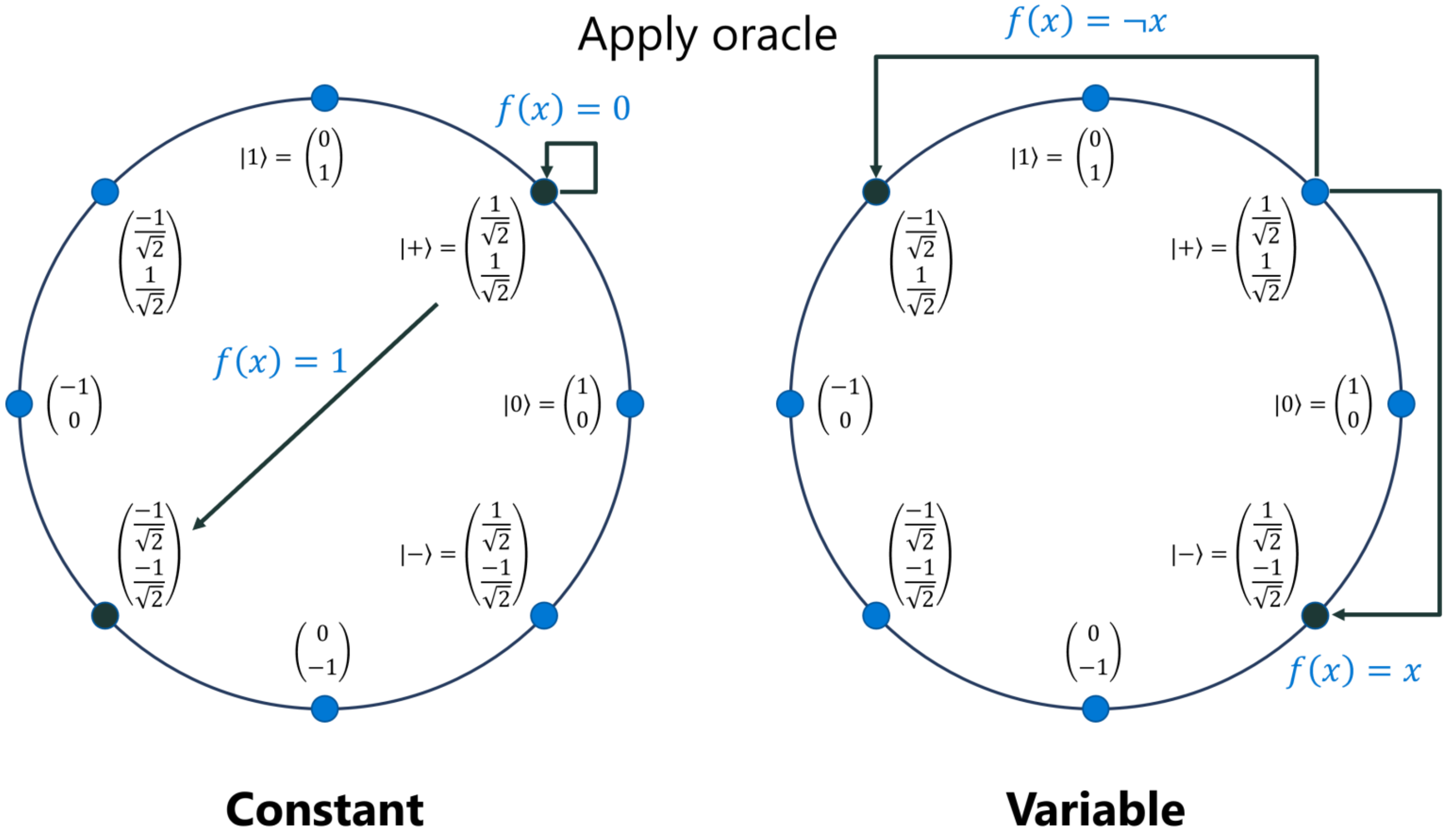
Oracular Algorithms

Mariia Mykhailova
Principal Software Engineer
Microsoft Quantum Systems



Lecture outline

Quantum oracles

Deutsch's algorithm

Deutsch-Jozsa algorithm

Bernstein-Vazirani algorithm

Quantum oracles

Classical oracles

An oracle is a black box operation used in an algorithm

Gives you the ability to do something but not the implementation details

Oracles are used in classical algorithms as well

Must be “just the right size”

- “Adding two numbers” is too small
- “Solving the whole problem that you’re trying to solve” is too big

Oracles can hide a lot of implementation complexity

Including undecidable problems

Frequently used in complexity theory

Phase oracles: definition

Quantum oracle is a black box unitary operation U_f that implements some classical function $f(x)$

Typically, $f : \{0,1\}^N \rightarrow \{0,1\}$ (N-bit input, 1-bit output)

Phase oracles encode $f(x)$ into the phase of the state

$$U_f\,|x\rangle = (-1)^{f(x)}|x\rangle$$

- If $f(x) = 0$, the phase doesn't change
- If $f(x) = 1$, the phase is multiplied by -1
- This defines oracle behavior on the basis states
- Behavior on superposition states follows from linearity of the oracle:

$$U_f \sum_{x=0}^{2^n-1} c_x|x\rangle = \sum_{x=0}^{2^n-1} c_x U_f|x\rangle$$

Example: implementing 1-bit phase oracles

x	$f_1(x)$
0	0
1	0

x	$f_2(x)$
0	1
1	1

x	$f_3(x)$
0	0
1	1

x	$f_4(x)$
0	1
1	0

$f_1(x) \equiv 0$: do nothing

$f_2(x) \equiv 1$: $U_f\,|x\rangle = -|x\rangle$ - apply global phase -1

```
R(PauliI, 2.0 * PI(), x[0]);
```

$f_3(x) \equiv x$: do nothing if $|0\rangle$, apply phase -1 if $|1\rangle$

```
Z(x[0]);
```

$f_4(x) \equiv 1 - x$: do nothing if $|1\rangle$, apply phase -1 if $|0\rangle$

```
X(x[0]); Z(x[0]); X(x[0]);
```

Deutsch’s algorithm

Deutsch's algorithm: problem statement

Consider 1-bit functions $f : \{0,1\} \to \{0,1\}$

x	$f_1(x)$
0	0
1	0

x	$f_2(x)$
0	1
1	1

x	$f_3(x)$
0	0
1	1

x	$f_4(x)$
0	1
1	0

Problem:

Determine if $f(0) = f(1)$ (i.e., $f(0) \oplus f(1) = 0$)

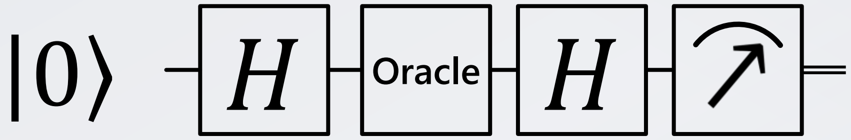
How many queries are required to solve it classically?

Two: you need to query both $f(0)$ and $f(1)$

How many queries to solve it quantumly?

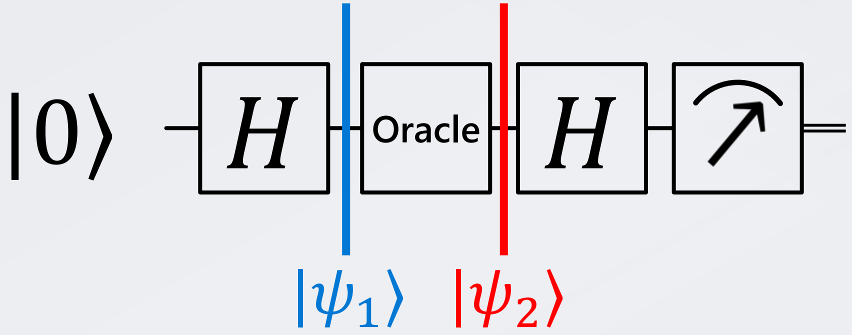
One!

Deutsch's algorithm



- Start with a qubit in the $|0\rangle$ state
- Apply Hadamard gate
- Apply oracle U_f
- Apply Hadamard gate
- Measure the result $f(0) \oplus f(1)$

Deutsch's algorithm



$$|\psi_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$|\psi_2\rangle = U_f \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}}(U_f|0\rangle + U_f|1\rangle) =$$
$$= \frac{1}{\sqrt{2}}((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle)$$

Deutsch's algorithm

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle)$$

x	$f_1(x)$
0	0
1	0

x	$f_2(x)$
0	1
1	1

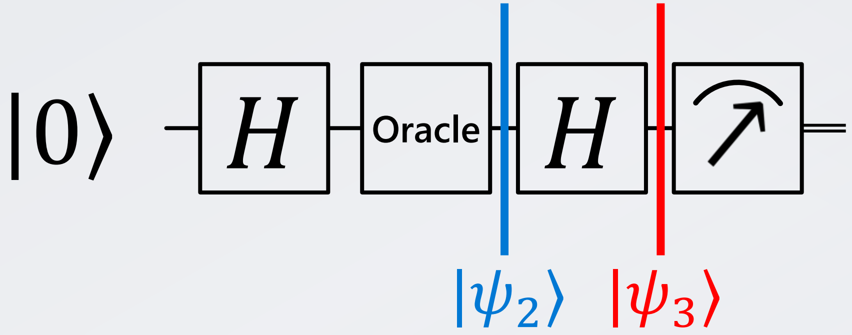
x	$f_3(x)$
0	0
1	1

x	$f_4(x)$
0	1
1	0

$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ $\quad$ $-\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ $\quad$ $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ $\quad$ $\frac{1}{\sqrt{2}}(-|0\rangle + |1\rangle)$

$|+\rangle$ $\quad$ $-|+\rangle$ $\quad$ $|-\rangle$ $\quad$ $-|-\rangle$

Deutsch's algorithm

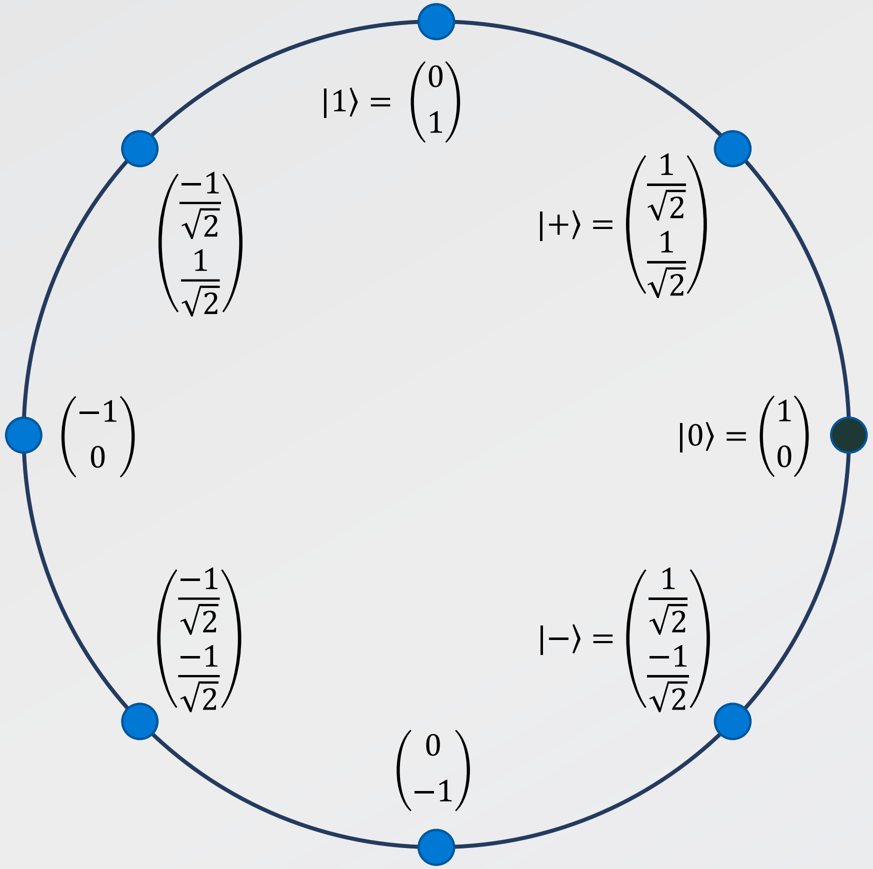


$$|\psi_2\rangle = \begin{cases} \pm|+\rangle & \text{if } f(0) = f(1) \\ \pm|-\rangle & \text{if } f(0) \neq f(1) \end{cases}$$

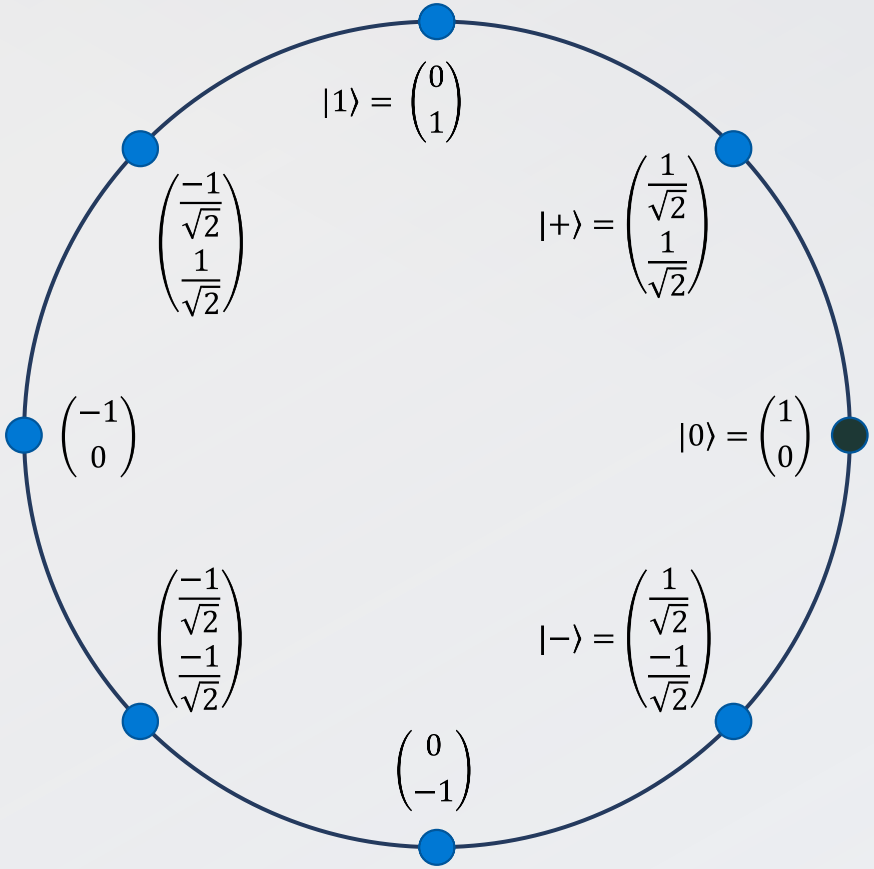
$$|\psi_3\rangle = \begin{cases} \pm|0\rangle & \text{if } f(0) = f(1) \\ \pm|1\rangle & \text{if } f(0) \neq f(1) \end{cases}$$

Deutsch algorithm on the unit circle

Start with the $|0\rangle$ state



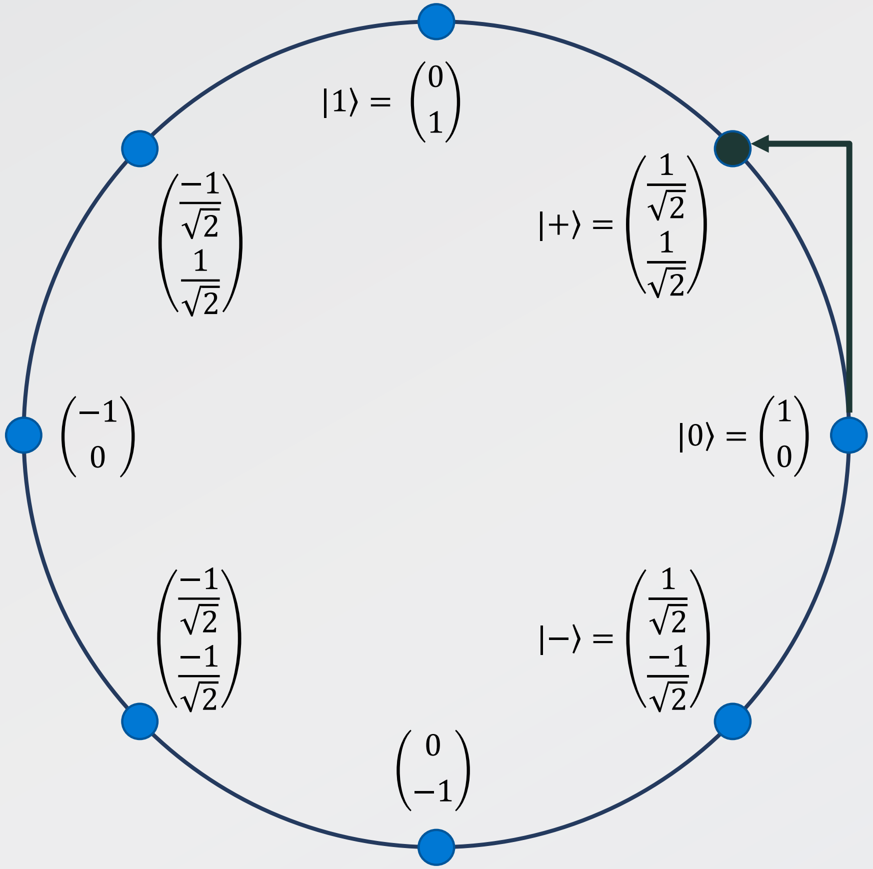
Constant



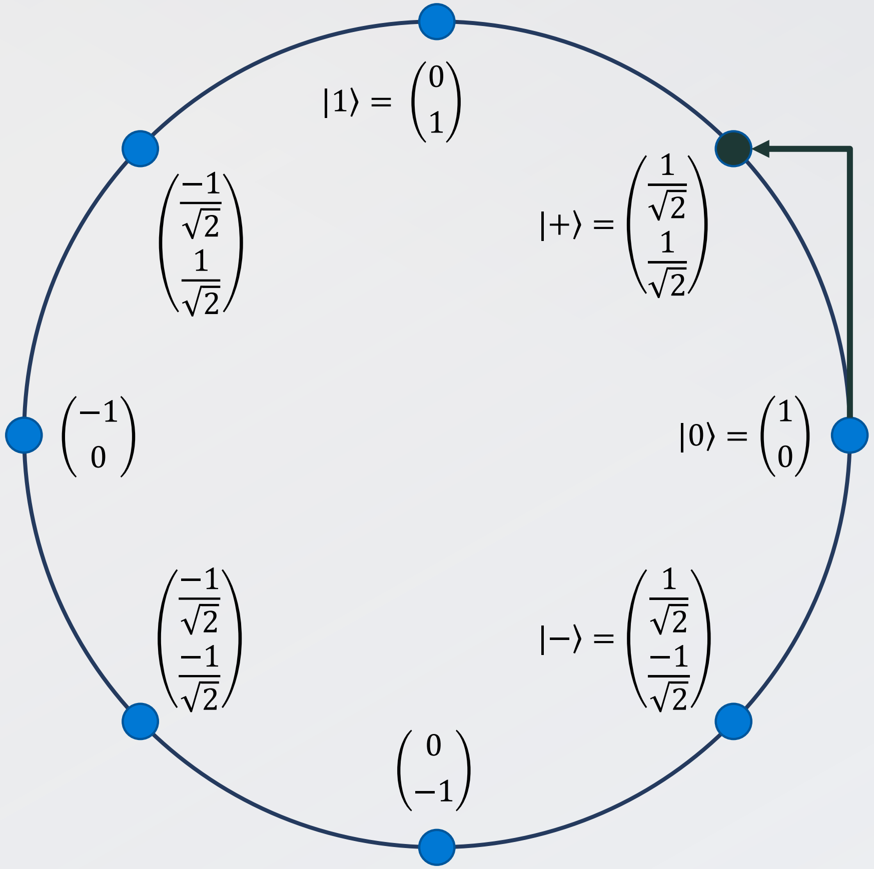
Variable

Deutsch algorithm on the unit circle

Apply Hadamard gate

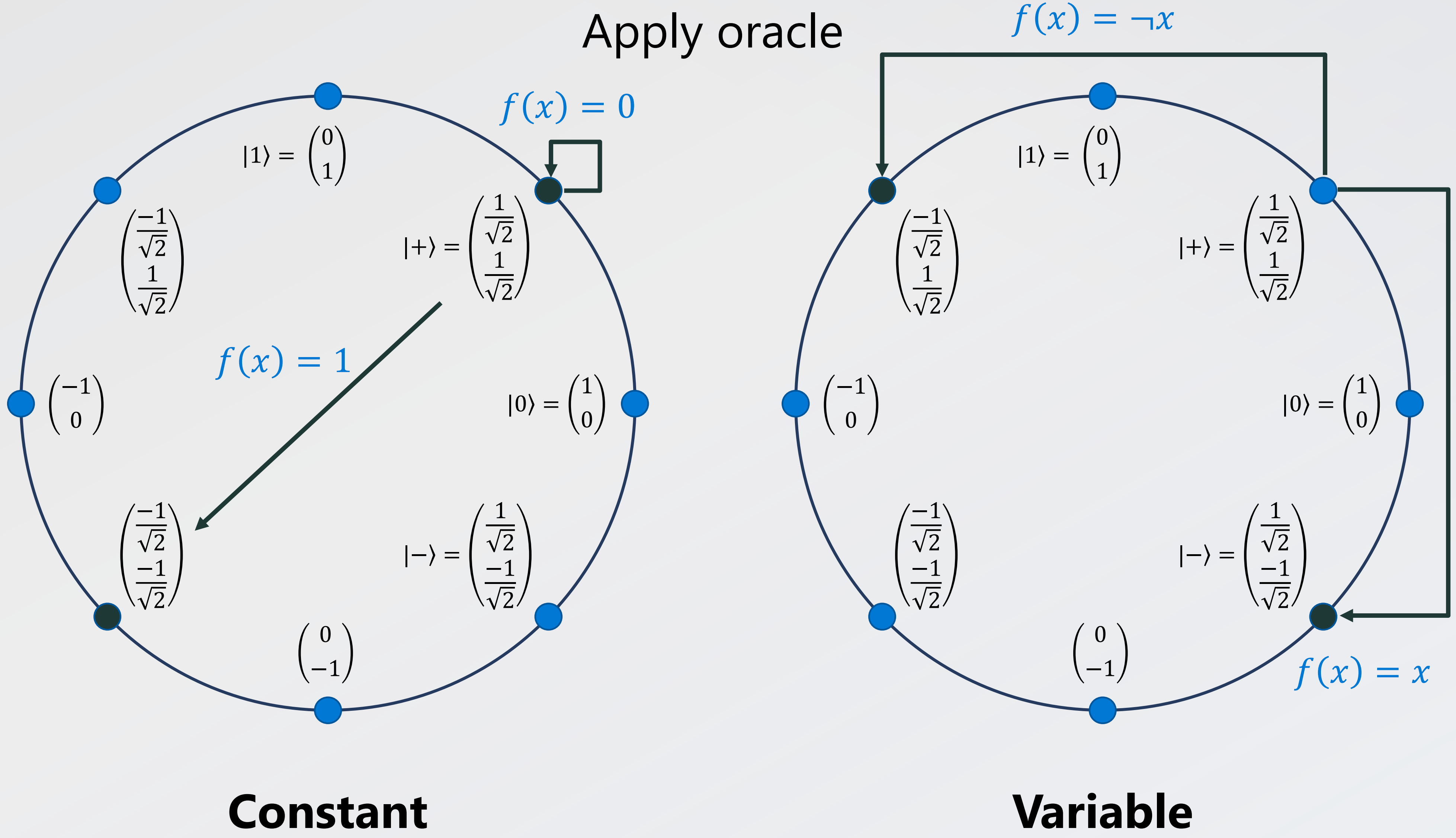


Constant



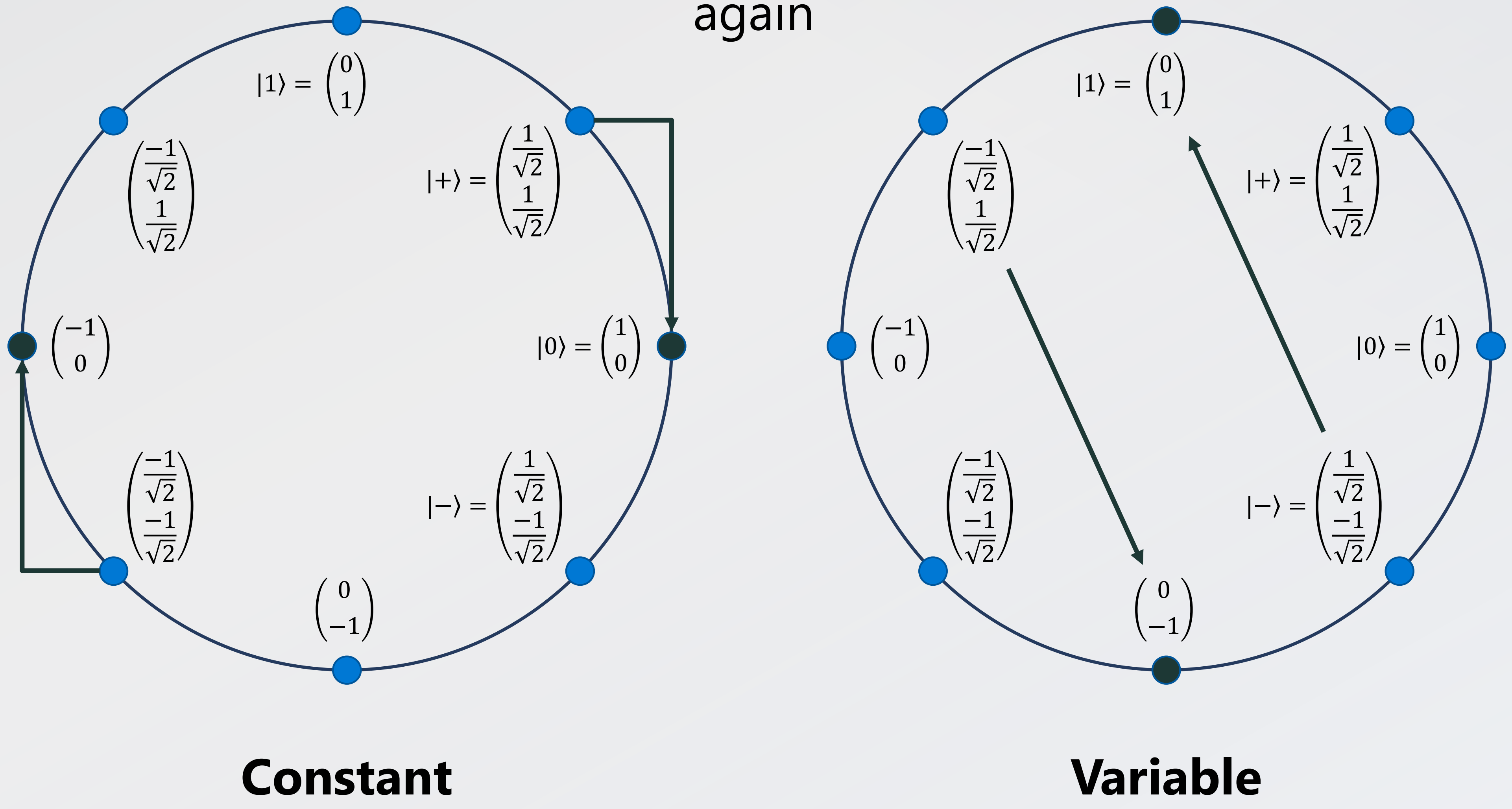
Variable

Deutsch algorithm on the unit circle



Deutsch algorithm on the unit circle

Apply Hadamard gate again



Deutsch's algorithm: final remarks

Doesn't really solve an interesting problem

Illustrates several important features of quantum algorithms

- Applying an oracle calculates the function for multiple inputs
- But you can't access all function values!
- Instead, we use a clever trick to extract a *global property* of a function
- Quantum interference: outcomes we want amplify each other, and outcomes we don't want cancel each other out

$$|\psi_3\rangle = H|\psi_2\rangle = \frac{1}{2}\left((-1)^{f(0)}(|0\rangle + |1\rangle) + (-1)^{f(1)}(|0\rangle - |1\rangle)\right) =$$
$$= \frac{1}{2}\left(\left((-1)^{f(0)} + (-1)^{f(1)}\right)|0\rangle + \left((-1)^{f(0)} - (-1)^{f(1)}\right)|1\rangle\right)$$

Deterministic algorithm

Deutsch-Jozsa algorithm

Deutsch-Jozsa algorithm: problem statement

Consider N-bit functions $f: \{0,1\}^N \to \{0,1\}$

You are guaranteed that the function is

- Either constant (all 0s or all 1s)
- Or balanced (exactly half of the values are 0s and half are 1s)

Problem

Determine whether $f(x)$ is constant or balanced

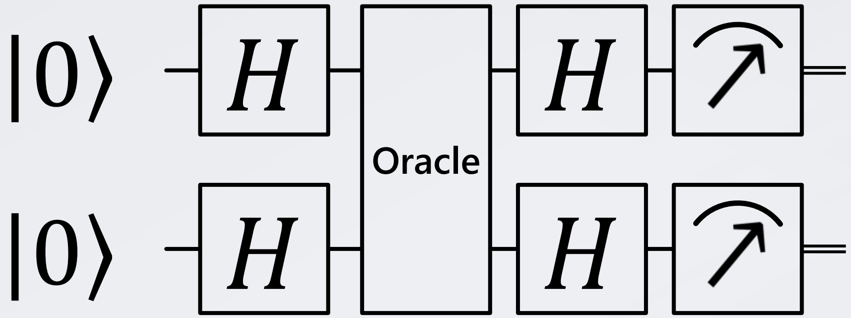
How many queries to solve it classically (deterministically)?

Up to $2^{N-1} + 1$ (first 2^{N-1} queries can return 0 even in balanced case)

How many queries to solve it quantumly?

Still one!

Deutsch-Jozsa algorithm



- Start with all qubits in the $|0\rangle$ state
- Apply Hadamard gate to each qubit
- Apply oracle U_f
- Apply Hadamard gate to each qubit

Measure all qubits:

If all results are 0, the function is constant, otherwise it is balanced

Review: applying Hadamard gate

Applying H gate to a single-qubit basis state $|x\rangle$:

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

$x \cdot z$ is product of bits

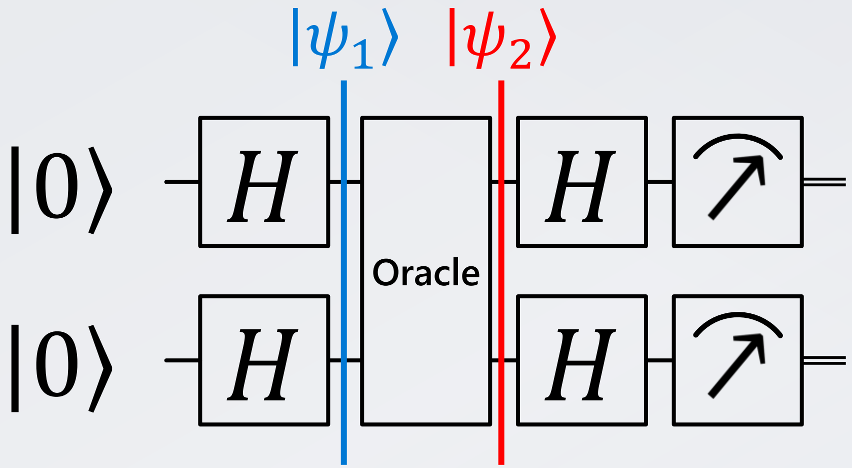
Applying H gate to each qubit of an n-qubit basis state $|x\rangle = |x_1 \dots x_n\rangle$:

$$H^{\otimes n}|x_1 \dots x_n\rangle = \frac{1}{\sqrt{2^n}} \sum_{z_i\in\{0,1\}} (-1)^{x_1 z_1 + \cdots + x_n z_n}|z_1 \dots z_n\rangle$$

$$H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{z=0}^{2^n-1} (-1)^{x\cdot z}|z\rangle$$

$x \cdot z$ is bitwise inner product of integers

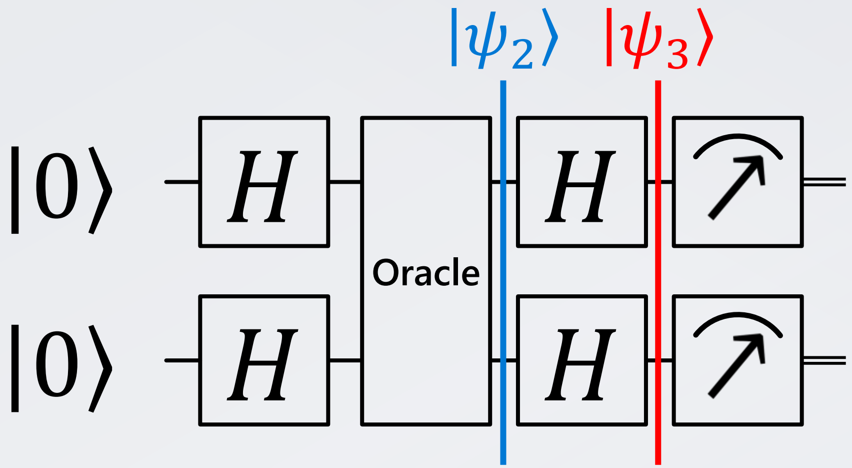
Deutsch-Jozsa algorithm



$$|\psi_1\rangle = H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$$

$$|\psi_2\rangle = U_f|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)}|x\rangle$$

Deutsch-Jozsa algorithm



$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle$$

$$|\psi_3\rangle = \frac{1}{2^n} \sum_{z=0}^{2^n-1} \sum_{x=0}^{2^n-1} (-1)^{x \cdot z + f(x)} |z\rangle$$

Deutsch-Jozsa algorithm

$$|\psi_3\rangle = \frac{1}{2^n} \sum_{z=0}^{2^n-1} \sum_{x=0}^{2^n-1} (-1)^{x \cdot z + f(x)} |z\rangle$$

Let's look at the amplitude of the $|0\rangle^{\otimes n}$ basis state ($z = 0$):

$$\frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)}$$

If f is constant: all summands are the same, so this amplitude is ± 1, all others must be 0

Measurement will yield all 0s

If f is balanced: half of the summands are $+1$ and half are -1, so this amplitude is 0

Measurement will yield another state (can't get a state with 0 amplitude)

Bernstein-Vazirani algorithm

Bernstein-Vazirani algorithm: the problem

Consider N-bit functions $f: \{0,1\}^N \to \{0,1\}$

You are guaranteed that the function can be represented as

$$f(x) = s \cdot x$$

($\cdot$ is dot product of bit vectors modulo 2)

Problem

Determine the hidden bit vector s

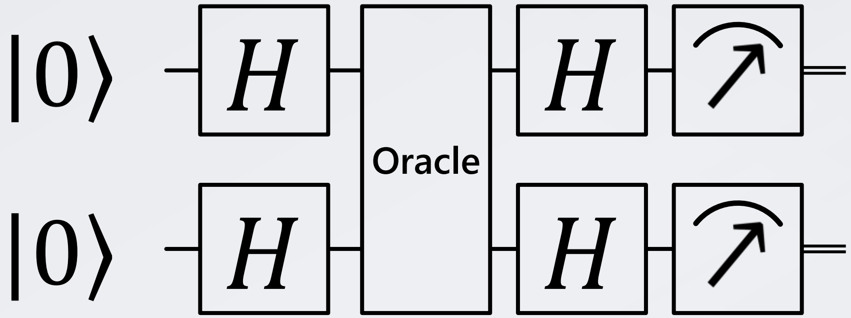
How many queries to solve it classically?

N (query each bit separately using special x)

How many queries to solve it quantumly?

Still one!

Bernstein-Vazirani algorithm



- Start with all qubits in $|0\rangle$ state
- Apply Hadamard gate to each qubit
- Apply oracle U_f
- Apply Hadamard gate to each qubit

Measure all qubits:

The measurement results are the bit vector s

Bernstein-Vazirani algorithm

$$|\psi_3\rangle = \frac{1}{2^n} \sum_{z=0}^{2^n-1} \sum_{x=0}^{2^n-1} (-1)^{x\cdot z + x\cdot s} |z\rangle$$

Let's look at the amplitudes of two basis states:

If $z = s$: for each summand $x \cdot z \oplus x \cdot s = 0$, each phase is $+1$, and the amplitude is 1

If $z \neq s$: we don't need to analyze this case explicitly!

- The state $|\psi_3\rangle$ is normalized, and we know that the amplitude of $|s\rangle$ is 1
- So, the amplitudes of the rest of the states are 0!

$|\psi_3\rangle = |s\rangle$