

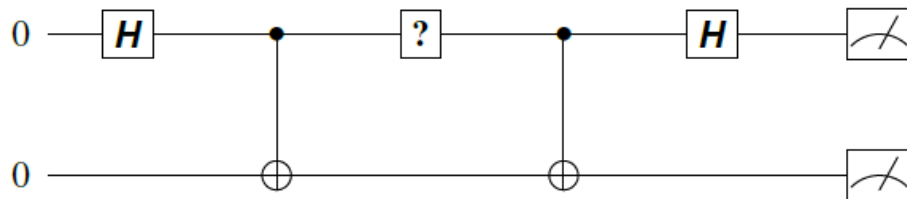
CSE439 Week 7: From Small-Scale to General-Scale Applications

The first two examples are mainly intended as further illustrations of the "maze" visualization.

Superdense Coding

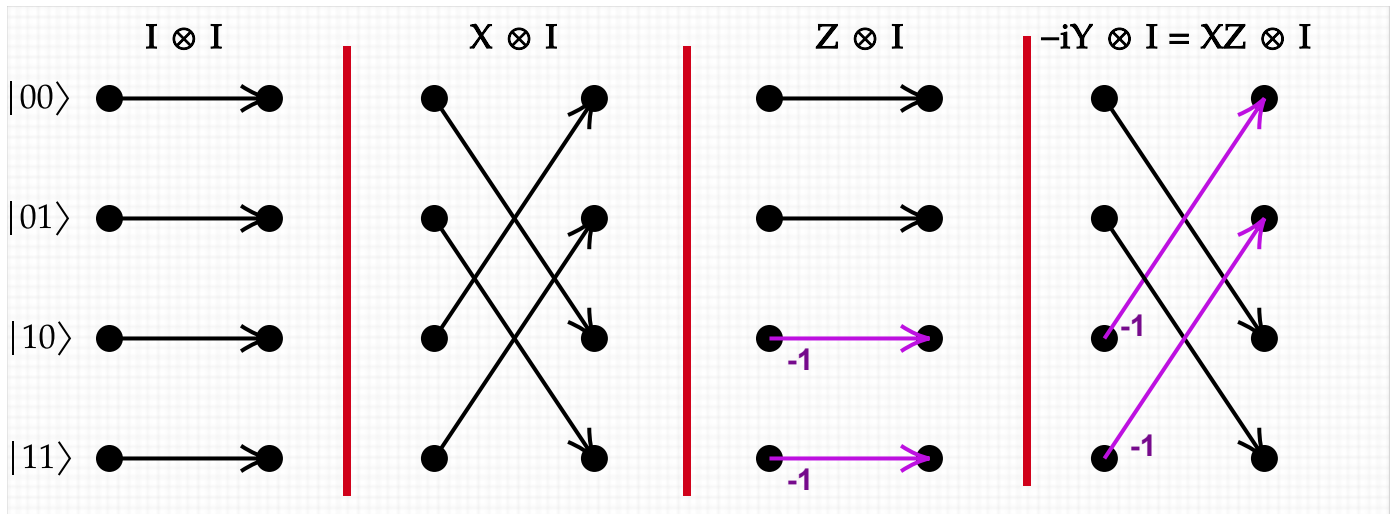
It is easy to rig cases F_0, F_1, F_2, F_3 where you can distinguish them exactly by asking one query and measuring both qubits. Just define $F_i(00) = i$, for instance, where i ranges over $\{00, 01, 10, 11\}$ ---or if you prefer, i ranges over the permutation elements 1, 2, 3, 4 as used above---and have the other values $F(01), F(10), F(11)$ go in cycle after that. See the above diagram for $F(11)$.

"Superdense coding" is a case where the rigging has a bit of surprise because it appears to convey 2 bits of information with just 1 qubit of communication *after* a certain point in time. This ought to be impossible by **Holevo's Theorem**, which we covered previously. The "cheat" in superdense coding is that the communicating parties "Alice" and "Bob" exchange 1 bit of information beforehand in order to set them up with an entangled qubit pair. Here is their circuit:

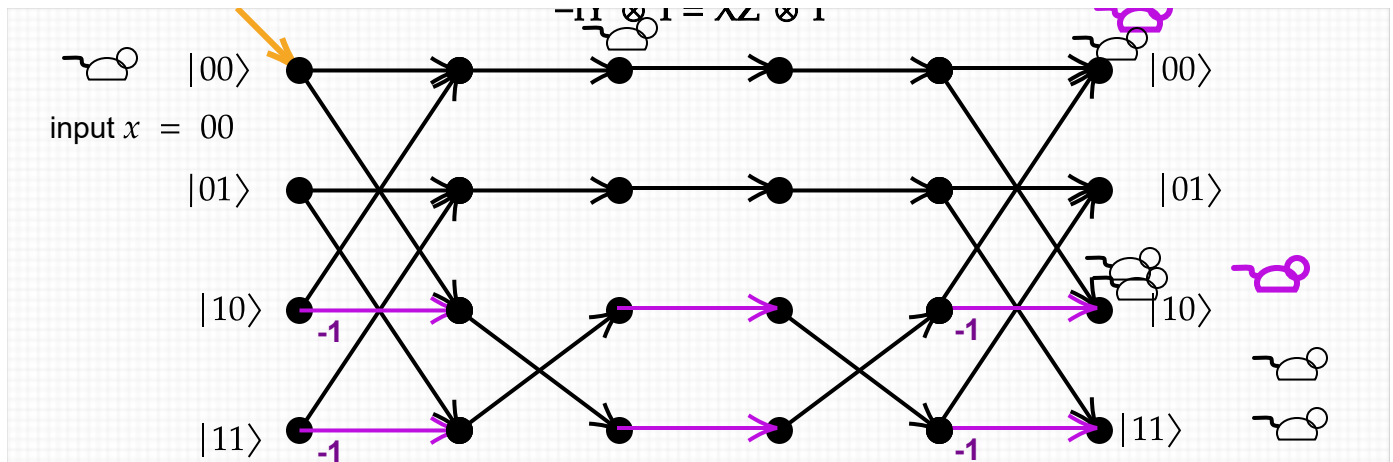


The opening Hadamard and CNOT set up the entangled pair. Alice then chooses one of the four Pauli operators for the unknown operation in the middle. After the second CNOT, she applies Hadamard to her qubit, measures, and sends the result to Bob. Bob then measures his qubit, and is able to infer which of the four operators Alice used. Well, he got a qubit from Alice to begin with, and even though it was before Alice made her 2-bit choice at the "?", it counts as 2 bits of "contact" anyway.

Even after the "magic" is explained away, this remains a nice illustration of a Deutsch-style learning problem using the four Pauli matrices. We want to identify one of the following four possibilities **exactly** by the results of **two** qubits.



This time the input is $|00\rangle$. To work it out via wavefronts (the figure below is left with $XZ \otimes I$ in the middle, but all four will be exemplified):

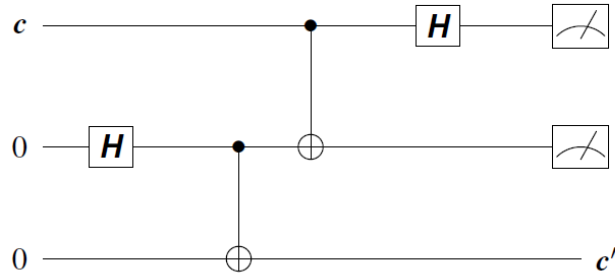


Example: Quantum Teleportation

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 $c = ae_0 + be_1$. 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 c in the first quantum coordinate, Alice's entangled qubit second, and Bob's last. The circuit

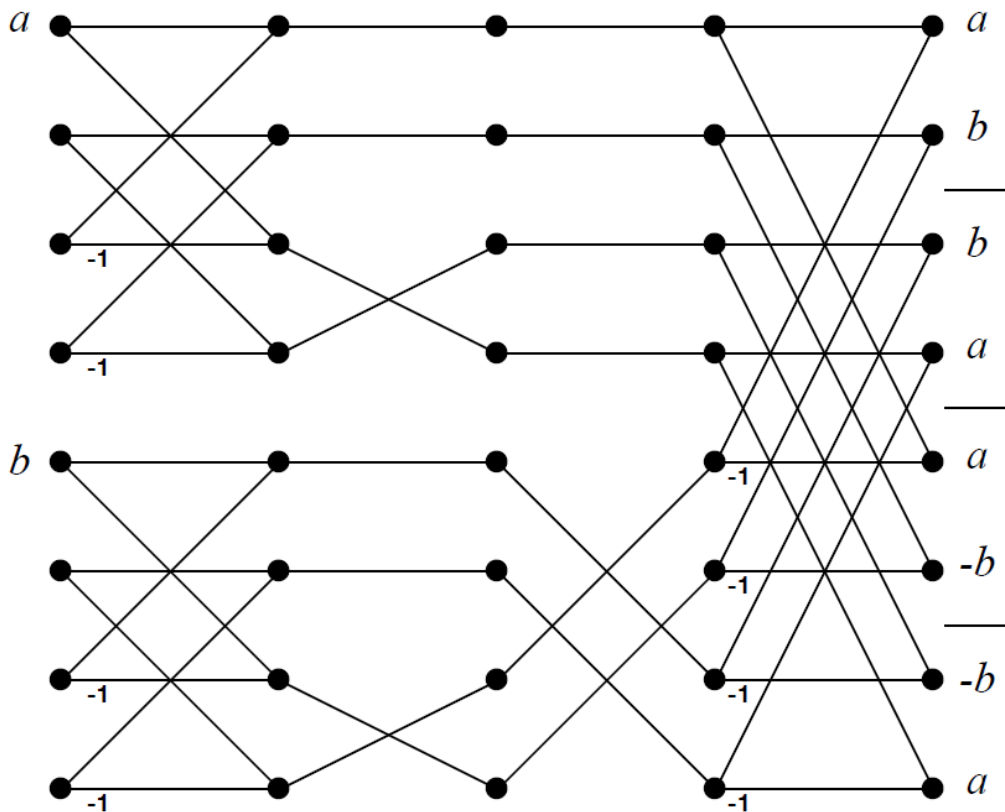
includes the Hadamard and **CNOT** gates used to entangle the latter two qubits.



With this indexing, the start state is $c \otimes e_{00}$, which equals $ae_{000} + be_{100}$. After the first two gates, the state is

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

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 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 e_{000} and 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 I, X, Z, iY , respectively, and applies it to his qubit c' to restore it to Alice's original value c .

[This is the divide between coverage under Prelim I and what comes after. Lecture may take some time to go over homeworks.]

Scaling Up --- And Analyzing With Symbolic Computation

We begin a "vibe shift" from pictorial to formulaic. Note that the pictorial way of tracing the matrix calculations---via Feynman paths---scales exponentially with the number n of qubits. Not only that, it also forks exponentially with the number of Hadamard gates (or square-root-of-NOT or other nondeterministic gates). *Formulas*, however, can scale linearly or at worst polynomially in n .

These chapters also lead into the two major applications where **quantum advantage** is strongly believed: Shor's and Grover's algorithms (to come in chapters 11--13). On the way are the Deutsch-Jozsa Problem---where it is unresolved whether the classical criticism of the original Deutsch's Problem fully applies---and Simon's Problem, where an exponential lower bound on the expected time for any classically randomized algorithm has been proven under reasonable stipulations.

Deutsch-Jozsa Extension (Ch. 9)

Getting back to Deutsch's Problem, Richard Jozsa added that if you only care about distinguishing **constant** functions $f: \{0, 1\}^n \rightarrow \{0, 1\}$ from **balanced** ones, then you can make 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 $|\text{ } \backslash _ (\text{ツ}) _ / \text{ } \rangle$.

The maze diagrams would get exponentially big, but we can track the computations via linear algebra. 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.

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 x^b 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

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

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. The final point is that:

- 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.

At this point, coverage transits to handwritten notes on the blackboard and/or projector:

<https://cse.buffalo.edu/~regan/cse439/CSE439week8notes.pdf>

Note also that week 7 generally includes the First Prelim Exam, while one lecture of week 8 is erased by the short Fall Break. So they are really "one week of lectures."