Practical Quantum Computing: Grover's algorithm

Jakub Tworzydlo

Institute of Theoretical Physics
Jakub.Tworzydlo@fuw.edu.pl

14/04/2026 Pasteura 5, Warszawa

Plan

1. Quantum cricuit model of computation
2. Quantum algorithm: formulation
3. Implementation of Grover's algorithm

Introduction

Sources of quantum algorithms:

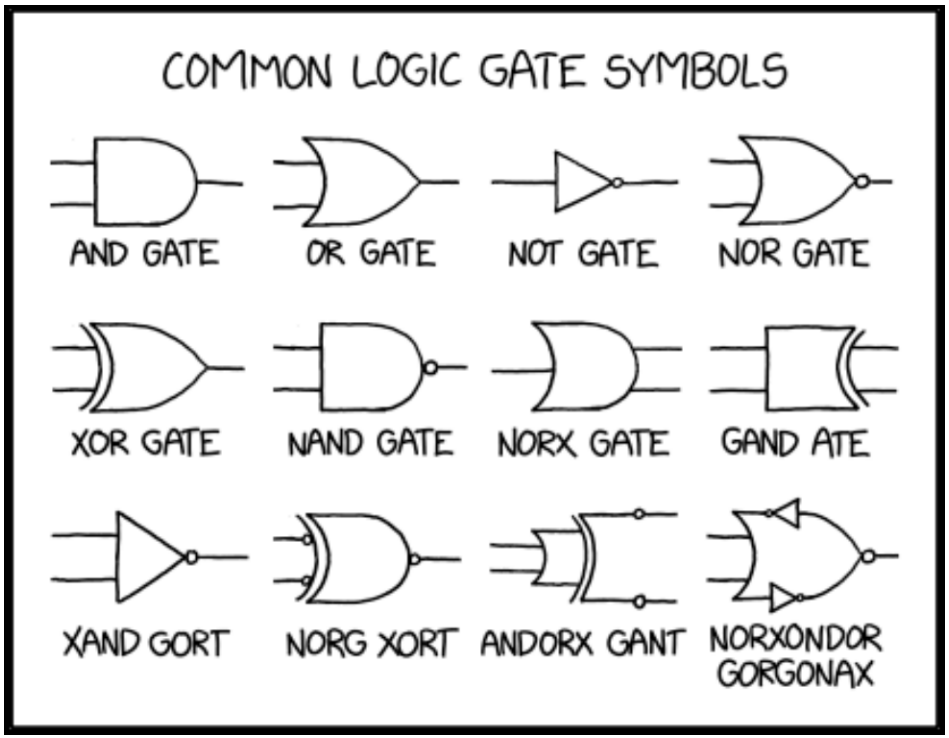
- **Search algorithms** — e.g., Grover's algorithm
- **Fourier-Transform-Based algorithms** — e.g., Shor's algorithm, Quantum Phase Estimation
- **Simulation algorithms** — e.g., quantum simulation of physical systems

Plan

1. Quantum cricuit model of computation
2. Quantum algorithm: formulation
3. Implementation of Grover's algorithm

Classical circuits

- Turing machine is equivalent to some digital circuit (with memory)
- classcial digital circuit consists of: wires, connectors, ancillas, logic gates (`NOT`, `OR`, `AND`, `XOR`)
- `NAND` gate is universal, but not reversible
- `CNOT` gate is reversible, but not universal



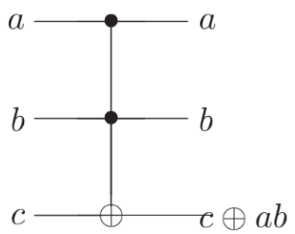
Classical reversible circuits

- every classical digital circuit can be run as a reversible circuit (with ancillas)
- universal reversible computation needs: reversible `NAND`, `FANOUT`
- Toffoli gate is universal

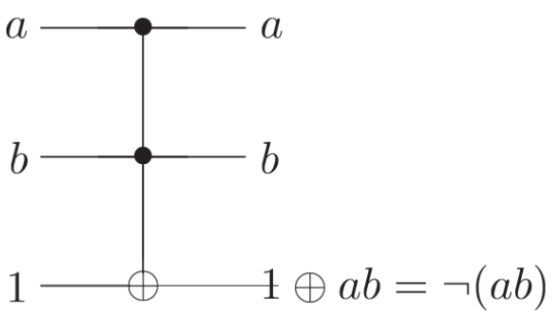
Reversible gates

Toffoli

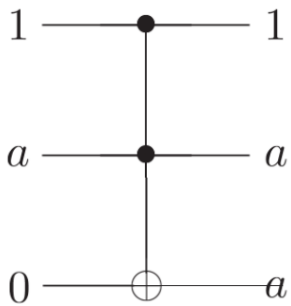
Inputs			Outputs		
a	b	c	a'	b'	c'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0



NAND

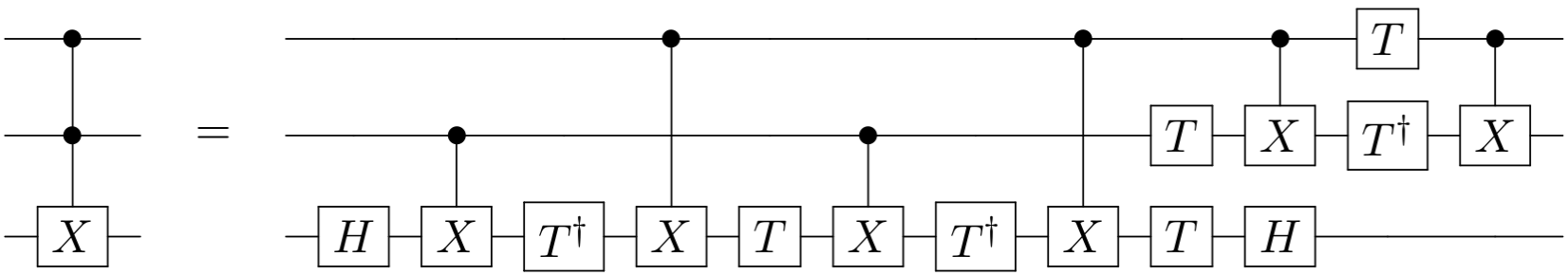


FANOUT



Quantum Toffoli gate

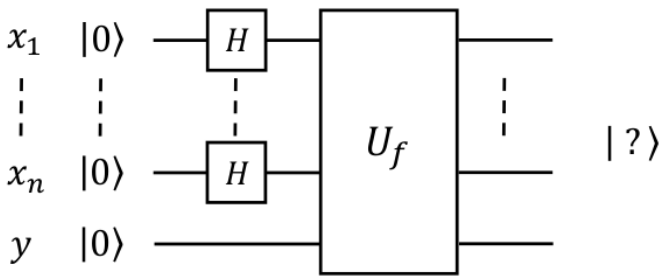
- classical C-NOT gate has a quantum counterpart *CX*;
  CC-NOT (Toffoli) is quantum *CCX*
- *CCX* is a complex gate (provided by Qiskit)
- quantum Toffoli can be build with *CX* gates, which is not possible classically (derivation in *Nielsen & Chaung chap. II.4.3*)



Quantum parallel computation

Let $U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$, we can achieve quantum parallelism:

Quantum parallelism



$$|0\rangle^{\otimes n} \otimes |0\rangle \xrightarrow{H^{\otimes n} \otimes \mathbb{I}} \left[\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right]^{\otimes n} \otimes |0\rangle$$

"the most democratic state" $|s\rangle$ - superposition of all possible input states

$$= \left(\frac{1}{\sqrt{2}}\right)^n \left(\sum_{x_1,\ldots,x_n=0}^{1} |x_1, \ldots, x_n\rangle\right) \otimes |0\rangle$$

$$\xrightarrow{U_f} \left(\frac{1}{\sqrt{2}}\right)^n \sum_{x_1,\ldots,x_n=0}^{1} |x_1, \ldots, x_n\rangle \otimes |f(x_1, \ldots, x_n)\rangle$$

superposition of function values for all possible inputs, with a single evaluation of U_f!

But what now? If we just measure the resulting state,

the output will be a randomly drawn input and its function value.

Plan

1. Quantum cricuit model of computation
2. Quantum algorithm: formulation
3. Implementation of Grover's algorithm

Database search

- $N = 2^n$ objects, labeled $x \in \{0, 1, 2, \ldots, 2^n - 1\}$; one marked item w (the winner)
- the oracle: black box function

$$f(x) = 0 \quad \text{for} \quad x \neq w$$
$$f(w) = 1$$

- "inverting a function": find a unque input that satisfies the constrain

Classical complexity: "brute force" search in $\mathcal{O}(N)$ time

Quantum oracle

All operations: $*$ are unitary $*$ performed on a quantum register $|x\rangle$

- imagine we have U_w, which implements our $f(x)$

$$U_w |x\rangle = -|x\rangle \text{ for } x = w$$
$$U_w |x\rangle = |x\rangle \quad \text{for } x \neq w$$

- construction... check it works(!)

$$U_w = \mathbb{I} - 2|w\rangle\langle w|$$

- time of computation: number of U_w operations

Grover operator

- amplitude amplification operator – useful trick
- ”most democratic” state $|s\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$
- unitary Grover operator

$$U_s = 2 |s\rangle \langle s| - \mathbb{I}$$

simple calculation gives $| \langle w|s\rangle |^2 = \frac{1}{N}$, $| \langle w|U_s U_w|s\rangle |^2 = \frac{1}{N}(3 - \frac{4}{N})^2$

with more tricky algebra

$$| \langle w| (U_s U_w)^t |s\rangle |^2 = \sin^2((2t+1)\theta),$$

where $\theta = \arcsin \frac{1}{\sqrt{N}}$.

Derivation

ALGEBRAIC DERIVATION OF Grover's algorithm

$$U_\omega = \mathbb{1} - 2\,|\omega\rangle\langle\omega| \qquad U_s = 2\,|s\rangle\langle s| - \mathbb{1}$$

with $|s\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$; where $N = 2^n$; $x = 0, 1 \dots, 2^{n-1}$

check how U_ω, U_s acts on these states:

$$U_\omega |s\rangle = |s\rangle - 2\,|\omega\rangle \underbrace{\langle\omega|s\rangle}_{1/\sqrt{N}} = 1\cdot|s\rangle - \frac{2}{\sqrt{N}}\,|\omega\rangle$$

$$U_\omega |\omega\rangle = -|\omega\rangle$$

$$U_s |s\rangle = |s\rangle$$

$$U_s |\omega\rangle = \tfrac{2}{\sqrt{N}}|s\rangle - |\omega\rangle$$

we can express U_ω, U_s in basis $\mathrm{span}(|s\rangle, |\omega\rangle)$

Derivation continued

$$U_\omega \xrightarrow{\text{equiv.}} \begin{pmatrix} 1 & 0 \\ -\frac{2}{\sqrt{N}} & -1 \end{pmatrix} \qquad U_s \xrightarrow{\text{equiv.}} \begin{pmatrix} 1 & 2/\sqrt{N} \\ 0 & -1 \end{pmatrix}$$

using this matrix representation:

$$U_s U_\omega |s\rangle = \left(1 - \frac{4}{N}\right)|s\rangle + \frac{2}{\sqrt{N}}|\omega\rangle$$

$$\left|\langle \omega | U_s U_\omega | s\rangle\right|^2 = \frac{1}{N}\left(3 - 4/N\right)^2$$

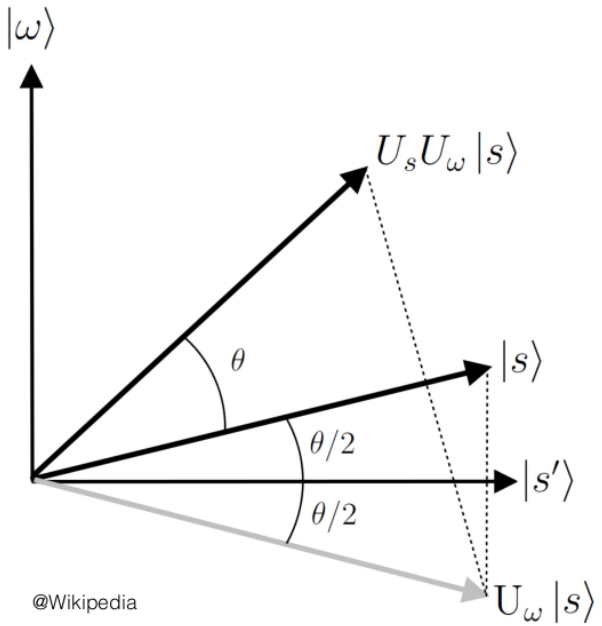
to calculate $(U_s U_\omega)^t$ (after t-iterations) one needs to diagonalize $(U_s U_\omega)$; ... result is

$$\left|\langle \omega | (U_s U_\omega)^t | s\rangle\right|^2 = \sin^2\left(\theta(2t+1)\right)$$

$$\theta = \operatorname{arc}\sin 1/\sqrt{N}$$

Geometric interpretation

Geometric intuition for how it works:



The n-qubit system starts in state $|s\rangle$. It lives in a subspace spanned by $|\omega\rangle$ and $|s'\rangle = \frac{1}{\sqrt{N-1}} \sum_{x \neq w} |x\rangle$: $|s\rangle = \frac{1}{\sqrt{N}} |\omega\rangle + \frac{\sqrt{N-1}}{\sqrt{N}} |s'\rangle$.

It can be verified that the application of the oracle U_ω and the diffusion operator $U_s = 2|s\rangle\langle s| - \mathbb{I}$ rotates the n-qubit state within the subspace by an angle $\theta = 2 \arcsin \frac{1}{\sqrt{N}}$.

It follows that after approximately $\pi\sqrt{N}/4$ iterations of $U_s U_\omega$, the state is rotated *close* to $|\omega\rangle$, which is then likely to be measured and thus determined.

Therefore, the algorithm has complexity $\mathcal{O}(\sqrt{N})$, quadratically improving the classical algorithm. It has, however, a non-unit probability of success, with the error probability approaching 0 with growing N.

Grover's algorithm (1997)

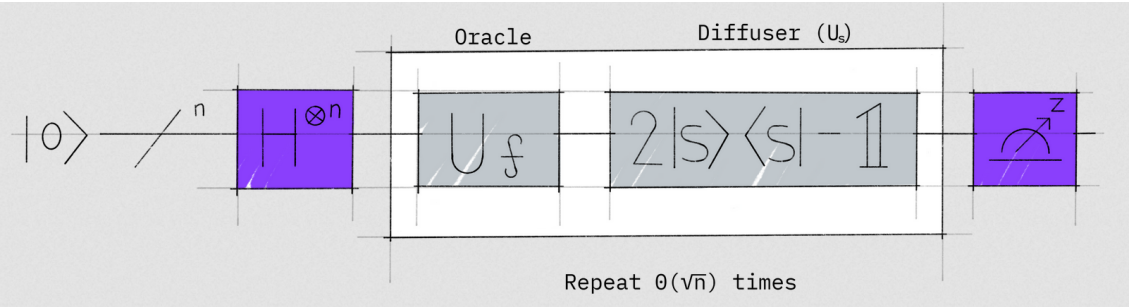
- maximal overalp is attained for $t^* \propto \pi\sqrt{N}/4$
- quadratic speedup from quantum computation: $\mathcal{O}(\sqrt{N})$ versus $\mathcal{O}(N)$
- Grover's algorithm is optimal (no more speedup possible)
- no new complexity classification, no solution to NP-complete problems

Final remarks:
"Is quantum search practical?" explains misunderstandings and perspectives
https://arxiv.org/abs/quant-ph/0405001
also good textbook reference: Nielsen & Chuang chap. II.6

Plan

1. Quantum cricuit model of computation
2. Quantum algorithm: formulation
3. Implementation of Grover's algorithm

Building circuit for Grover's search



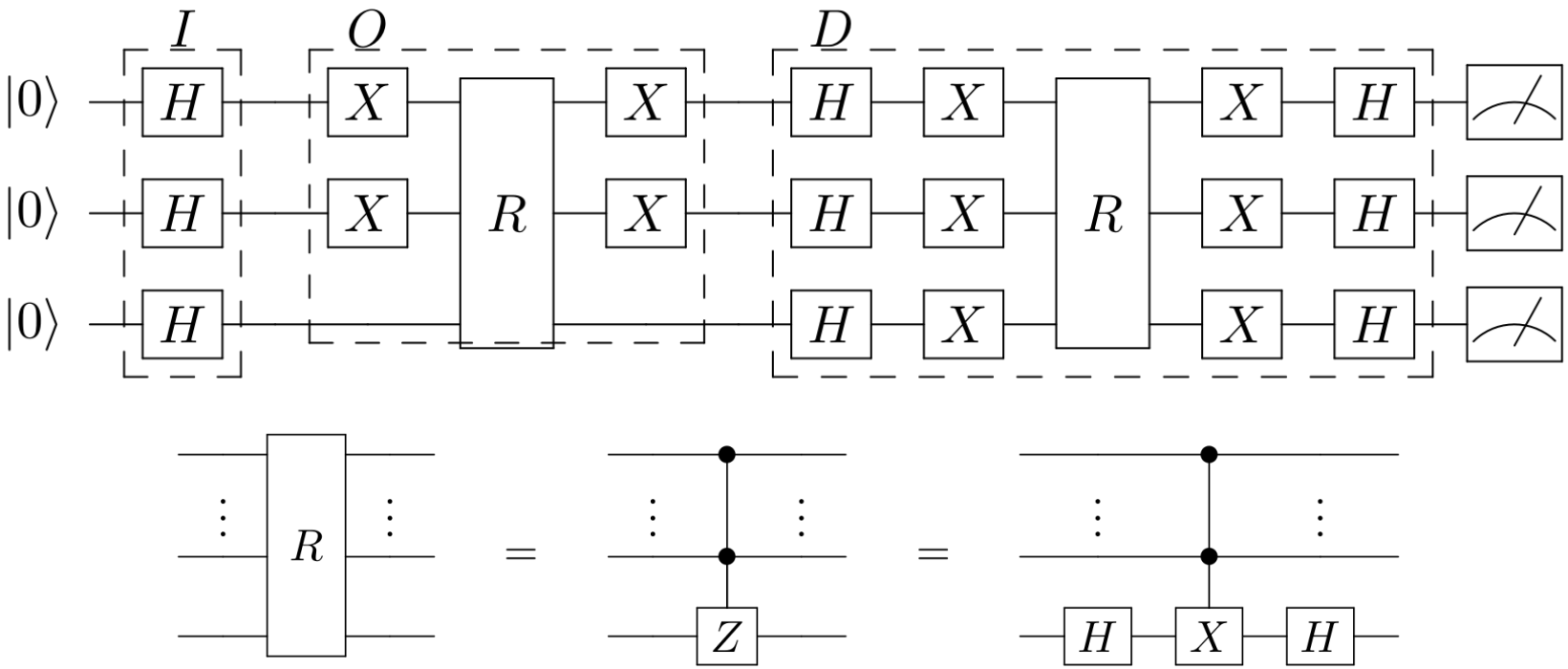
Oracle

- we need $V_w = X^{1-w_0} \otimes \ldots \otimes X^{1-w_{N-1}}$, which flips a qubit only when $w_i = 0$
- $[CCZ]$ gate flips the sign for $w = [1, 1, 1]$ (all qubits are 1)
- the oracle is then $U_w = V_w \, [CCZ] \, V_w$

Grover operator

- same for $X^{(n)} \, [CCZ] \, X^{(n)}$ – flips sign only for $|0\rangle$
- we express $X^{(n)} \, [CCZ] \, X^{(n)} = \mathbb{I} - 2 \, |0\rangle \, \langle 0|$
- finally: $H^{(n)} X^{(n)} \, [CCZ] \, X^{(n)} H^{(n)} = \mathbb{I} - 2 \, |s\rangle \, \langle s| = -U_s$

Final diagram



- $n = 3$ qubits, marked state $w = [0, 0, 1]$
- one iteration: initialization, oracle, Grover operator (diffusor D)
- note that the core of calculation is done by Toffoli gate
- measurement in computational basis

Historic data

Grover's search algorithm run on IBM quantum devices back in 2019 (by Alicja Dutkiewicz)

Tabela 4.2: Prawdopodobieństwa sukcesu

Liczba kubitów	Liczba iteracji	Średnie prawdopodobieńswto sukcesu [%]			
		Klasycznie	Teoretycznie	*IBM Q Tenerife*	*IBM Q Yorktown*
2	1	25	100	83.074 ± 0.017	84.713 ± 0.063
3	1	12.5	78.1	41.416 ± 0.017	19.617 ± 0.031
	2	25	94.5	29.873 ± 0.014	29.785 ± 0.038
4	1	6.25	47.3	6.8831 ± 0.0060	5.667 ± 0.018
			Po preselekcji	6.7354 ± 0.0075	6.293 ± 0.026
	2	12.5	90.8	6.1130 ± 0.0061	6.305 ± 0.019
			Po preselekcji	6.0601 ± 0.0082	6.299 ± 0.026
	3	25	96.1	5.6970 ± 0.0048	6.038 ± 0.019
			Po preselekcji	5.670 ± 0.007	6.005 ± 0.025

Kolorem zielonym oznaczono wyniki lepsze, a kolorem czerwonym gorsze niż najlepszy algorytm klasyczny.