Week 5: Basic Quantum Algorithms

Michael Silver, ECE 2T6
University of Toronto Quantum Computing Club

Week 3 Recap

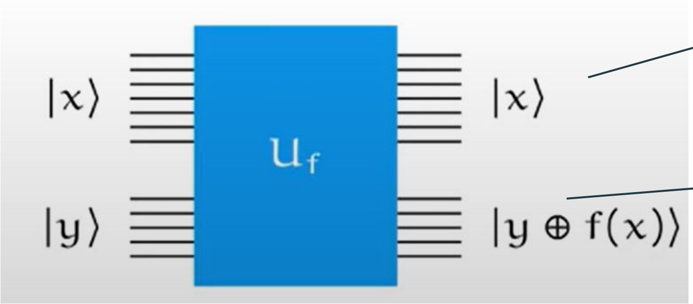
Quantum Oracle and Grover's Algorithm Walkthrough

Query Gates; a Type of Oracle

- Recall: Oracles, black-box operation that encodes a function *f* into a quantum circuit
    - Example: Grover's Algorithm oracle function, f(x) = 1 if x is target, f(x) = 0 otherwise, $U_f|x\rangle = (-1)^{f(x)}|x\rangle$ made state of target $|\psi\rangle \rightarrow -|\psi\rangle$
- **Query Gates**: U_f for any function $f\colon \Sigma^n \rightarrow \Sigma^m$ is defined as

$$U_f(|y\rangle|x\rangle) = |y \oplus f(x)\rangle|x\rangle$$

for all $x \epsilon \Sigma^n$ and $y \epsilon \Sigma^m$



Remains Unchanged

XOR

Deutsch’s Problem

- There are four binary functions of the form f: Σ⟶Σ:

a	$f_1(a)$	a	$f_2(a)$	a	$f_3(a)$	a	$f_4(a)$
0	0	0	0	0	1	0	1
1	0	1	1	1	0	1	1

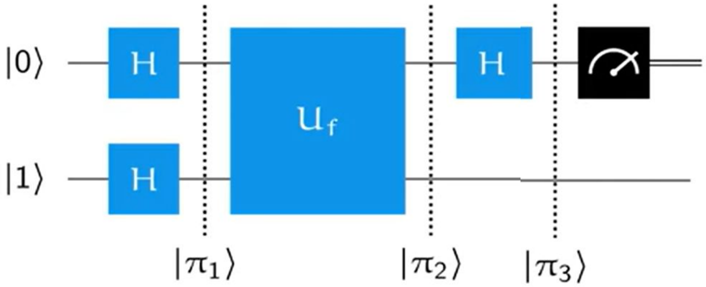
Balanced

Constant

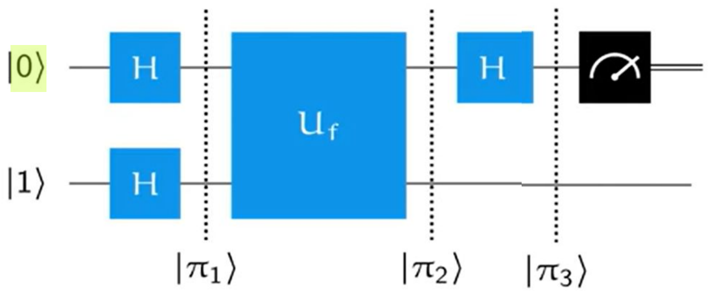
- Output: 0 if f is constant, 1 if f is balanced
- To classically solve: we would need 2 queries to find info about both inputs

Deutsch's Algorithm

- Can solve Deutsch's problem using a single query



- Note: $|\pi_1\rangle, |\pi_2\rangle, |\pi_3\rangle$ represent the **entire system state**, much much easier to perform calculations and think about what's going on that way



$$\begin{cases} 0 & \text{if f is constant,} \\ 1 & \text{if f is balanced} \end{cases}$$

$$|\pi_1\rangle = |-\rangle|+\rangle = \frac{1}{2}(|0\rangle - |1\rangle)|0\rangle + \frac{1}{2}(|0\rangle - |1\rangle)|1\rangle$$

$$\begin{aligned} |\pi_2\rangle &= \frac{1}{2}(|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle)|0\rangle + \frac{1}{2}(|0 \oplus f(1)\rangle - |1 \oplus f(1)\rangle)|1\rangle \\ &= \frac{1}{2}(-1)^{f(0)}(|0\rangle - |1\rangle)|0\rangle + \frac{1}{2}(-1)^{f(1)}(|0\rangle - |1\rangle)|1\rangle \\ &= |-\rangle\left(\frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}}\right) \\ &= (-1)^{f(0)}|-\rangle\left(\frac{|0\rangle + (-1)^{f(0)\oplus f(1)}|1\rangle}{\sqrt{2}}\right) \\ &= \begin{cases} (-1)^{f(0)}|-\rangle|+\rangle & f(0) \oplus f(1) = 0 \\ (-1)^{f(0)}|-\rangle|-\rangle & f(0) \oplus f(1) = 1 \end{cases} \end{aligned}$$

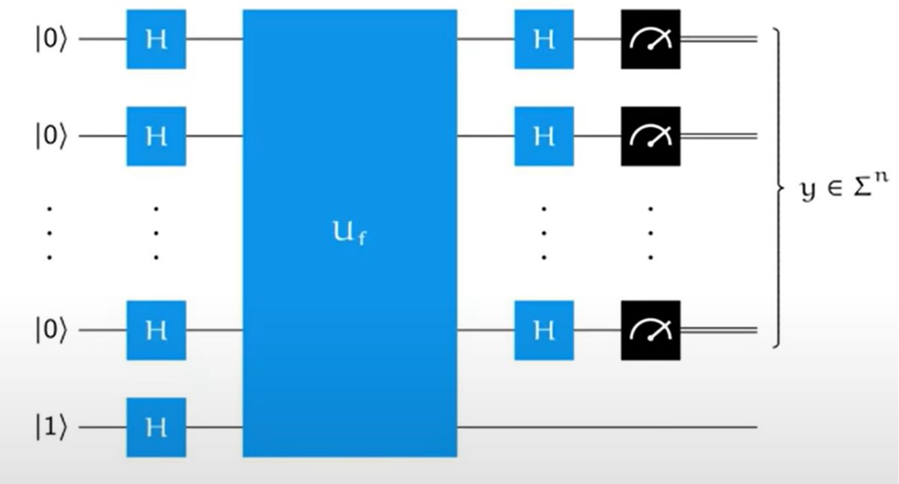
Using formula $|0 \oplus a\rangle - |1 \oplus a\rangle = (-1)^a(|0\rangle - |1\rangle)$

$$\begin{aligned} |\pi_3\rangle &= \begin{cases} (-1)^{f(0)}|-\rangle|0\rangle & f(0) \oplus f(1) = 0 \\ (-1)^{f(0)}|-\rangle|1\rangle & f(0) \oplus f(1) = 1 \end{cases} \\ &= (-1)^{f(0)}|-\rangle|f(0) \oplus f(1)\rangle \end{aligned}$$

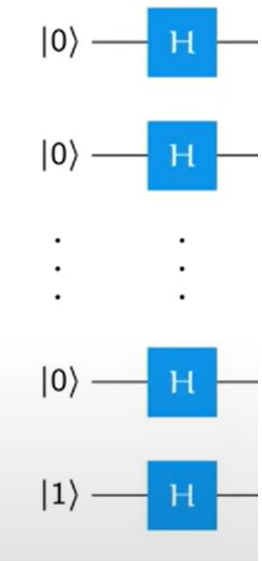
What does this equation mean???

Deutsch-Jozsa Problem

- Similar to Deutsch problem, but for a larger input
- Note: for a larger-input function, we could have sets that are neither balanced nor constant, we ignore these cases (assume f should either be balanced or constant)



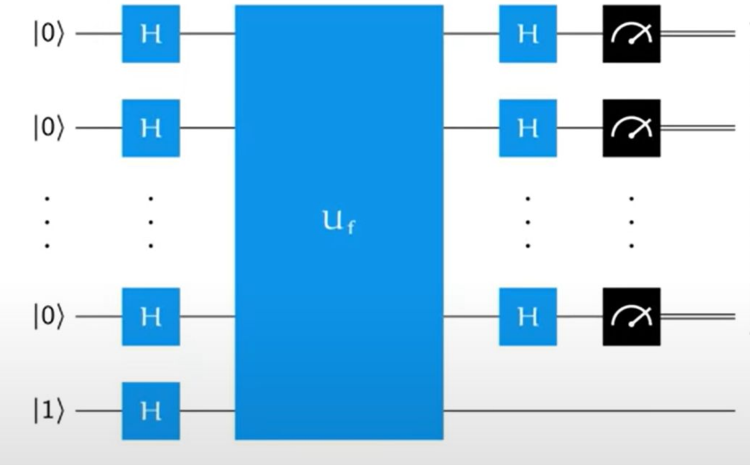
Note on Hadamard Transform



- The Hadamard transform is just the simultaneous application of Hadamard gates across multiple qubits

$$H|x\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}(-1)^x|1\rangle$$
$$= \frac{1}{\sqrt{2}}\sum_{y\in\{0,1\}}(-1)^{xy}|y\rangle$$

$$H^{\otimes n}|x_{n-1}\ldots x_1x_0\rangle$$
$$= (H|x_{n-1}\rangle\otimes\ldots\otimes H|x_0\rangle)$$
$$= \frac{1}{\sqrt{2^n}}\sum_{y\in\Sigma^n}(-1)^{xy}|y\rangle = H^{\otimes n}|x\rangle$$



$$|\pi_1\rangle = |-\rangle \otimes \frac{1}{\sqrt{2^n}} \sum_{x\in\Sigma^n} |x\rangle$$

$$|\pi_2\rangle = |-\rangle \otimes \frac{1}{\sqrt{2^n}} \sum_{x\in\Sigma^n} (-1)^{f(x)} |x\rangle$$

$$|\pi_3\rangle = |-\rangle \otimes \frac{1}{2^n} \sum_{y\in\Sigma^n} \sum_{x\in\Sigma^n} (-1)^{f(x)+x\cdot y} |y\rangle$$

The probability for the measurements to give $y = 0^n$ is

$$p(0^n) = \left| \frac{1}{2^n} \sum_{x\in\Sigma^n} (-1)^{f(x)} \right|^2 = \begin{cases} 1 & \text{if } f \text{ is constant} \\ 0 & \text{if } f \text{ is balanced} \end{cases}$$

How does this compare classically?

- The Deutsch-Jozsa algorithm solves the Deutsch-Jozsa problem <u>without error</u> with a single query (O(1))
- Using brute-force classical algorithm, would take at least $2^{n-1} + 1$ queries (1 more than half of list) ($O(2^n)$)
- A probabilistic algorithm can solve using few queries (O(k))
  - Choose k inputs uniformly at random
  - If $f(x^1) = \ldots = f(x^k)$, then answer 0 (constant), else answer 1 (balanced)
  - If f constant, algorithm is correct with 100% probability
  - If f balanced, algorithm is correct with $1-2^{-k+1}$
  - Shows limited quantum advantage