymbol{e}_0 + b\boldsymbol{e}_1$. Alice has no knowledge of this state and hence cannot tell Bob how to prepare it, yet entirely by means of operations on her side of the lake she can ensure that Bob can possess a qubit in the identical state.

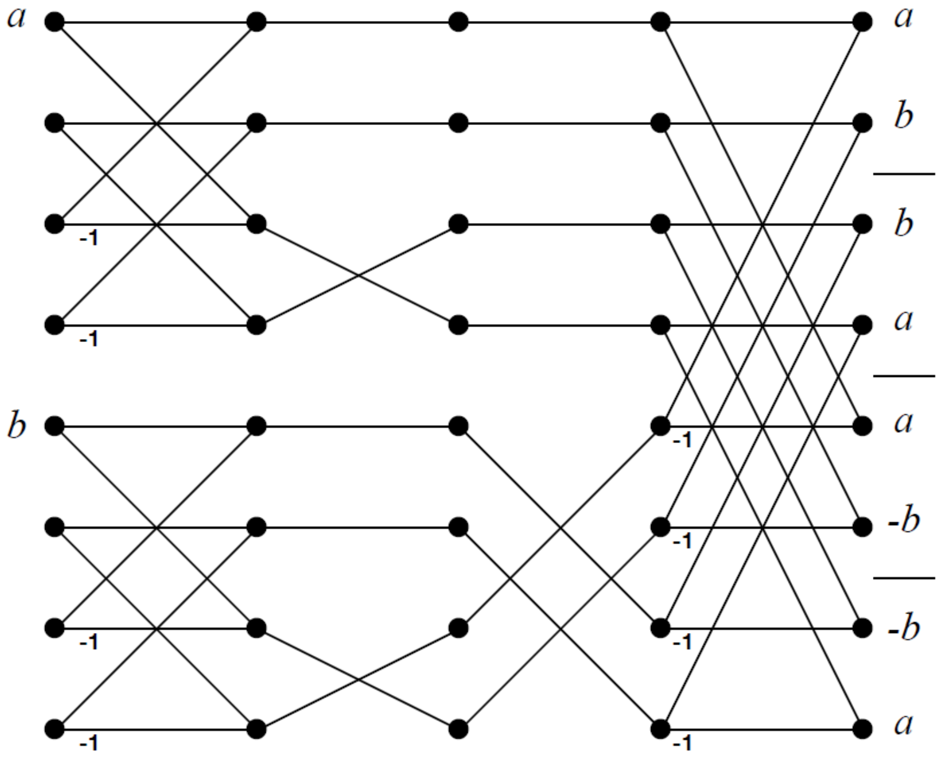
The following quantum circuit shows the operations, with $\boldsymbol{c}$ in the first quantum coordinate, Alice's entangled qubit second, and Bob's last. The circuit includes the Hadamard and ***CNOT*** gates used to entangle the latter two qubits.



With this indexing, the start state is $\boldsymbol{c} \otimes \boldsymbol{e}_{00}$, which equals $a\boldsymbol{e}_{000} + b\boldsymbol{e}_{100}$. After the first two gates, the state is

$$\boldsymbol{c} \otimes \frac{1}{\sqrt{2}} (\boldsymbol{e}_{00} + \boldsymbol{e}_{11}),$$

with Alice still in possession of the first coordinate of the entangled basis vectors. The point is that the rest of the circuit involves operations by Alice alone, including the measurements, all done on her side of the lake. This is different from using a two-qubit swap gate to switch the $\boldsymbol{c}$ part to Bob, which would cross the lake. No quantum interference is involved, so a maze diagram helps visualize the results even with "arbitrary-phase Phils" lined up at the first and fifth rows shown in figure 8.5, which are the entrances for $\boldsymbol{e}_{000}$ and $\boldsymbol{e}_{100}$.



> Because Bob's qubit is the rightmost index, the measurement of Alice's two qubits selects one of the four pairs of values divided off by the bars at the right. Each pair superposes to yield the value of Bob's qubit *after* the two measurements "collapse" Alice's part of the system. The final step is that Alice sends two *classical* bits across the lake to tell Bob what results she got, that is, which quadrant was selected by nature. The rest is in some sense the inverse of Alice's step in the superdense coding: Bob uses the two bits to select one of the Pauli operations $\boldsymbol{I}, \boldsymbol{X}, \boldsymbol{Z}, i\boldsymbol{Y}$, respectively, and applies it to his qubit $\boldsymbol{c}'$ to restore it to Alice's original value $\boldsymbol{c}$.

The point here is not that the two bits sent by Alice were at staggered time intervals, but rather than the quantum state **c** is exactly replicated on Bob's turf. This is without his (or Alice's) knowing what that state is. A ***known*** quantum state can always be re-**prepared**; moreover, the standard basis states can be duplicated using CNOT gates. An arbitrary quantum state **c** cannot be copied, however. That is to say, there is no unitary operation U such that for all quantum states **c**,

$$U(\mathbf{c} \otimes \mathbf{e}_0) = \mathbf{c} \otimes \mathbf{c}.$$

Or in Dirac notation, one cannot do $U|\mathbf{c}0\rangle = |\mathbf{cc}\rangle$. Although there unitary 4×4 matrices that do this when **c** is $|0\rangle$ or $|1\rangle$, there is none that works in general. This is the **no-cloning theorem**, which si covered in section 6.2.

Deutsch-Jozsa Extension

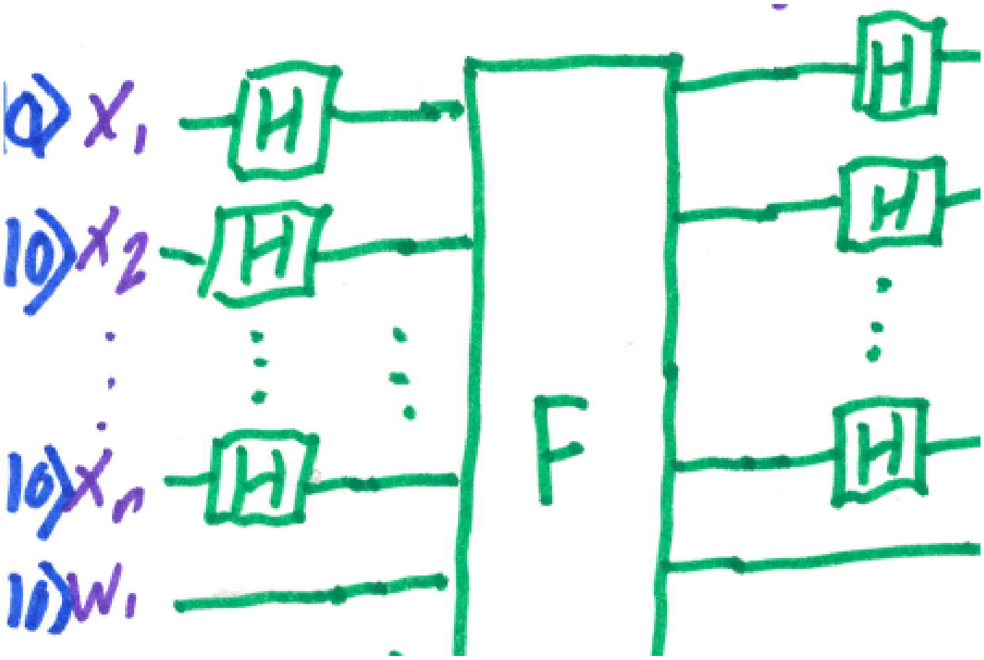
Now we consider problems as the complexity parameter n is scaled up. The maze diagrams would get exponentially big, but we can track the computations via linear algebra.

Call a function $f: \{0,1\}^{n} \rightarrow \{0,1\}$ ***balanced*** if f has 2^{n-1} values of 0 and 2^{n-1} values of 1. When $n = 1$, the problem of distinguishing ***constant*** functions $f: \{0,1\}^{n} \rightarrow \{0,1\}$ from ***balanced*** ones is Deutsch's original problem. Richard Jozsa observed that for higher n, the classical algorithms require $2^{n-1} + 1$ queries, while the quantum ones can still do it on one query to a completely separable superposed state.

This is a conditional problem, called a **promise problem**, in that it only applies when f is in one of those two cases. If f is neither balanced nor constant, then "all bets are off"---any answer is fine, even "$|$ ¯_(ツ)_/¯ $\rangle$". It is like Deutsch's setup except with $\mathbf{H}^{\otimes n}$ in place of the first $\mathbf{H}$, input $|0^n 1\rangle$ in place of $|01\rangle$, and targets (ignoring the $\sqrt{2}$ normalizers):

- constant $\mapsto |0^n\rangle(|0\rangle + |1\rangle)$ (instead of $(|00\rangle + |01\rangle)$, so that 0^n is certainly measured.
- balanced $\mapsto |?\rangle$ (instead of $(|10\rangle + |11\rangle)$, such that 0^n is certainly *not* measured.

Here is a serviceable diagram, intending the final bit to be $w_1 = |1\rangle$. As before, F is the reversible form of f, so that $F(x_1 \cdots x_n w_1) = x_1 \cdots x_n \cdot (w_1 \oplus f(x_1 \cdots x_n))$:



The key observation is that for any f, any argument $x \in \{0,1\}^n$, and $b \in \{0,1\}$, the amplitude in the component xb of the final quantum state ϕ is

$$\frac{1}{\sqrt{2^{n+1}}}\sum_{t \in \{0,1\}^n} (-1)^{x \bullet t}(-1)^{f(t) \oplus b}.$$

Here $x \bullet t$ means taking the dot-products $x_i \cdot t_i$ (which is the same as $x_i \wedge t_i$) and adding them up modulo 2 (which is the same as XOR-ing them). Well, when $x = 0^n$ this is always just zero, so the first term is $(-1)^0$ and just drops out, leaving

$$\phi\left(0^n b\right) = \frac{1}{\sqrt{2^{n+1}}}(-1)^b \sum_{t \in \{0,1\}^n} (-1)^{f(t)}.$$

Note that the $(-1)^b$ term is independent of the sum over t, so it comes out of the sum---and this is why we get two equal possibilities in the original Deutsch's algorithm as well. Thus we have verified the folloing points of the analysis:

- When f is ***constant***, these terms are all the same, so they ***amplify***---giving $\frac{1}{\sqrt{2}}$ for the constant-false function and $\frac{-1}{\sqrt{2}}$ for constant-true. Both of these amplitudes square to $\frac{1}{2}$ and so together soak up all the output probability, so that 0^n is measured with certainty.
- When f is ***balanced***, the big sum has an equal number of $+1$ and -1 terms, so they all ***interfere*** and *cancel*. Hence 0^n will certainly not be measured.

Added: A *randomized* classical algorithm can efficiently tell with high probability whether f is constant by querying some random strings. If it ever gets different answers $f(y) \neq f(y')$ then definitely f is not constant. (So, under the condition of the "promised problem," it must be balanced.) If it always gets the same answer, then since any balanced function gives 50-50 probability on random strings, it can quickly figure that f is constant. But it is still the case that a deterministic algorithm needs exponentially many queries and hence exponential time.

Simon's Algorithm

Here we are given $f : \{0, 1\}^n \to \{0, 1\}^n$ with a promise property. The promise is that there is a "hidden string" $s \in \{0, 1\}^n$ such that for all $x, y \in \{0, 1\}^n$:

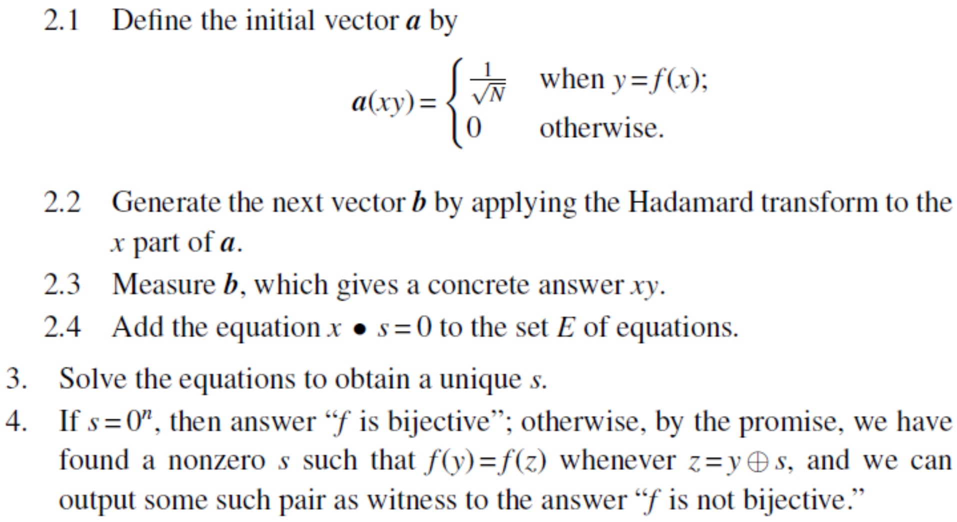
$$f(x) = f(y) \iff y = x \oplus s.$$

When $s = 0^n$, this is equivalent to f being 1-to-1. So the promise is that if f is not 1-to-1, then it is 2-to-1 on $\{0, 1\}^n$ in a very special way controlled by a single string s. For a classical algorithm that is allowed only to query $f(u)$ for $u \in \{0, 1\}^n$ singly, the chance of finding a "colliding" v such that $f(v) = f(u)$ is exponentially tiny. A meta-question is whether "white-box" knowledge of an efficient formula ϕ for f can help find collisions---at least under the "hidden subgroup" promise of there being an s. But in the black-box case, one can actually prove that classical subexponential time has only a vanishing probability of finding a collision.

The "Quantum Black Box" assumption in this case is (IMHO) a little stronger: it is the feasibility of creating the **functional superposition** state

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

In practice, one cannot get this to full precision over exponentially many terms (or can we?) The question shirked by many texts, including mine, is what kind of approximations to states like Φ_f enable the propounded conclusions to go through. Presuming perfection, here is the algorithm:



2.1 Define the initial vector $\boldsymbol{a}$ by

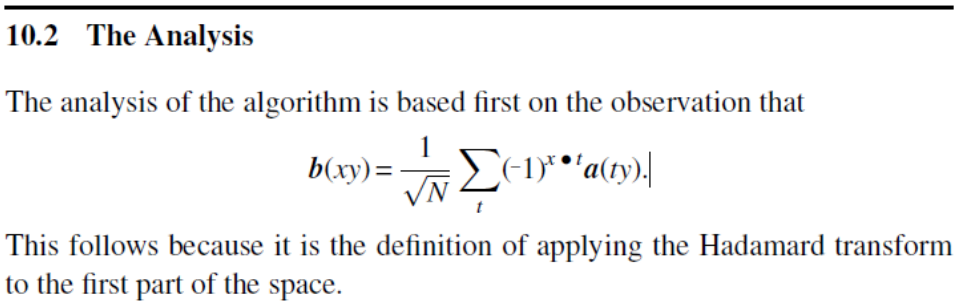
$$\boldsymbol{a}(xy) = \begin{cases} \frac{1}{\sqrt{N}} & \text{when } y = f(x); \\ 0 & \text{otherwise.} \end{cases}$$

2.2 Generate the next vector $\boldsymbol{b}$ by applying the Hadamard transform to the x part of $\boldsymbol{a}$.

2.3 Measure $\boldsymbol{b}$, which gives a concrete answer xy.

2.4 Add the equation $x \bullet s = 0$ to the set E of equations.

3. Solve the equations to obtain a unique s.
4. If $s = 0^n$, then answer "f is bijective"; otherwise, by the promise, we have found a nonzero s such that $f(y) = f(z)$ whenever $z = y \oplus s$, and we can output some such pair as witness to the answer "f is not bijective."



10.2 The Analysis

The analysis of the algorithm is based first on the observation that

$$\boldsymbol{b}(xy) = \frac{1}{\sqrt{N}} \sum_t (-1)^{x \bullet t} \boldsymbol{a}(ty).$$

This follows because it is the definition of applying the Hadamard transform to the first part of the space.

Simon's Theorem: This algorithm---which alternates quantum and classical stages---distinguishes the cases $s = 0^n$ and $s \neq 0^n$ with high probability. Whereas, any sub-exponential time classical probabilistic algorithm has only a negligible advantage over guessing which case holds.

[Lecture carried this over into Friday, with a diagram of the circuit.]