The Logic in Computer Science Column

by

Yuri Gurevich

Computer Science and Engineering
University of Michigan, Ann Arbor, MI 48109, USA
gurevich@umich.edu

AN INVITATION TO QUANTUM COMPUTING

Andreas Blass
Mathematics Dept, University of Michigan
Ann Arbor, MI 48109, USA, ablass@umich.edu

Yuri Gurevich
Computer Science & Engineering, University of Michigan;
Dept of Math Sciences, Stevens Institute of Technology

Abstract

When a computer scientist attempts to understand quantum computing, he may stumble over the physics that seems to be a prerequisite. As a result, the attempt may be abandoned. This little pedagogical essay is aimed to help with this problem. We present the specific example of Grover's search algorithm, but we put computation first and postpone physics.

1 Introductory dialog

Quisani[1]**:** I expected to see both of you.

Author[2]**:** Let's start. Andreas will be here shortly.

Q: How good is your physics?

A: Don't ask. In my student years I was interested in many subjects but physics wasn't one of them. Andreas, on the other hand, has his Bachelor degree in physics. Both of his parents were physicists.

Q: Do you know, say, relativity theory or, at least, special relativity theory?

[1] A former student of the second author.

[2] Just the second author in this section but both authors, speaking one at a time, in the later sections.

A: As a school boy, I read popular expositions of relativity theory but could not understand it. The more I read, the more confusing it seemed. In my working-class suburb of an industrial Soviet city there was nobody to ask. Once a visiting lecturer spoke about relativity theory. For a while, things seemed clear. Then he mentioned a Gedankenexperiment where a spaceship leaves the planet Earth. After traveling around for one year, the spaceship returns to Earth where 100 years passed. I asked him why it isn't the other way round. We can imagine that it is Earth that went for a voyage and so more time should have passed on the spaceship. He answered something, but I did not understand a thing. A couple of years later, already at a university, I came across a book on Riemannian geometry with a chapter on special relativity [8, §IV]. It was simple and beautiful.

Q: What about quantum physics?

A: Unfortunately I knew nothing about it until a few years ago.

Q: I'd love to understand quantum computing. But I haven't studied quantum physics, and I forgot whatever physics I did study. Quantum computing seems daunting to a computer scientist, like myself, because of all that physics. Is physics necessary for quantum computing?

A: Yes and no. Let me try to explain this by analogy. Is engineering necessary for to computer science?

Yes, to a large extent computer science deals with engineering artifacts, like operating systems. More than once I saw how clever engineering circumvents difficult mathematical problems. Without engineering, computer science is incomplete. In my view, it is an integral part of a more complete discipline which is computer science and engineering. Yet, one can make a considerable advance in computer science with little exposure to engineering. Examples, starting from the Church and Turing theses, are abundant.

Quantum computing is computing with quantum computers whose very existence depends first of all on quantum physics (and other disciplines including engineering). Quantum physics is indispensible for quantum computing. Yet, one can make a considerable advance in quantum computing with little knowledge of quantum physics. One certainly can understand much of the existing literature on quantum computing with little knowledge of quantum physics.

Q: I am a bit sceptical about your last point. I tried to do exactly that and failed.

A: To illustrate this point, we will explain to you Grover's algorithm [5], the second most famous quantum algorithm. We will presuppose no knowledge of quantum physics.

2 A function inversion problem, or looking for a needle in a haystack

Oracle Search Problem Given access to a Boolean oracle $f : \{0,1\}^n \to \{0,1\}$, solve the equation $fx = 1$.

To simplify the exposition, we restrict attention to the case where the equation $fx = 1$ has a unique solution.

Example Given access to a huge phone directory, sorted by name, find the name of a person by her phone number.

Let $N = 2^n$. It seems obvious that, on average, any algorithm solving the problem has to query the oracle $\geq \frac{1}{2}N$ times. Yet, Grover's algorithm solves the problem with high probability and only $\left\lfloor (\pi/4)\sqrt{N} \right\rfloor$ queries. How does it do that?

A quantum algorithm is an algorithm for quantum hardware. Let's separate the concerns. First, let's understand Grover's algorithm itself, how it works and how it gets away with relatively few queries. Then we'll consider how an ideal quantum computer executes the algorithm.

3 Preliminaries

Recall that Euclidean spaces are finite dimensional vector spaces over the field $\mathbb{R}$ of real numbers endowed with an inner product structure. Finite-dimensional Hilbert spaces are finite-dimensional vector spaces over the field $\mathbb{C}$ of complex numbers endowed with an inner product structure. There are also infinite-dimensional Hilbert spaces but we are not considering them here; our Hilbert spaces are by default finite-dimensional.

3.1 Bra-ket notation

Physicists use a clever *braket* notation introduced by Paul Adrien Maurice Dirac in 1939 [3]. The inner product of vectors x, y of a given Hilbert space is denoted $\langle x \,|\, y \rangle$ and is viewed as an application of a linear functional $\langle x|$, called "bra-x", to a vector $|y\rangle$, called "ket-y". You can write inside the ket symbol any convenient description of the intended vector, e.g, |the first special vector⟩.

As in Euclidean spaces, the length of a vector $|x\rangle$ is $\sqrt{\langle x \,|\, x \rangle}$.

Q: Wait, I played only with inner product over $\mathbb{R}$. Normally, I think of the inner product of two vector $\vec{x}$ and $\vec{y}$ as $|\vec{x}| \cdot |\vec{y}| \cdot \cos\theta$ where θ is the angle between $\vec{x}$ and $\vec{y}$, but I am aware of the axiomatic definition of inner product over $\mathbb{R}$. Is the inner product structure over $\mathbb{C}$ much different? Can $\langle x \,|\, y\rangle$ be a complex number that is not real?

A: Over $\mathbb{C}$, the inner product $\langle x \,|\, y\rangle$ can take any complex value. One property is *positive definiteness*: $\langle x \,|\, x\rangle \geq 0$ and $\langle x \,|\, x\rangle = 0$ if and only if $x = 0$. Another property is *conjugate symmetry*: $\langle y \,|\, x\rangle$ is the complex conjugate of $\langle x \,|\, y\rangle$. In a special case when $\langle x \,|\, y\rangle$ is real, we have $\langle y \,|\, x\rangle = \langle x \,|\, y\rangle$, as in real Euclidean spaces.

The final property is *linearity* in one of the arguments. In mathematics, it is typically linearity in the first argument. In physics, it is always linearity in the second argument: $\langle x \,|\, y+z\rangle = \langle x \,|\, y\rangle + \langle x \,|\, z\rangle$ and $\langle x \,|\, cy\rangle = c\langle x \,|\, y\rangle$ for any complex number c. If forced to choose, we'll use the physicist version.

Finally, if θ is the angle between vectors $|x\rangle$, $|y\rangle$, then $\cos\theta$ is the real part of $\dfrac{\langle x \,|\, y\rangle}{\||x\rangle| \cdot \||y\rangle|}$.

3.2 Tensor product

Let $\mathcal{H}_1, \mathcal{H}_2$ be Hilbert spaces with fixed orthonormal bases $|0\rangle, \ldots, |m-1\rangle$ and $|0'\rangle, \ldots, |(n-1)'\rangle$ respectively.

The *tensor product* $\mathcal{H}_1 \otimes \mathcal{H}_2$ of $\mathcal{H}_1$ and $\mathcal{H}_2$ is an $(m \times n)$-dimensional Hilbert space. Each pair $|i\rangle, |j'\rangle$ of basis vectors from $\mathcal{H}_1, \mathcal{H}_2$ respectively, gives rise to a ket vector in $\mathcal{H}_1 \otimes \mathcal{H}_2$ denoted $|i\rangle \otimes |j'\rangle$ or simply $|ij'\rangle$; the corresponding bra vector, i.e., the corresponding linear functional is $\langle i| \otimes \langle j'|$ or simply $\langle ij'|$. On these $m \times n$ ket vectors, the inner product is given by the formula

$$\langle ij' \,|\, k\ell'\rangle = \langle i \,|\, k\rangle \cdot \langle j' \,|\, \ell'\rangle = \begin{cases} 1 & \text{if } (i,j) = (k,\ell), \\ 0 & \text{otherwise,} \end{cases}$$

so that the $m \times n$ ket vectors $|i\rangle \otimes |j'\rangle$ form an orthonormal basis of $\mathcal{H}_1 \otimes \mathcal{H}_2$, and an arbitrary vector in $\mathcal{H}_1 \otimes \mathcal{H}_2$ has the form $\sum a_{ij}|ij'\rangle$. The inner product extends to the whole $\mathcal{H}_1 \otimes \mathcal{H}_2$ by linearity:

$$\left(\sum_{i,j} a_{ij}\langle ij'|\right)\left(\sum_{k,\ell} b_{k\ell}|k\ell'\rangle\right) = \sum_{i,j} a_{ij}b_{ij}.$$

If $|v_1\rangle = \sum_i a_i|i\rangle \in \mathcal{H}_1$ and $|v_2\rangle = \sum_j b_j|j'\rangle \in \mathcal{H}_2$, define $|v_1\rangle \otimes |v_2\rangle = \sum_{i,j} a_i b_j |i\,j'\rangle$. Vectors in $\mathcal{H}_1 \otimes \mathcal{H}_2$ obtained this way are *pure tensors*. Not every vector in $\mathcal{H}_1 \otimes \mathcal{H}_2$ is a pure tensor unless $m \leq 1$ or $n \leq 1$. For example, $|00'\rangle + |11'\rangle$ is not a pure tensor. Indeed, if

$$|00'\rangle + |11'\rangle = \sum a_i|i\rangle \otimes \sum b_j|j'\rangle = \sum a_i b_j |i\,j'\rangle,$$

then $a_0 b_0 = a_1 b_1 = 1$ and therefore $a_0 b_1 \neq 0$ but the representation $|00'\rangle + |11'\rangle$ of the vector shows that $a_0 b_1 = 0$.

3.3 Standard bases

The definition of the tensor product does not require that $\mathcal{H}_1, \mathcal{H}_2$ be disjoint; they could even be equal.

Elements of the two-dimensional Hilbert space $\mathbb{C}^2$ are pairs (x_0, x_1) of complex numbers. The ket vectors

$$|0\rangle = (1, 0),$$
$$|1\rangle = (0, 1).$$

form the *standard orthonormal basis* of $\mathbb{C}^2$. The *standard orthonormal basis* of $(\mathbb{C}^2)^{\otimes 2} = \mathbb{C}^2 \otimes \mathbb{C}^2$ comprises four vectors $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$. More generally, the *standard orthonormal basis* of the tensor product $(\mathbb{C}^2)^{\otimes k}$ of k copies of $\mathbb{C}^2$ comprises vectors $|x\rangle$ where x ranges over the set $\{0, 1\}^k$ of binary strings of length k. Note that the dimension of Hilbert space $(\mathbb{C}^2)^{\otimes k}$ is 2^k.

The following trivial observation will be very useful below. To define a linear operator on a vector space with a given basis, it suffices to define it on the basis vectors. In particular, to define a linear operator on $(\mathbb{C}^2)^{\otimes k}$, it suffices to define it on vectors $|x\rangle$ for every $x \in \{0, 1\}^k$.

> **Q:** Everything you've said so far makes sense if the scalars are real rather than complex numbers.
>
> **A:** This is true. In fact, Grover's algorithm itself makes good sense and even is easier to think about when the scalars are real. But quantum computers work with complex numbers, and so we stick to complex scalars.

4 Grover's algorithm

Let x range over $\{0,1\}^n$ and b range over $\{0,1\}$. If $x = (x_1, \ldots, x_n) \in \{0,1\}^n$, let $|x\rangle = |x_1\rangle \otimes \cdots \otimes |x_n\rangle \in (\mathbb{C}^2)^{\otimes n}$. Then $|x\rangle \otimes |b\rangle \in (\mathbb{C}^2)^{\otimes(n+1)}$.

4.1 Classical to quantum oracle

In the Oracle Search Problem, we are given a Boolean oracle $f : \{0,1\}^n \to \{0,1\}$. The oracle f gives rise to a reversible, i.e., bijective, transformation

$$F(x,b) = (x, b \oplus fx) = \begin{cases} (x, b \oplus 1) & \text{if } fx = 1 \\ (x,b) & \text{if } fx = 0 \end{cases}$$

of the set $\{0,1\}^{n+1}$. Here and below, $\oplus$ is addition modulo 2. By linearity, F gives rise to a unique linear operator U on $(\mathbb{C}^2)^{\otimes(n+1)}$ such that

$$U\Big(|x \otimes b\rangle\Big) = |x\rangle \otimes |b \oplus fx\rangle.$$

U is a unitary operator, i.e., it preserves the inner product. Therefore it preserves the distances between vectors, it preserves the length of a vector, and it leaves the zero vector in place.

> **Q:** What is the point of all this transformation from f to F and then to U? What is wrong with f to begin with?
>
> **A:** The reason is physics. Quantum systems are transformed by unitary operators. Although f gives rise to a linear transformation from $(\mathbb{C}^2)^{\otimes n}$ to $\mathbb{C}^2$, this transformation is not unitary.
>
> **Q:** Of course! It can't be unitary because it maps into a lower-dimensional space.
>
> **A:** Indeed, in general, if a linear transformation maps one orthonormal basis to another, then it is unitary if and only if it is a bijection on the bases.

4.2 Hadamard operator H

The unit vectors $(|0\rangle + |1\rangle)/\sqrt{2}$ and $(|0\rangle - |1\rangle)/\sqrt{2}$ in $\mathbb{C}^2$ are known as $|+\rangle$ and $|-\rangle$ respectively. They constitute another orthonormal basis for $\mathbb{C}^2$ that is often useful.

The Hadamard operator H is the unitary transformation from the basis $\{|0\rangle, |1\rangle\}$ to the basis $\{|+\rangle, |-\rangle\}$ such that $H|0\rangle = |+\rangle$ and $H|1\rangle = |-\rangle$. It is easy to check that $H|+\rangle = |0\rangle$ and $H|-\rangle = |1\rangle$, so that H^2 is the identity operator on $\mathbb{C}^2$.

Operator $H \otimes H$ on $\mathbb{C}^2 \otimes \mathbb{C}^2$ applies H to each of the two qubits. For example, $H^{\otimes 2}|0\rangle|1\rangle = |+\rangle|-\rangle$. Generally, an operator $H^{\otimes k}$ on $(\mathbb{C}^2)^{\otimes k}$ applies H to each of the k qubits. If $x = (x_1, \ldots, x_k) \in \{0,1\}^k$ then $H^{\otimes k}(|x_1\rangle \otimes \cdots \otimes |x_k\rangle) = H|x_1\rangle \otimes \cdots \otimes H|x_k\rangle$.

4.3 Operator V

The bilinearity of $\otimes$ has a possibly surprising effect. Recall that F leaves the first n bits unchanged and modifies only the last bit. Its linearization U thus acts as the identity on the first n tensor factors and modifies only the last factor. Nevertheless, in some contexts, U can also be regarded as modifying only the first n factors. Since $|-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$, we have

$$U\Big(|x\rangle \otimes |-\rangle\Big) = \begin{cases} |x\rangle \otimes (|1\rangle - |0\rangle)/\sqrt{2} & \text{if } fx = 1 \\ |x\rangle \otimes (|0\rangle - |1\rangle)/\sqrt{2} & \text{if } fx = 0 \end{cases}$$
$$= |x\rangle \otimes (-1)^{fx}|-\rangle = (-1)^{fx}|x\rangle \otimes |-\rangle.$$

Let $V|x\rangle = (-1)^{fx}|x\rangle$. Then

$$U\Big(|x\rangle \otimes |-\rangle\Big) = \Big(V|x\rangle\Big) \otimes |-\rangle.$$

To compute $V|x\rangle$, apply U to $|x\rangle \otimes |-\rangle$ and ignore the final $|-\rangle$.

4.4 A real plane of interest

Recall that x ranges over $\{0,1\}^n$ and $N = 2^n$. Let s be the unique solution of the equation $fx = 1$. Consider unit vectors

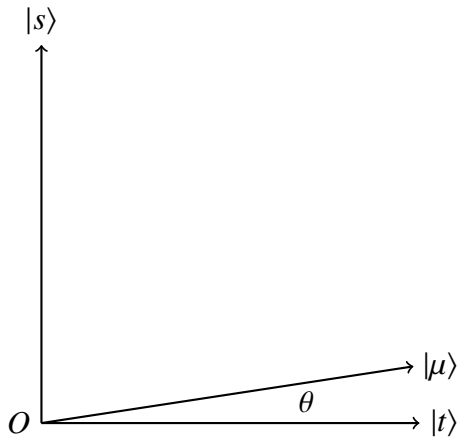
$$|t\rangle = \frac{\sum_{x \neq s} |x\rangle}{\sqrt{N-1}} \quad \text{and} \quad |\mu\rangle = \frac{\sum_x |x\rangle}{\sqrt{N}} = \frac{\sqrt{N-1}}{\sqrt{N}}|t\rangle + \frac{1}{\sqrt{N}}|s\rangle.$$

Q: Wait, that ket notation has been bothering me for a while. But now it becomes really confusing. On the one hand, the notation $|x\rangle$ turns a bitstring x into a vector $|x\rangle$. On the other hand, you keep using the ket notation with things other than bitstrings inside. Earlier you used $|+\rangle$ and $|-\rangle$, and now you are using $|t\rangle$ and $|\mu\rangle$.

A: We agree that the notation may seem ambiguous. We follow the physics tradition mentioned above: inside the ket, you can write anything that conveniently identifies the intended vector.

Q: Convenience is in the eye of the beholder.

Vector $|t\rangle$ is orthogonal to $|s\rangle$ because $|x\rangle$ is orthogonal to $|s\rangle$ for every x in $\{0,1\}^n$ which is different from s. Further, $|\mu\rangle$ is the mean of all the vectors $|x\rangle$, and $V|\mu\rangle = \frac{\sqrt{N-1}}{\sqrt{N}}|t\rangle - \frac{1}{\sqrt{N}}|s\rangle$. Let Π be the real plane spanned by $|s\rangle$ and $|t\rangle$. Notice that $|\mu\rangle$ makes a small angle θ with $|t\rangle$ toward $|s\rangle$.

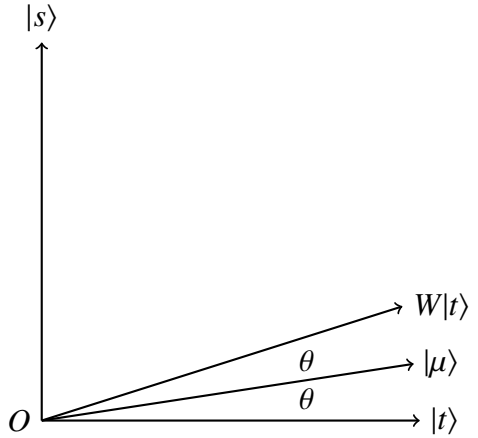


We have $\theta \approx \sin\theta = \cos(\frac{\pi}{2} - \theta) = \langle s \,|\, \mu\rangle = 1/\sqrt{N}$.

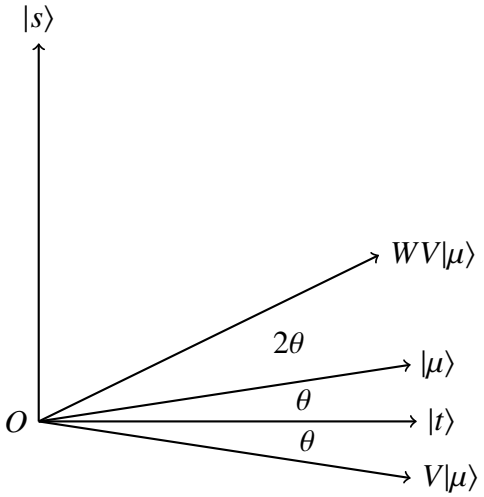
4.5 Reflections and rotations

Since $V|s\rangle = -|s\rangle$ and $V|x\rangle = |x\rangle$ for every $x \neq s$, V is the reflection of $(\mathbb{C}^2)^{\otimes n}$ in the hyperplane orthogonal to $|s\rangle$. On Π, V is a reflection in the line of $|t\rangle$.

Let W be the reflection of $(\mathbb{C}^2)^{\otimes n}$ in the line spanned by the mean vector $|\mu\rangle$, i.e., $W|\mu\rangle = |\mu\rangle$ and $W|v\rangle = -|v\rangle$ for every vector $|v\rangle$ orthogonal to $|\mu\rangle$. W is unitary. In particular, on Π, W is the reflection in the line of $|\mu\rangle$.



The unitary operator of particular interest to us is WV. $WV|t\rangle = W|t\rangle$, and so WV rotates $|t\rangle$ by 2θ around O. WV also rotates $|\mu\rangle$ by 2θ:



It follows that WV rotates the whole plane Π by 2θ around O.

4.6 Approximating the unique solution

$(WV)^k$ rotates Π by $2k\theta$, so that the angle between $|t\rangle$ and $(WV)^k|\mu\rangle$ is $(2k+1)\theta$. We would like to find an integer k such that $(WV)^k|\mu\rangle$ is closest to $|s\rangle$, i.e., $(2k+1)\theta$ is closest to $\pi/2$ and k is closest to $\frac{\pi}{4}\frac{1}{\theta} - \frac{1}{2}$. Recall that $\sin\theta = 1/\sqrt{N}$. Since $\sin x < x$ for $x > 0$ and $\sin x$ is very close to x when x is close to zero, we have

$$\frac{\pi}{4}\frac{1}{\theta} = \frac{\pi}{4}\frac{1}{\sin\theta}\frac{\sin\theta}{\theta} = \frac{\pi}{4}\sqrt{N}\cdot\frac{\sin\theta}{\theta} < \frac{\pi}{4}\sqrt{N},$$

and the interval $(\frac{\pi}{4}\frac{1}{\theta}, \frac{\pi}{4}\sqrt{N})$ is tiny for large values of N.

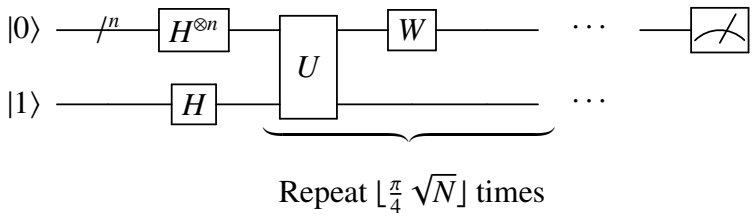
It is most probable (as N varies) that there are no integers in the interval and therefore $\left\lfloor (\pi/4)\sqrt{N} \right\rfloor$ is the integer closest to $\frac{\pi}{4}\frac{1}{\theta} - \frac{1}{2}$, i.e., $(WV)^{\lfloor (\pi/4)\sqrt{N} \rfloor}|\mu\rangle$ is closest to $|s\rangle$ among all vectors $(WV)^k|\mu\rangle$. If there is an integer in the interval, then $\left\lfloor (\pi/4)\sqrt{N} \right\rfloor - 1$ is the integer closest to $\frac{\pi}{4}\frac{1}{\theta} - \frac{1}{2}$, but the angle between $|s\rangle$ and $(WV)^{\lfloor (\pi/4)\sqrt{N} \rfloor}|\mu\rangle$ is very close to $\theta/2$. So $(WV)^{\lfloor (\pi/4)\sqrt{N} \rfloor}|\mu\rangle$ is always a good approximation to $|s\rangle$.

4.7 A diagram of Grover's algorithm

The operator WV works with the n qubits, but there is also the $(n+1)^{\text{st}}$ qubit which is needed to query the oracle U. Let I be the identity operator on $\mathbb{C}^2$. The operator $(WV)\otimes I$ operates as WV on the first n qubits and as I on the $(n+1)^{\text{st}}$ qubit. Notice that the mean vector $|\mu\rangle$ is a pure tensor, namely,

$$\begin{aligned}|\mu\rangle &= \frac{1}{\sqrt{N}}\sum_x |x\rangle = \frac{1}{2^{n/2}}\Big((|0\rangle + |1\rangle)\otimes\cdots\otimes(|0\rangle + |1\rangle)\Big)\\ &= |+\rangle^n = H^{\otimes n}|0^n\rangle.\end{aligned}$$

The following diagram depicts Grover's algorithm.



Here the sign $/^n$ indicates a bundle of n wires. The upper line is the time line of the bundle, and the lower line is the time line of the $(n+1)^{\text{st}}$ qubit. In this diagram (as well as in the diagrams below), time goes left to right.

The meter sign at the right end indicates measurement. You examine the final state $(WV)^{\lfloor (\pi/4)\sqrt{N} \rfloor}|\mu\rangle$ of the bundle and output the n-bit string s such that $|s\rangle$ is approximated by the final state.

5 Discussion about Grover's algorithm

Q: You said that Grover's algorithm solves the Oracle Search Problem with only $\left\lfloor (\pi/4)\sqrt{N} \right\rfloor$ queries, and that seemed puzzling, even impossible. But the algorithm employs a more informative oracle, be it U or V. A query $|\psi\rangle$ to V may involve many strings x in $\{0,1\}^n$, i.e., many queries to the original oracle f, and the reply $V|\psi\rangle$ involves information about f on all those queries x. E.g., the query $|\mu\rangle = \sum_x |x\rangle/\sqrt{N}$ involves all queries to f, and the reply $V|\mu\rangle$ involves all values of f. As a result, it does not seem so puzzling that the algorithm gets away with fewer queries. A question arises whether Grover's algorithm is optimal. Maybe one can do even better.

A: The question has been addressed, first by Charles Bennet, Ethan Bernstein, Gilles Brassard and Umesh Vazirani in 1997 [1] and more recently by Cătălin Dohotaru and Peter Høyer in [4]. It turns out that the communication complexity of Grover's algorithm is optimal as far as the number of oracle calls is concerned.

Q: On another issue, consider that bundle of n qubits in the diagram of Grover's algorithm. The states of the bundle are given by vectors in $(\mathbb{C}^2)^{\otimes n}$ which have exponential dimension. The mean state $|\mu\rangle$ is a pure vector, but there seems to be no reason to expect that every $(WV)^k|\mu\rangle$ is a pure vector. How do you work with exponential-size vectors?

A: That is why we need a quantum computer.

6 Enter quantum computer

6.1 State space of a quantum system

According to quantum theory, the state space of a quantum system Q is a Hilbert space $\mathcal{H}$. The states of Q are represented by nonzero vectors in $\mathcal{H}$, with collinear vectors representing the same state. Here collinearity of two vectors $|\psi_1\rangle$ and $|\psi_2\rangle$ means that $|\psi_2\rangle = c|\psi_1\rangle$ for some nonzero complex number c.

For brevity, instead of "the state represented by vector $|\psi\rangle$", we say "the state $|\psi\rangle$."

The simplest system that is genuinely quantum is known as a one-qubit system or simply a qubit. Its state space is $\mathbb{C}^2$. There are numerous physical implementations of a qubit, but here we abstract from the implementation.

If a quantum system Q consists of disjoint[3]quantum systems Q_1, Q_2 then the state space of Q is the tensor product $\mathcal{H}_1 \otimes \mathcal{H}_2$ of the state spaces of systems Q_1, Q_2 respectively.

It follows that the state space of a k-qubit quantum system is $(\mathbb{C}^2)^{\otimes k}$. Grover's algorithm deals only with such systems.

6.2 Quantum computer: a rough sketch

Think about some machine model of classical computing, possibly with an oracle. The memory is split into numerous cells. In the beginning of a computation, the memory is in some initial global state. The computation proceeds by applying some basic operations to the memory and possibly oracle calls. At the end, the output is read from special output cells of the final global state of the memory.

In a broad-brush outline, quantum computers typically operate in a similar way. The role of computer memory is played by some quantum system Q. Usually this quantum memory is a k-qubit system for some k. In the beginning, Q is in some initial state. The computation proceeds by applying some unitary operations, known as quantum gates, or possibly applying an oracle, a special sort of unitary operator, to Q, and at the end a subsystem of Q is measured.

6.3 A universal gate set

No finite collection of gates allows us to synthesize precisely every unitary operator even on $\mathbb{C}^2$, because there are uncountably many such operators. But there are finite gate sets $\mathcal{B}$ which are universal in the sense that, for any unitary operator A on any $(\mathbb{C}^2)^{\otimes k}$, an arbitrarily close approximation can be synthesized.

> **Q:** What does "can be synthesized" mean here?
>
> **A:** A good question. By linearity, it suffices to restrict attention to how A works on the basis vectors of $(\mathbb{C}^2)^{\otimes k}$.
>
> We say that a unitary k-qubit operator A can be synthesized over $\mathcal{B}$ if there is a composition A' of $\mathcal{B}$-gates which works as follows. Given (i) input qubits $|b_1\rangle,\ldots,|b_k\rangle$ of A and (ii) optionally some auxiliary qubits $|a_1\rangle,\ldots,|a_j\rangle$, set to $|0\rangle$ or $|1\rangle$, A' outputs $|b_1\rangle,\ldots,|b_k\rangle$, $A(|b_1\rangle,\ldots,|b_k\rangle)$ and possibly some extra qubits $|g_1\rangle,\ldots,|g_l\rangle$. The auxiliary qubits $|a_1\rangle,\ldots,|a_j\rangle$ are known as *ancillas*. The extra $|g_1\rangle,\ldots,|g_l\rangle$ qubits sometimes are called the garbage qubits.

[3]For the cognoscenti: We assume here that the two physical systems are not identical. If they were identical, we would need the symmetric or anti-symmetric part of the tensor product.

We will use a universal set $\{H, \mathit{Toffoli}\}$ of only two gates acting on arbitrary qubits [9]. Here H is the one-qubit Hadamard gate mentioned above, and *Toffoli* is a three-qubit gate, named after Tommaso Toffoli who introduced it [10], and known also as the double-controlled Not.

The *Toffoli* gate applies to three distinct qubits. Two of them work as *controls* and the third one works as the *target*. By linearity, it suffices to define *Toffoli* on the basis vectors $|c_1\rangle|c_2\rangle|t\rangle$ of $\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2$ where $|c_1\rangle$,$|c_2\rangle$ are controls and $|t\rangle$ is the target. If both controls $|c_1\rangle$ and $|c_1\rangle$ are $|1\rangle$ then the target $|t\rangle$ is flipped, i.e., the new value is $|t \oplus 1\rangle$ where, as before, $\oplus$ is addition modulo 2. If at least one of the two controls is $|0\rangle$ then the target remains unchanged. In both cases, all qubits, except the target, remain unchanged.

Q: Let me understand this. How does *Toffoli* work on $(C^2)^{\otimes 4}$ when the first and fourth qubits are the controls and the second qubit is the target?

A: In this case, the result of applying *Toffoli* to state $|b_1\rangle|b_2\rangle|b_3\rangle|b_4\rangle$ is

$$\begin{cases} |b_1\rangle|b_2 \oplus 1\rangle|b_3\rangle|b_4\rangle & \text{if } |b_1\rangle = |b_4\rangle = |1\rangle, \\ |b_1\rangle|b_2\rangle|b_3\rangle|b_4\rangle & \text{if } |b_1\rangle = |0\rangle \text{ or } |b_4\rangle = |0\rangle. \end{cases}$$

6.4 Some auxiliary quantum gates

We introduce a few auxiliary quantum gates that will play a role in Grover's algorithm, and we show how to synthesize them in our gate basis.

Gate X

Define the one-qubit operator X by the equations

$$X|0\rangle = |1\rangle, \quad \text{and} \quad X|1\rangle = |0\rangle.$$

Unsurprisingly this gate is also called Not because it interchanges the two "truth values" $|0\rangle$ and $|1\rangle$. Using two ancillas, we can express X by means of *Toffoli*.

$$|1\rangle|1\rangle X|\psi\rangle = \mathit{Toffoli}\Big(|1\rangle|1\rangle|\psi\rangle\Big).$$

This equation is obvious when $|\psi\rangle$ is a standard basis vector but, by linearity, it holds also for arbitrary $|\psi\rangle$.

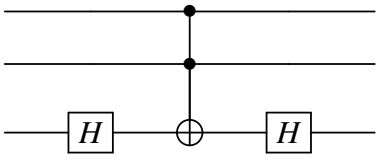
Gate Z

Define $Z = HXH$. We have

$$Z|0\rangle = HXH|0\rangle = HX(|0\rangle + |1\rangle)/\sqrt{2} = H(|1\rangle + |0\rangle)/\sqrt{2} = |0\rangle,$$
$$Z|1\rangle = HXH|1\rangle = HX(|0\rangle - |1\rangle)/\sqrt{2} = H(|1\rangle - |0\rangle)/\sqrt{2} = -|1\rangle.$$

Double-controlled Z

The following diagram illustrates the synthesis of double-controlled Z.



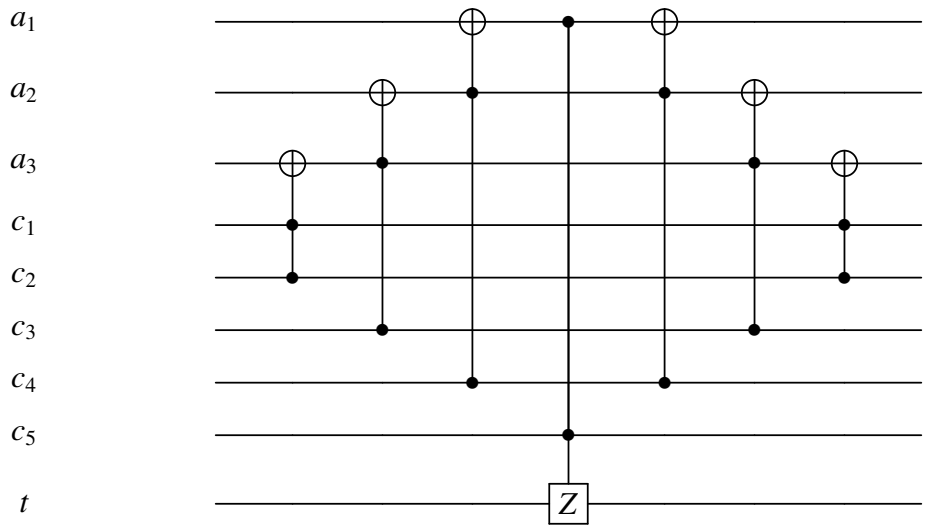
The diagram represents a little 3-qubit algorithm. Each row is the time line of one qubit; as usual, time flows left to right. Each column is an application of a gate. The second gate is *Toffoli*; that is how it is usually drawn.

First, H is executed on the third qubit. Second *Toffoli* is executed, with the first two qubits as the two controls and the third qubit the target. Finally, again H is executed on the third qubit. To prove that the diagram works as intended, consider a basis vector $|c_1\rangle|c_2\rangle|t\rangle$ in $(\mathbb{C}^2)^{\otimes 3}$. If at least one of the control qubits is $|0\rangle$ then the Toffoli gate does nothing, and H^2 is applied to the target. But H^2 is the identity operator. Now suppose that both control qubits are $|1\rangle$. The HXH is applied to the target. But $HXH = Z$.

> **Q:** In principle you should be able to control any unitary operator U.
>
> **A:** Yes, if U works on r qubits then the k-controlled version $c^k U$ of U works on $(k+r)$-qubit basis vectors as expected. If all k controls are $|1\rangle$ then execute U on the remaining r qubits and leave all other qubits unchanged. If at least one control is $|0\rangle$ then do nothing.

6.5 Multiply controlled Z

We illustrate the synthesis of $c^5 Z$.



The symbols $a_1, \ldots, t$ on the left do not belong to the diagram; they just label the lines of the diagram. The diagram itself represents a 9-qubit algorithm. Again, each line is the time line of one qubit, and time flows left to right. The first three qubits, a_1, a_2, a_3, are ancillas. It is presumed that originally they all are in state (represented by the vector) $|0\rangle$. The next 5 qubits are control qubits of the desired operator c^5Z, and the last qubit is the target of the desired algorithm. The columns represent gates; we treat c^2Z as an auxiliary gate.

To verify the correctness of the algorithm, consider a basis vector

$$|a_1\rangle|a_2\rangle|a_3\rangle|c_1\rangle|c_2\rangle|c_3\rangle|c_4\rangle|c_5\rangle|t\rangle.$$

First assume $|c_j\rangle$=$|1\rangle$ for $j = 1, \ldots, 5$. The first three gates set all ancillas to $|1\rangle$. The fourth gate executes Z on the target. The last three gates reset the ancillas back to $|0\rangle$. Now suppose that some $|c_j\rangle$=$|0\rangle$. Several cases arise depending on the value of j. If $j = 5$ then the fourth gate would not change the target qubit. If $j = 4$, then the first ancilla will remain in state $|0\rangle$, and so the state of the target qubit will not change. By now, it should obvious how to finish the proof.

6.6 The synthesis of U

First transform the given Boolean oracle $f : \{0,1\}^n \to \{0,1\}$ into a Boolean circuit C which computes the reversible function $F(x, b) = (x, b \oplus fx)$, and then transform C to the desired quantum oracle U.

There are various techniques in the literature which help to accomplish these tasks. Much depends on how f is given. If f is given by a Boolean circuit, rewrite

the circuit in such a way that the only gates used are negation and conjunction. Then treat the boolean variables as qubits. To this end, employ ancillas as follows. Instead of $\neg b$ use $X|b\rangle$, i.e., use $\mathit{Toffoli}(|1\rangle|1\rangle|b\rangle)$. Instead of $a \wedge b$ use $\mathit{Toffoli}(|a\rangle|b\rangle|0\rangle)$.

6.7 The synthesis of W

It suffices to compute the operator $-W$. If $\left\lfloor (\pi/4)\sqrt{N} \right\rfloor$ is odd then the output vector of Grover's algorithm acquires a factor -1. But that factor makes no difference. Recall that collinear vectors represent the same state. And what is measured at the end is the state and not a representation of it.

The operator $-W$ multiplies the initial vector $|\mu\rangle$ by -1 and leaves every vector $|v\rangle$ orthogonal to $|\mu\rangle$ unchanged. It turns out that $-W$ is exactly the operator

$$O = H^{\otimes n} \cdot X^{\otimes n} \cdot c^{n-1}Z \cdot X^{\otimes n} \cdot H^{\otimes n}.$$

It suffices to prove that O multiplies $|\mu\rangle$ by -1 and leaves unchanged every vector orthogonal to $|\mu\rangle$.

Recall that operator $c^{n-1}Z$ multiplies $|1^n\rangle$ by -1 and leaves every other basis vector unchanged. Therefore it leaves every vector orthogonal to $|1^n\rangle$ unchanged. It follows that operator $X^{\otimes n} \cdot c^{n-1}Z \cdot X^{\otimes n}$ multiplies $|0^n\rangle$ by -1 and leaves every vector orthogonal to $|0^n\rangle$ unchanged. Unitary operator $H^{\otimes n}$ moves $|0^n\rangle$ to $|\mu\rangle$. Notice that $(H^{\otimes n})^2$ is the identity operator, and so $H^{\otimes n}$ is its own inverse. Now it is easy to see that O multiplies $|\mu\rangle$ by -1 and leaves every vector orthogonal to $|\mu\rangle$ unchanged. Indeed $H^{\otimes n}$ transforms $|\mu\rangle$ to $|0^n\rangle$, then $X^{\otimes n} \cdot c^{n-1}Z \cdot X^{\otimes n}$ multiplies $|0^n\rangle$ by -1, and then $H^{\otimes n}$ transforms $-|0^n\rangle$ to $-|\mu\rangle$. If $|v\rangle$ is orthogonal to $|\mu\rangle$ then $H^{\otimes n}$ transforms $|v\rangle$ to a vector $|w\rangle$ orthogonal to $|0^n\rangle$, then $X^{\otimes n} \cdot c^{n-1}Z \cdot X^{\otimes n}$ leaves $|w\rangle$ unchanged, and then $H^{\otimes n}$ transforms $|w\rangle$ back to $|v\rangle$. Thus O leaves unchanged every vector orthogonal to $|\mu\rangle$.

It remains to notice that every factor in O can be synthesized, and therefore O itself can be synthesized.

6.8 Measurement

The final step of Grover's algorithm is the so-called "measurement in the standard basis" of the n-qubit system in state $(WV)^{\lfloor(\pi/4)\sqrt{N}\rfloor}|\mu\rangle$. A quantum computer is supposed to be able to perform such a measurement feasibly in any state

$$|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$$

of the n-qubit system. The result of the measurement is one of the n-bit binary strings x, and the post-measurement state of the system is $|x\rangle$. According to quantum mechanics, the probability of obtaining string x is $|\alpha_x|^2$. In particular, if $|\psi\rangle$ is very close to one of the basis vectors $|y\rangle$, then α_y is close to 1 while all other coefficients α_x are close to 0. In such a case, the result of the measurement will very probably be y.

In the case $|\psi\rangle = (WV)^{\lfloor(\pi/4)\sqrt{N}\rfloor}|\mu\rangle$, we can check whether the result x of the measurement satisfies the equation $fx = 1$. The probability $|\alpha_s|^2$ that the result is s is close to 1 because $|\psi\rangle$ is so close to $|s\rangle$. So it is very probable that the measurement will result in s. In the unfortunate and rare case, of probability $\varepsilon = 1 - |\alpha_s|^2$, that the measurement results in some $x \neq s$, execute Grover's algorithm again and, if needed, again. The probability ε^k that you fail k times in a row decreases exponentially in k.

7 Final discussion

Q: This is not as simple as you guys pretend. A lot is expected from the quantum computer that executes Grover's algorithm.

1. Initialize input states, including ancillas, at least to the prescribed states in the standard basis.
2. Apply Hadamard and Toffoli gates to any specified qubits.
3. Measure the final state in the standard basis.
4. Count (so as to be able to iterate the operator WV the right number of times).
5. Preprocess the given Boolean oracle f to produce the appropriate quantum oracle U.

A: Counting can be done classically. A quantum computer comes with a classical controller which can count iterations. As far as the measurement is concerned, one can show that the desired measurement can be performed by measuring every one of the n relevant qubits separately in its own standard basis $\{|0\rangle, |1\rangle\}$, with the result being 0 or 1.

Q: It is the f-to-U marvel that bothers me most. You have never returned to the original example of the phone directory. How do you turn this thick book into a quantum oracle?

A: The phone book is a nice metaphor. Real uses of Grover's algorithm might involve "virtual phone books." Here is one example [2].

Suppose that some gangsters use a standard encoding algorithm E with 56-bit keys. You know the algorithm E but not the secret key s that they use. You happened to intercept a matching pair of a clear text T and the encoded text $T' = E_s(T)$. Your goal is to find the key s. This problem can be viewed as a phone directory problem where every possible key x is a name and $E_x(T)$ is corresponding phone number. Given the "phone number" T', find the "name" s in the "phone directory."

Q: I get it. For fixed T and T', the Boolean oracle

$$f(x) = \begin{cases} 1 & \text{if } E_x(T) = T' \\ 0 & \text{otherwise} \end{cases}$$

is given by a reasonable size program which, I trust, can be feasibly transformed into a Boolean circuit and then into the desired quantum oracle U.

A: Yes.

Acknowledgment

This exposition of Grover's algorithm is influenced by that in David Mermin's book [6].

References

[1] Charles Henry Bennet, Ethan Bernstein, Gilles Brassard and Umesh Vazirani, "Strength and weaknesses of quantum computing," *SIAM Journal on Computing* 26:5, 1997

[2] Gilles Brassard, "Searching a quantum phone book," *Science* 275:5300 627–628

[3] Paul A. M. Dirac, "A new notation for quantum mechanics," *Mathematical Proceedings of the Cambridge Philosophical Society* 35:3 416—418

[4] Cătălin Dohotaru and Peter Høyer, "Exact quantum lower bound for Grover's problem," arXiv:0810.3647, 2008

[5] Lov Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th Annual ACM Symposium on the Theory of Computing*, 1996

[6] N. David Mermin, "Quantum computer science: An introduction," Cambridge University Press, 2007

[7] Michael A. Nielsen and Isaac L. Chuang, "Quantum computation and quantum information," 10th Anniversary Edition, Cambridge University Press, 2010

[8] Petr K. Rashevsky, "Riemannian geometry and tensor analysis," Nauka 1967 (in Russian)

[9] Yaoyun Shi, "Both Toffoli and Controlled-Not need little help to do universal quantum computation," *Quantum Information & Computation* 3:1 84—92, 2003. arXiv:0205115

[10] Tommaso Toffoli, "Reversible computing," in ICALP, *International Colloquium on Automata, Languages, and Programming* 632–644, 1980.