Universal Quantum Gates

Juha-Matti Huusko

University of Eastern Finland

Wikipedia:

A set of universal quantum gates is **any set of gates** to which any operation possible on a quantum computer can be reduced, that is, any other unitary operation can be expressed as a finite sequence of gates from the set. https://en.wikipedia.org/wiki/Quantum_logic_gate#Universal_quantum_gates

An infinite number of gates

Quantum gates can be represented with 2x2 unitary matrices.

$$H = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}, \quad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} = T^2,$$

$$R_x(\theta) = \begin{bmatrix} \cos(\theta/2) & i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$$

There is an infinite number of interesting quantum gates, for example,

$$R_x(\pi/2), R_2(\pi/3), R_3(\pi/4), \ldots$$

which can be needed in calculations.

An infinite number of gates

UNIVERSITY OF EASTERN FINLAND

Example

A circuit for Quantum Fourier Transform is composed of H gates and the controlled version of

$$R_m = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^m}. \end{bmatrix}$$

Here m can be any number. Depending on the situation, a different R_m is needed.

https://en.wikipedia.org/wiki/Quantum_Fourier_transform

How to find a set of universal quantum gates? Which gates are enough?

$\{CNOT, H, S\}$ is not enough

UNIVERSITY OF EASTERN FINLAND

Theorem (Gottesman–Knill 1998)

A quantum computer using

- *Preparation of qubits in computational basis states,*
- *gates $\{CNOT, H, S\}$ (so-called Clifford gates)*
- *Measurements in the computational basis.*

can be simulated efficiently on a classical computer.

Not all quantum circuits can be simulated efficiently on a classical computer. (This was mentioned during the course.)

Therefore, not all quantum circuits can be expressed by gates $\{CNOT, H, S\}$.

https://en.wikipedia.org/wiki/Gottesman%E2%80%93Knill_theorem

$\{CNOT, H, S\}$ is not enough, but add T!

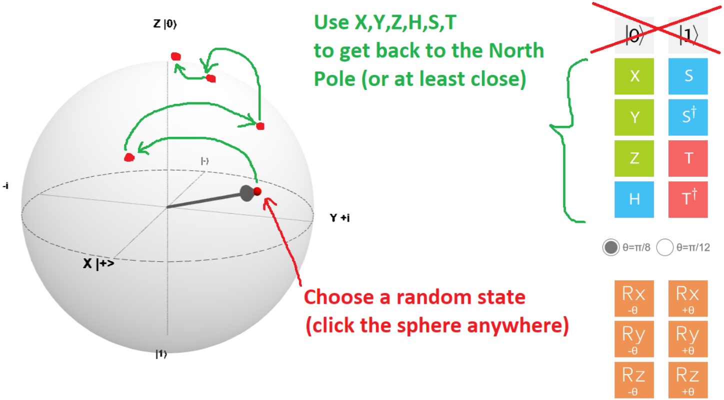
UNIVERSITY OF EASTERN FINLAND

The set $\{CNOT, H, S, T\}$ is a set of universal quantum gates.

Let's not look for a proof.

Playing with the Bloch sphere for 2 mins can convince.

$\{CNOT, H, S, T\}$ is enough



https://sami.andberg.net/bloch/bloch.html

$\{CNOT, H, S, T\}$ is enough, but is it "fast"?

UNIVERSITY OF EASTERN FINLAND

If we have a gate G, how many gates from $\{CNOT, H, S, T\}$ are needed to approximate it?

Example

Let G be some strange quantum gate. Let P be a product of universal gates. Let $G - P$ have absolute values of its elements less than 0.001. How many factors P usually has? 10?, 100?, 1000?

$\{CNOT, H, S, T\}$ is enough and fast

UNIVERSITY OF EASTERN FINLAND

Lause (Solovay-Kitaev)

If U is a set of universal gates, then any gate G can be approximated by a "fairly short" sequence of gates.

https://en.wikipedia.org/wiki/Solovay%E2%80%93Kitaev_theorem

"The algorithm runs in $O(\log^{2.71}(1/\varepsilon))$ time, and produces as output a sequence of $O(\log^{3.97}(1/\varepsilon))$ quantum gates which is guaranteed to approximate the desired quantum gate to an accuracy within ε."
https://arxiv.org/abs/quant-sinxsfbxβph/0505030

$\{CNOT, H, S, T\}$ is enough and fast

UNIVERSITY OF EASTERN FINLAND

Example

Let G be some strange quantum gate. Let P be a product of universal gates. Let $G - P$ have absolute values of its elements less than 0.001. How many factors P usually has? 10?, 100?, 1000?

Now $\varepsilon = 0.001$ and $1/\varepsilon = 1000$. Hence

$$C \log^{3.97}(1000) \approx C(\log(1000))^4 = C3^4 = 81C$$

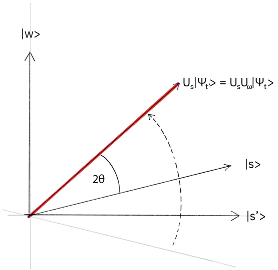
gates are needed.

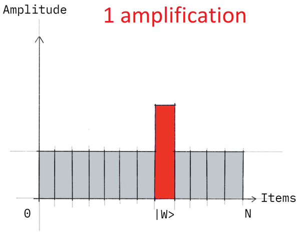
For $\varepsilon = 0.01$, only $C(\log 100)^4 = 2^4 = 16C$ gates are needed.

(Here C is the constant in the "big-Oh" $O(\log^{3.97}(1/\varepsilon))$.)

UNIVERSITY OF EASTERN FINLAND

Grover's algorithm uses the the amplitude amplification trick

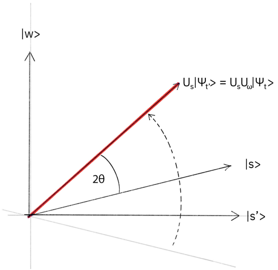


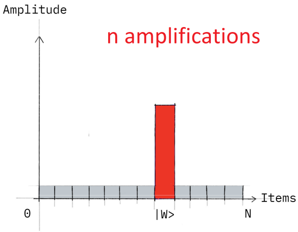


https://quantum-sinxsfbxβcomputing.ibm.com/composer/docs/iqx/guide/grovers-sinxsfbxβalgorithm

Perhaps no need to be so accurate

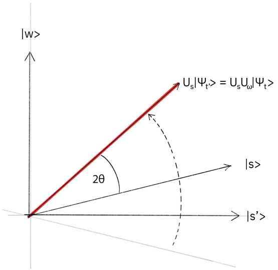
Grover's algorithm uses the the amplitude amplification trick

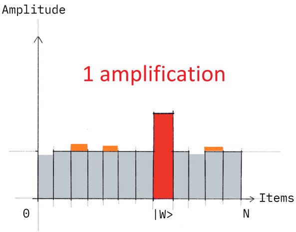




https://quantum-sinxsfbxβcomputing.ibm.com/composer/docs/iqx/guide/grovers-sinxsfbxβalgorithm

Grover's algorithm uses the the amplitude amplification trick

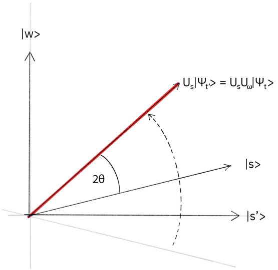


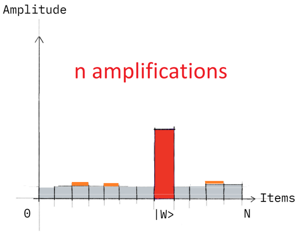


https://quantum-sinxsfbxβcomputing.ibm.com/composer/docs/iqx/guide/grovers-sinxsfbxβalgorithm

Perhaps no need to be so accurate

Grover's algorithm uses the the amplitude amplification trick





https://quantum-sinxsfbxβcomputing.ibm.com/composer/docs/iqx/guide/grovers-sinxsfbxβalgorithm