Quantum graph algorithms

Joran van Apeldoorn

IViR
Instituut voor Informatierecht
Institute for Information Law

QuSoft
Research Center for Quantum Software

October 13, 2021

Basics

Quantum states

- A bit is 0 or 1, a qubit is in a superposition of $|0\rangle$ and $|1\rangle$:

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$$

- If we measure then we get one outcome.
  The probability of measuring $|0\rangle$ is $|\alpha_0|^2$.
  The probability of measuring $|1\rangle$ is $|\alpha_1|^2$.
- Quantum states are normalized complex vectors, the classical states $|0\rangle, |1\rangle, |2\rangle, \ldots$ form a basis.
- For a qubit:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- We combine qubits to create bigger states via tensor products.

Quantum gates

- We can change states by applying unitaries, since they keep vectors normalized.

Quantum gates

- We can change states by applying unitaries, since they keep vectors normalized.
- Unitaries on only a few qubits are called gates.

Quantum gates

- We can change states by applying unitaries, since they keep vectors normalized.
- Unitaries on only a few qubits are called <u>gates</u>.

$$I := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

$$Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Quantum gates

- We can change states by applying unitaries, since they keep vectors normalized.
- Unitaries on only a few qubits are called <u>gates</u>.

$$I := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

$$Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

- I does nothing.

Quantum gates

- We can change states by applying unitaries, since they keep vectors normalized.
- Unitaries on only a few qubits are called <u>gates</u>.

$$I := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

$$Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

- I does nothing.
- X changes $|0\rangle$ into $|1\rangle$ and vice versa.

Quantum gates

- We can change states by applying unitaries, since they keep vectors normalized.
- Unitaries on only a few qubits are called gates.

$$I := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

$$Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

- I does nothing.
- X changes $|0\rangle$ into $|1\rangle$ and vice versa.
- Z adds a -1 in front of $|1\rangle$.

Quantum gates

- We can change states by applying unitaries, since they keep vectors normalized.
- Unitaries on only a few qubits are called <u>gates</u>.

$$I := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

$$Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

- I does nothing.
- X changes $|0\rangle$ into $|1\rangle$ and vice versa.
- Z adds a -1 in front of $|1\rangle$.
- H changes $|0\rangle$ and $|1\rangle$ into $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$.

Changing the basis

Claim: H is a basis transform

Changing the basis

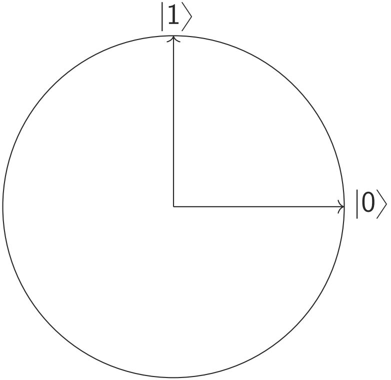
Claim: H is a basis transform

$$|+\rangle := \frac{|0\rangle + |1\rangle}{\sqrt{2}} \qquad |-\rangle := \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Changing the basis

Claim: H is a basis transform

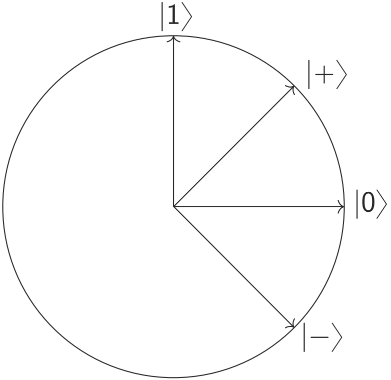
$$|+\rangle := \frac{|0\rangle + |1\rangle}{\sqrt{2}} \qquad |-\rangle := \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$



Changing the basis

Claim: H is a basis transform

$$|+\rangle := \frac{|0\rangle + |1\rangle}{\sqrt{2}} \qquad |-\rangle := \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$



A first circuit

$$-\boxed{H}-\boxed{Z}-\boxed{H}-$$

What does this circuit do? We only need to try a basis (Why?).

A first circuit

What does this circuit do? We only need to try a basis (Why?).

- On $|0\rangle$:

$$|0\rangle \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

A first circuit

What does this circuit do? We only need to try a basis (Why?).

- On $|0\rangle$:

$$|0\rangle \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}} \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

A first circuit

$$-\boxed{H}-\boxed{Z}-\boxed{H}-$$

What does this circuit do? We only need to try a basis (Why?).

- On $|0\rangle$:

$$|0\rangle \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}} \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}} \mapsto \frac{1}{2}\left(|0\rangle + |1\rangle - (|0\rangle - |1\rangle)\right)$$

A first circuit

What does this circuit do? We only need to try a basis (Why?).

- On $|0\rangle$:

$$|0\rangle \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}} \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}} \mapsto \frac{1}{2}\left(|0\rangle + |1\rangle - (|0\rangle - |1\rangle)\right) = |1\rangle$$

A first circuit

What does this circuit do? We only need to try a basis (Why?).

- On $|0\rangle$:

$$|0\rangle \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}} \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}} \mapsto \frac{1}{2}\left(|0\rangle + |1\rangle - (|0\rangle - |1\rangle)\right) = |1\rangle$$

- On $|1\rangle$:

$$|1\rangle \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

A first circuit

$$
-\!\!-\!\!\boxed{H}\!\!-\!\!-\!\!\boxed{Z}\!\!-\!\!-\!\!\boxed{H}\!\!-\!\!-
$$

What does this circuit do? We only need to try a basis (Why?).

- On $|0\rangle$:

$$
|0\rangle \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}} \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}} \mapsto \frac{1}{2}\left(|0\rangle + |1\rangle - (|0\rangle - |1\rangle)\right) = |1\rangle
$$

- On $|1\rangle$:

$$
|1\rangle \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}} \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}}
$$

A first circuit

What does this circuit do? We only need to try a basis (Why?).

- On $|0\rangle$:

$$|0\rangle \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}} \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}} \mapsto \frac{1}{2}\left(|0\rangle + |1\rangle - (|0\rangle - |1\rangle)\right) = |1\rangle$$

- On $|1\rangle$:

$$|1\rangle \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}} \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}} \mapsto \frac{1}{2}\left(|0\rangle + |1\rangle + |0\rangle - |1\rangle\right)$$

A first circuit

What does this circuit do? We only need to try a basis (Why?).

- On $|0\rangle$:

$$|0\rangle \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}} \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}} \mapsto \frac{1}{2}\left(|0\rangle + |1\rangle - (|0\rangle - |1\rangle)\right) = |1\rangle$$

- On $|1\rangle$:

$$|1\rangle \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}} \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}} \mapsto \frac{1}{2}\left(|0\rangle + |1\rangle + |0\rangle - |1\rangle\right) = |0\rangle$$

This is a X gate!

A first circuit

$$—\boxed{H}—\boxed{Z}—\boxed{H}—$$

What does this circuit do? We only need to try a basis (Why?).

- On $|0\rangle$:

$$|0\rangle \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}} \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}} \mapsto \frac{1}{2}\left(|0\rangle + |1\rangle - (|0\rangle - |1\rangle)\right) = |1\rangle$$

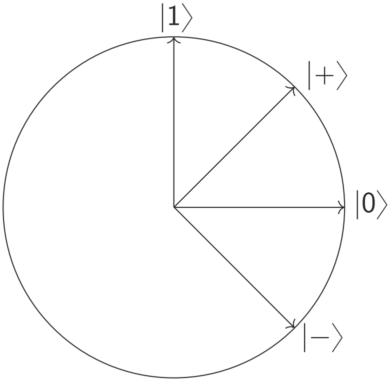
- On $|1\rangle$:

$$|1\rangle \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}} \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}} \mapsto \frac{1}{2}\left(|0\rangle + |1\rangle + |0\rangle - |1\rangle\right) = |0\rangle$$

This is a X gate! Z is just X in the $\{|+\rangle, |-\rangle\}$ basis (and vice versa).

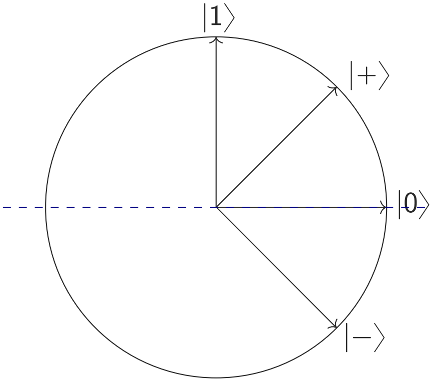
Reflections

We can also see this in our image.



Reflections

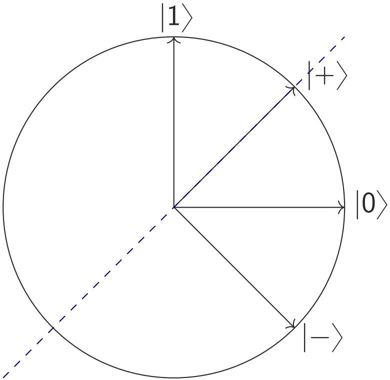
We can also see this in our image.
Z is a reflection through the $|0\rangle$ state.



Reflections

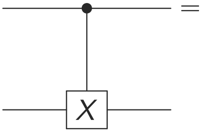
We can also see this in our image.

X is a reflection through the $|+\rangle$ state.

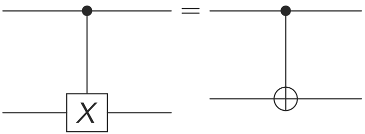


More gates

- We can also make gates controlled:



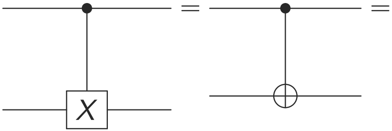
- We can also make gates controlled:



$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

More gates

- We can also make gates controlled:

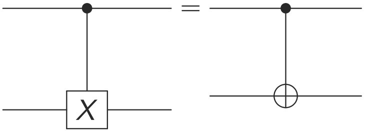


$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

- Classical algorithms make calls to the memory to get the input.

More gates

- We can also make gates controlled:

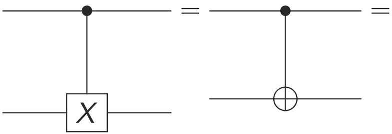


$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

- Classical algorithms make calls to the memory to get the input.
- Quantum algorithms get an <u>oracle</u> that mimics this.

More gates

- We can also make gates controlled:



$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

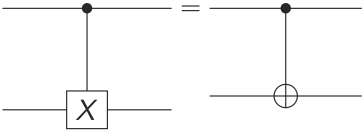
- Classical algorithms make calls to the memory to get the input.
- Quantum algorithms get an oracle that mimics this.
- A binary oracle for an input $x \in \{0,1\}^n$ is a unitary

$$O_x |i\rangle|b\rangle = |i\rangle|b \oplus x_i\rangle$$

where $\oplus$ is addition modulo 2 (or the XOR)

More gates

- We can also make gates controlled:



$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

- Classical algorithms make calls to the memory to get the input.
- Quantum algorithms get an oracle that mimics this.
- A binary oracle for an input $x \in \{0,1\}^n$ is a unitary

$$O_x \, |i\rangle|b\rangle = |i\rangle|b \oplus x_i\rangle$$

  where $\oplus$ is addition modulo 2 (or the XOR)
- Unitaries always have an inverse
  $\Rightarrow$ quantum circuits are always reversible.

Amplitude amplification

Randomized algorithms

Let us get back to classical computing for a while:

Randomized algorithms

Let us get back to classical computing for a while:

- Maybe our randomized algorithm does not always work, it has a success probability p.

Randomized algorithms

Let us get back to classical computing for a while:

- Maybe our randomized algorithm does not always work, it has a success probability p.
- Formally, The algorithm outputs:

Randomized algorithms

Let us get back to classical computing for a while:

- Maybe our randomized algorithm does not always work, it has a success probability p.
- Formally, The algorithm outputs:
  - a 1 and a good solution with probability p, or

Randomized algorithms

Let us get back to classical computing for a while:

- Maybe our randomized algorithm does not always work, it has a success probability p.
- Formally, The algorithm outputs:
  - a 1 and a <u>good</u> solution with probability p, or
  - a 0 and a <u>bad</u> solution with probability $1 - p$.
- An algorithm works <u>with high probability</u> if $p \geqslant 2/3$.

Randomized algorithms

Let us get back to classical computing for a while:

- Maybe our randomized algorithm does not always work, it has a success probability p.
- Formally, The algorithm outputs:
  - a 1 and a good solution with probability p, or
  - a 0 and a bad solution with probability $1 - p$.
- An algorithm works with high probability if $p \geqslant 2/3$.
- What can we do if p is small?

Randomized algorithms

Let us get back to classical computing for a while:

- Maybe our randomized algorithm does not always work, it has a success probability p.
- Formally, The algorithm outputs:
  - a 1 and a good solution with probability p, or
  - a 0 and a bad solution with probability $1 - p$.
- An algorithm works with high probability if $p \geqslant 2/3$.
- What can we do if p is small?
- Repeat

$$\mathcal{O}\left(\frac{1}{p}\right)$$

  times.

Quantum algorithms & small success probability

Back to quantum:

Quantum algorithms & small success probability

Back to quantum:

- Quantum algorithms are inherently random.

Quantum algorithms & small success probability

Back to quantum:

- Quantum algorithms are inherently random.
- A Hadamard gate can be used to flip a coin:

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

Quantum algorithms & small success probability

Back to quantum:

- Quantum algorithms are inherently random.
- A Hadamard gate can be used to flip a coin:

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

- A quantum algorithm might produce the state

$$U|0\rangle = \alpha_G |G\rangle |1\rangle + \alpha_B |B\rangle |0\rangle$$

Quantum algorithms & small success probability

Back to quantum:

- Quantum algorithms are inherently random.
- A Hadamard gate can be used to flip a coin:

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

- A quantum algorithm might produce the state

$$U|0\rangle = \alpha_G |G\rangle|1\rangle + \alpha_B |B\rangle|0\rangle$$

- $|G\rangle|1\rangle$ is the "Good" part of the state, $|B\rangle|0\rangle$ is the "Bad" part.

Quantum algorithms & small success probability

Back to quantum:

- Quantum algorithms are inherently random.
- A Hadamard gate can be used to flip a coin:

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

- A quantum algorithm might produce the state

$$U|0\rangle = \alpha_G |G\rangle |1\rangle + \alpha_B |B\rangle |0\rangle$$

- $|G\rangle |1\rangle$ is the "Good" part of the state, $|B\rangle |0\rangle$ is the "Bad" part.
- Is we just measure then the success probability is $p = |\alpha_G|^2$.

Three states

Let us write $|\psi\rangle := \alpha_G |G\rangle |1\rangle + \alpha_B |B\rangle |0\rangle$

Three states

Let us write $|\psi\rangle := \alpha_G |G\rangle |1\rangle + \alpha_B |B\rangle |0\rangle$

- What is the inner product between the good and the bad part?

Three states

Let us write $|\psi\rangle := \alpha_G |G\rangle |1\rangle + \alpha_B |B\rangle |0\rangle$

- What is the inner product between the good and the bad part?

$$\langle G|B\rangle\langle 1|0\rangle = \langle G|B\rangle \cdot 0 = 0$$

Three states

Let us write $|\psi\rangle := \alpha_G |G\rangle |1\rangle + \alpha_B |B\rangle |0\rangle$

- What is the inner product between the good and the bad part?

$$\langle G|B\rangle\langle 1|0\rangle = \langle G|B\rangle \cdot 0 = 0$$

- So these two states are orthogonal!

Three states

Let us write $|\psi\rangle := \alpha_G |G\rangle |1\rangle + \alpha_B |B\rangle |0\rangle$

- What is the inner product between the good and the bad part?

$$\langle G|B\rangle\langle 1|0\rangle = \langle G|B\rangle \cdot 0 = 0$$

- So these two states are orthogonal!
- $|\psi\rangle$ is written in the $\{|G\rangle|1\rangle, |B\rangle|0\rangle\}$ basis.

Three states

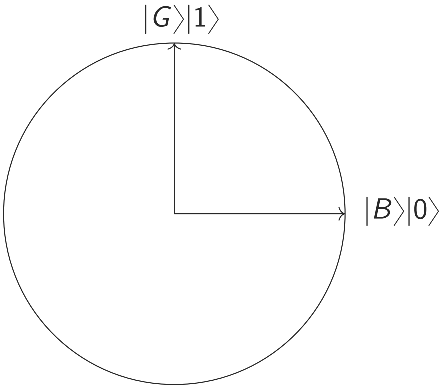
Let us write $|\psi\rangle := \alpha_G |G\rangle |1\rangle + \alpha_B |B\rangle |0\rangle$

- What is the inner product between the good and the bad part?

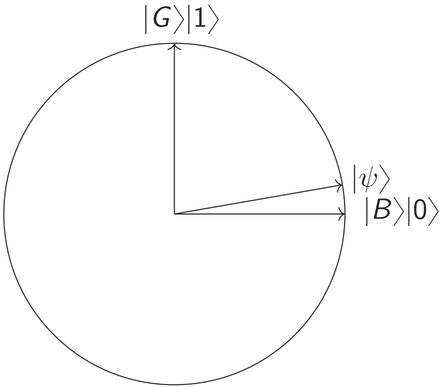
$$\langle G|B\rangle \langle 1|0\rangle = \langle G|B\rangle \cdot 0 = 0$$

- So these two states are orthogonal!
- $|\psi\rangle$ is written in the $\{|G\rangle|1\rangle, |B\rangle|0\rangle\}$ basis.
- Everything is in a 2-dimensional subspace.

Three states - a picture

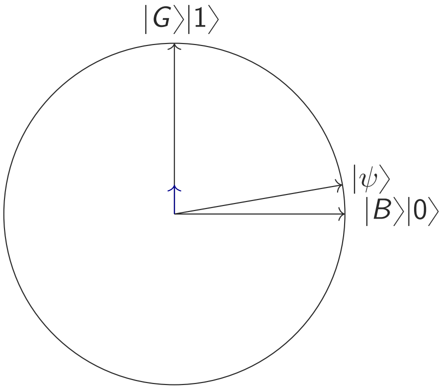


Three states - a picture



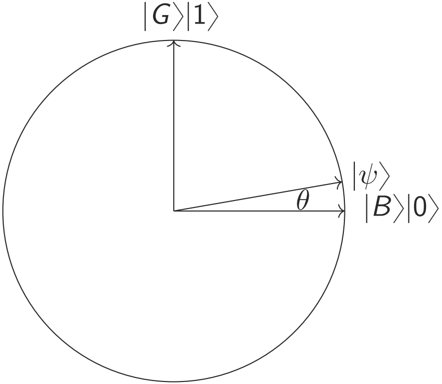
$$|\psi\rangle = \qquad \alpha_G\,|G\rangle\,|1\rangle + \qquad \alpha_B\,|B\rangle\,|0\rangle$$

Three states - a picture



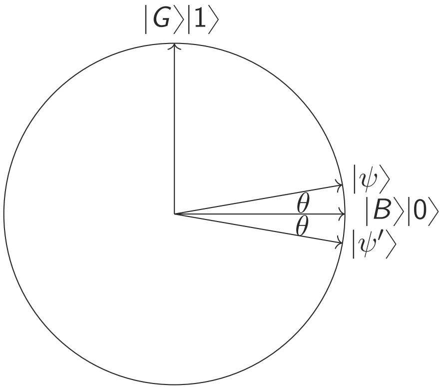
$$|\psi\rangle = \qquad \alpha_G\,|G\rangle\,|1\rangle + \qquad \alpha_B\,|B\rangle\,|0\rangle$$

Three states - a picture



$$|\psi\rangle = \sin(\theta)\ \ |G\rangle\,|1\rangle + \cos(\theta)\ \ |B\rangle\,|0\rangle$$

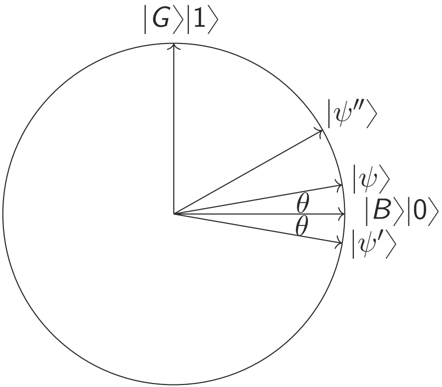
Three states - a picture



$$|\psi\rangle = \sin(\theta)\ \ |G\rangle|1\rangle + \cos(\theta)\ \ |B\rangle|0\rangle$$

$$\left|\psi'\right\rangle = -\sin(\theta)\,|G\rangle\,|1\rangle + \cos(\theta)\,|B\rangle\,|0\rangle$$

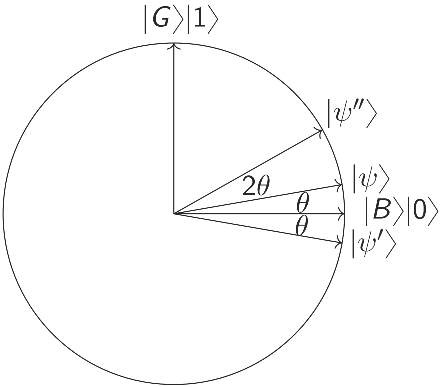
Three states - a picture



$$|\psi\rangle = \sin(\theta)\ \ |G\rangle|1\rangle + \cos(\theta)\ \ |B\rangle|0\rangle$$

$$|\psi'\rangle = -\sin(\theta)\,|G\rangle|1\rangle + \cos(\theta)\,|B\rangle|0\rangle$$

Three states - a picture

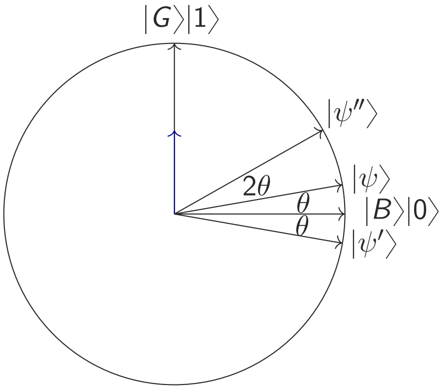


$$|\psi\rangle = \sin(\theta) \ |G\rangle|1\rangle + \cos(\theta) \ |B\rangle|0\rangle$$

$$|\psi'\rangle = -\sin(\theta)|G\rangle|1\rangle + \cos(\theta)|B\rangle|0\rangle$$

$$|\psi''\rangle = \sin(3\theta)|G\rangle|1\rangle + \cos(3\theta)|B\rangle|0\rangle$$

Three states - a picture



$$|\psi\rangle = \sin(\theta)\ \ |G\rangle|1\rangle + \cos(\theta)\ \ |B\rangle|0\rangle$$

$$|\psi'\rangle = -\sin(\theta)|G\rangle|1\rangle + \cos(\theta)|B\rangle|0\rangle$$

$$|\psi''\rangle = \sin(3\theta)|G\rangle|1\rangle + \cos(3\theta)|B\rangle|0\rangle$$

Amplitude amplification

- Two reflections: through $|B\rangle|0\rangle$ and $|\psi\rangle$.

Amplitude amplification

- Two reflections: through $|B\rangle|0\rangle$ and $|\psi\rangle$.
- The product is a rotation $\mathcal{A}$, with angle 2θ.

Amplitude amplification

- Two reflections: through $|B\rangle|0\rangle$ and $|\psi\rangle$.
- The product is a <u>rotation</u> $\mathcal{A}$, with angle 2θ.
- After k iterations of $\mathcal{A}$ we get

$$\sin\left((2k+1)\theta\right)|G\rangle|1\rangle + \cos\left((2k+1)\right)|B\rangle|0\rangle$$

Amplitude amplification

- Two reflections: through $|B\rangle|0\rangle$ and $|\psi\rangle$.
- The product is a <u>rotation</u> $\mathcal{A}$, with angle 2θ.
- After k iterations of $\mathcal{A}$ we get

$$\sin\left((2k+1)\theta\right)|G\rangle|1\rangle + \cos\left((2k+1)\right)|B\rangle|0\rangle$$

- If $(2k+1)\theta \approx \pi/2$ then $\left|\sin\left((2k+1)\theta\right)\right|^2 \approx 1$.

Amplitude amplification

- Two reflections: through $|B\rangle|0\rangle$ and $|\psi\rangle$.
- The product is a rotation $\mathcal{A}$, with angle 2θ.
- After k iterations of $\mathcal{A}$ we get

$$\sin\left((2k+1)\theta\right)|G\rangle|1\rangle + \cos\left((2k+1)\right)|B\rangle|0\rangle$$

- If $(2k+1)\theta \approx \pi/2$ then $\left|\sin\left((2k+1)\theta\right)\right|^2 \approx 1$.
- Since $\theta = \arcsin\alpha_G$ we want

$$k \approx \left(\frac{\pi}{2\arcsin\alpha_G} - 1\right)/2$$

Amplitude amplification

- Two reflections: through $|B\rangle|0\rangle$ and $|\psi\rangle$.
- The product is a <u>rotation</u> $\mathcal{A}$, with angle 2θ.
- After k iterations of $\mathcal{A}$ we get

$$\sin\left((2k+1)\theta\right)|G\rangle|1\rangle + \cos\left((2k+1)\right)|B\rangle|0\rangle$$

- If $(2k+1)\theta \approx \pi/2$ then $\left|\sin\left((2k+1)\theta\right)\right|^2 \approx 1$.
- Since $\theta = \arcsin \alpha_G$ we want

$$k \approx \left(\frac{\pi}{2\arcsin \alpha_G} - 1\right)/2 = \mathcal{O}\left(\frac{1}{|\alpha_G|}\right)$$

Amplitude amplification

- Two reflections: through $|B\rangle|0\rangle$ and $|\psi\rangle$.
- The product is a <u>rotation</u> $\mathcal{A}$, with angle 2θ.
- After k iterations of $\mathcal{A}$ we get

$$\sin\left((2k+1)\theta\right)|G\rangle|1\rangle + \cos\left((2k+1)\right)|B\rangle|0\rangle$$

- If $(2k+1)\theta \approx \pi/2$ then $\left|\sin\left((2k+1)\theta\right)\right|^2 \approx 1$.
- Since $\theta = \arcsin \alpha_G$ we want

$$k \approx \left(\frac{\pi}{2\arcsin\alpha_G} - 1\right)/2 = \mathcal{O}\left(\frac{1}{|\alpha_G|}\right) = \mathcal{O}\left(\frac{1}{\sqrt{p}}\right)$$

Amplitude amplification

- Two reflections: through $|B\rangle|0\rangle$ and $|\psi\rangle$.
- The product is a <u>rotation</u> $\mathcal{A}$, with angle 2θ.
- After k iterations of $\mathcal{A}$ we get

$$\sin\left((2k+1)\theta\right)|G\rangle|1\rangle + \cos\left((2k+1)\right)|B\rangle|0\rangle$$

- If $(2k+1)\theta \approx \pi/2$ then $\left|\sin\left((2k+1)\theta\right)\right|^2 \approx 1$.
- Since $\theta = \arcsin \alpha_G$ we want

$$k \approx \left(\frac{\pi}{2\arcsin \alpha_G} - 1\right)/2 = \mathcal{O}\left(\frac{1}{|\alpha_G|}\right) = \mathcal{O}\left(\frac{1}{\sqrt{p}}\right)$$

- Nice, but can we actually implement these reflections?

Implementing the reflections

The reflection through $|B\rangle|0\rangle$:

Implementing the reflections

The reflection through $|B\rangle|0\rangle$:

- Do nothing to the bad state.
- Add a -1 to the good state.

Implementing the reflections

The reflection through $|B\rangle|0\rangle$:

- Do nothing to the bad state.
- Add a -1 to the good state.

Apply a Z gate to the last bit.

Implementing the reflections

The reflection through $|B\rangle|0\rangle$:

- Do nothing to the bad state.
- Add a -1 to the good state.

Apply a Z gate to the last bit.

The reflection through $|\psi\rangle$:

Implementing the reflections

The reflection through $|B\rangle|0\rangle$:

- Do nothing to the bad state.
- Add a -1 to the good state.

Apply a Z gate to the last bit.

The reflection through $|\psi\rangle$:

- Do nothing to $|\psi\rangle$.
- Add a -1 to states orthogonal to it.

Implementing the reflections

The reflection through $|B\rangle|0\rangle$:

- Do nothing to the bad state.
- Add a -1 to the good state.

Apply a Z gate to the last bit.

The reflection through $|\psi\rangle$:

- Do nothing to $|\psi\rangle$.
- Add a -1 to states orthogonal to it.

Use that $|\psi\rangle = U\,|0\rangle$:

1. Apply U^{-1} to map $|\psi\rangle$ to $|0\rangle$.
2. Reflect through $|0\rangle$.
3. Apply U to map $|0\rangle$ to back to $|\psi\rangle$.

Example: the search problem

Before amplitude amplification there was Grover ('96).

Example: the search problem

Before amplitude amplification there was Grover ('96).

Search problem

Input: $x \in \{0,1\}^N$ with k ones.
Goal: Find an i such that $x_i = 1$ with few queries.

Example: the search problem

Before amplitude amplification there was Grover ('96).

Search problem

Input: $x \in \{0,1\}^N$ with k ones.
Goal: Find an i such that $x_i = 1$ with few queries.

How can we do this classically?

Example: the search problem

Before amplitude amplification there was Grover ('96).

Search problem

Input: $x \in \{0,1\}^N$ with k ones.
Goal: Find an i such that $x_i = 1$ with few queries.

How can we do this classically?

- Go over the bits: $\mathcal{O}(N)$ queries.

Example: the search problem

Before amplitude amplification there was Grover ('96).

Search problem

Input: $x \in \{0,1\}^N$ with k ones.
Goal: Find an i such that $x_i = 1$ with few <u>queries</u>.

How can we do this classically?

- Go over the bits: $\mathcal{O}(N)$ queries.
- Randomly pick an i and repeat: $\mathcal{O}(N/k)$ queries.

Example: the search problem

Before amplitude amplification there was Grover ('96).

Search problem

Input: $x \in \{0,1\}^N$ with k ones.
Goal: Find an i such that $x_i = 1$ with few <u>queries</u>.

How can we do this classically?

- Go over the bits: $\mathcal{O}(N)$ queries.
- Randomly pick an i and repeat: $\mathcal{O}(N/k)$ queries.

Using amplitude amplification:

- Superposition over i, x_i and amplify: $\mathcal{O}\left(\sqrt{N/k}\right)$ queries.

Example: the search problem

Before amplitude amplification there was Grover ('96).

Search problem

Input: $x \in \{0,1\}^N$ with k ones.
Goal: Find an i such that $x_i = 1$ with few queries.

How can we do this classically?

- Go over the bits: $\mathcal{O}(N)$ queries.
- Randomly pick an i and repeat: $\mathcal{O}(N/k)$ queries.

Using amplitude amplification:

- Superposition over i, x_i and amplify: $\mathcal{O}\left(\sqrt{N/k}\right)$ queries.

To find all: $O(\sqrt{Nk})$

Graphs

Building a spanning tree

Goal: given adjacency matrix queries for a connected graph, find a spanning tree.

Building a spanning tree

Goal: given adjacency matrix queries for a connected graph, find a spanning tree.

- Start with an empty tree and $c = n$ components.

Building a spanning tree

Goal: given adjacency matrix queries for a connected graph, find a spanning tree.

- Start with an empty tree and $c = n$ components.
- There are at least $c - 1$ edges that connect two components, out of n^2 possible edges.

Building a spanning tree

Goal: given adjacency matrix queries for a connected graph, find a spanning tree.

- Start with an empty tree and $c = n$ components.
- There are at least $c - 1$ edges that connect two components, out of n^2 possible edges.
- Grover search using $O(\sqrt{n^2/(c-1)})$ queries to find such edge.

Building a spanning tree

Goal: given adjacency matrix queries for a connected graph, find a spanning tree.

- Start with an empty tree and $c = n$ components.
- There are at least $c - 1$ edges that connect two components, out of n^2 possible edges.
- Grover search using $O(\sqrt{n^2/(c-1)})$ queries to find such edge.
- Repeat n times:

Building a spanning tree

Goal: given adjacency matrix queries for a connected graph, find a spanning tree.

- Start with an empty tree and $c = n$ components.
- There are at least $c - 1$ edges that connect two components, out of n^2 possible edges.
- Grover search using $O(\sqrt{n^2/(c-1)})$ queries to find such edge.
- Repeat n times:

$$\sum_{c=2}^{n} n\sqrt{1/(c-1)} \leqslant n \int_0^n c^{-1/2} dc = O(n^{1.5})$$

BFS

Goal: given adjacency list queries, perform a breadth-first search.

BFS

Goal: given adjacency list queries, perform a breadth-first search.

- In BFS for each node we do:

BFS

Goal: given adjacency list queries, perform a breadth-first search.

- In BFS for each node we do:
  - Check all outgoing edges for unvisited nodes.

BFS

Goal: given adjacency list queries, perform a breadth-first search.

- In BFS for each node we do:
  - Check all outgoing edges for unvisited nodes.
  - Add those nodes to the queue

BFS

Goal: given adjacency list queries, perform a breadth-first search.

- In BFS for each node we do:
  - Check all outgoing edges for unvisited nodes.
  - Add those nodes to the queue
- Classically we check each edge twice: $O(E)$ queries/time.

BFS

Goal: given adjacency list queries, perform a breadth-first search.

- In BFS for each node we do:
  - Check all outgoing edges for unvisited nodes.
  - Add those nodes to the queue
- Classically we check each edge twice: $O(E)$ queries/time.
- Quantumly we may use Grover's search:

BFS

Goal: given adjacency list queries, perform a breadth-first search.

- In BFS for each node we do:
  - Check all outgoing edges for unvisited nodes.
  - Add those nodes to the queue
- Classically we check each edge twice: $O(E)$ queries/time.
- Quantumly we may use Grover's search:
  - Say node v_j has d_j neighbors, t_j not visited.

BFS

Goal: given adjacency list queries, perform a breadth-first search.

- In BFS for each node we do:
  - Check all outgoing edges for unvisited nodes.
  - Add those nodes to the queue
- Classically we check each edge twice: $O(E)$ queries/time.
- Quantumly we may use Grover's search:
  - Say node v_j has d_j neighbors, t_j not visited.
  - We know $\sum_j d_j = 2m$, $\sum_j t_j = n$.

BFS

Goal: given adjacency list queries, perform a breadth-first search.

- In BFS for each node we do:
  - Check all outgoing edges for unvisited nodes.
  - Add those nodes to the queue
- Classically we check each edge twice: $O(E)$ queries/time.
- Quantumly we may use Grover's search:
  - Say node v_j has d_j neighbors, t_j not visited.
  - We know $\sum_j d_j = 2m$, $\sum_j t_j = n$.

$$\sum_{j=1}^{n} \sqrt{d_j t_j} \leqslant \sqrt{\sum_{j=1}^{n} d_j \sum_{j=1}^{n} t_j} = O(\sqrt{nm})$$

BFS

Goal: given adjacency list queries, perform a breadth-first search.

- In BFS for each node we do:
  - Check all outgoing edges for unvisited nodes.
  - Add those nodes to the queue
- Classically we check each edge twice: $O(E)$ queries/time.
- Quantumly we may use Grover's search:
  - Say node v_j has d_j neighbors, t_j not visited.
  - We know $\sum_j d_j = 2m$, $\sum_j t_j = n$.

$$\sum_{j=1}^{n} \sqrt{d_j t_j} \leqslant \sqrt{\sum_{j=1}^{n} d_j \sum_{j=1}^{n} t_j} = O(\sqrt{nm})$$

Application: Matching in $O(V\sqrt{E})$

That was it!