

Grover's Algorithm: Quantum Database Search*

C. Lavor

Instituto de Matemática e Estatística
Universidade do Estado do Rio de Janeiro - UERJ
Rua São Francisco Xavier, 524, 6º andar, bl. D, sala 6018,
Rio de Janeiro, RJ, 20550-900, Brazil
e-mail: carlile@ime.uerj.br

L.R.U. Manssur, R. Portugal
Coordenação de Ciência da Computação
Laboratório Nacional de Computação Científica - LNCC
Av. Getúlio Vargas 333, Petrópolis, RJ, 25651-070, Brazil
e-mail: {leon,portugal}@lncc.br

July 25, 2003

Abstract

We review Grover's algorithm by means of a detailed geometrical interpretation and a worked out example. Some basic concepts of Quantum Mechanics and quantum circuits are also reviewed. This work is intended for non-specialists which have basic knowledge on undergraduate Linear Algebra.

1 Introduction

The development of quantum software and hardware is an exciting new area posing extremely difficult challenges for researchers all over the world. It promises a new era in Computer Science, but it is not clear at all whether it will be possible to build a hardware of reasonable size. Quantum hardware of macroscopic sizes suffer the decoherence effect which is an unsurmountable tendency to behave classically.

An important landmark in hardware development is the experience performed at IBM's lab in San Jose, California, which factored the number 15 into its prime factors using a quantum algorithm (Shor's algorithm [1]) executed in a molecule, perfluorobutadienyl iron complex [2]. This "quantum computer" has seven "quantum bits". Such

*Contents based on lecture notes from graduate courses in Quantum Computation given at LNCC.

insignificant amount of bits that could be manipulated in the experience shows the challenge in hardware development.

Quantum software development is facing difficulties too, though the superiority of quantum circuits over classical ones was already established. In this context, Grover's algorithm [3, 4] plays an important role, since it provides a proof that quantum computers are faster than classical ones in database searching. The best classical algorithm for an unstructured database search has complexity $O(N)$, without possibility of further improvement, while the best quantum algorithm has complexity $O(\sqrt{N})$.

Historically, Deutsch's algorithm [5] was the first example of a quantum circuit faster than its classical counterpart, while Bernstein and Vazirani [6] and Simon [7] provided the first examples of quantum algorithms exponentially faster than their classical counterparts. These algorithms determine some sort of functions' periods and their only application seems to be for proving that quantum circuits are faster than classical ones.

Some of the most impressive results so far in quantum computation are the Shor's algorithms [1] for factoring integers and for finding discrete logarithms, which also provided an exponential speed up over the known classical algorithms. Shor's results attracted a lot of attention because they render most of current cryptography methods useless, if we assume it is possible to build quantum hardware of reasonable size.

This work is an introductory review of Grover's algorithm. We have put all our efforts to write as clear as possible for non-specialists. We assume familiarity with undergraduate Linear Algebra, which is the main mathematical basis of Quantum Mechanics. Some previous knowledge of Quantum Mechanics is welcome, but not necessary for a persistent reader. The reader can find further material in [8, 9, 10, 11, 12, 13].

Section 2 reviews basic notions about classical computers preparing for the quantum generalization, which is presented in Section 3. Section 4 introduces the notion of quantum circuits and presents some basic examples. Section 5 describes Grover's algorithm. Sections 6 and 8 give details of the geometrical interpretation while Section 7 presents a worked out example. Finally, Section 9 shows the decomposition of Grover's circuit in terms of the universal gates.

2 The Classical Computer

A classical computer can be understood in a very broad sense as a machine that reads a certain amount of data, encoded as zeroes and ones, performs calculations, and prints in the end output data as zeroes and ones again. Zeroes and ones are states of some physical quantity, the electric potential in classical computers. Internally, a zero is a state of low electric potential and a one is a state of high electric potential. This point is crucial to the generalization we will discuss ahead.

Zeroes and ones form a binary number which can be converted to decimal notation. Let us think of the computer as calculating a function

$$f : \{0, \dots, N - 1\} \rightarrow \{0, \dots, N - 1\},$$

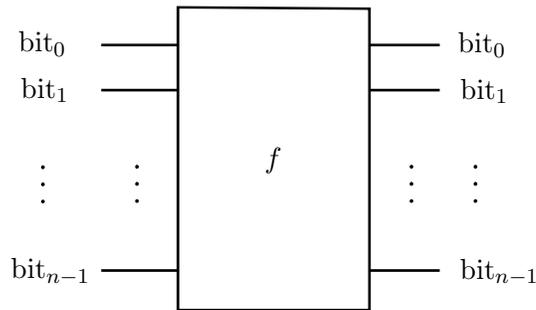


Figure 1: Outline of the classical computer.

where N is a number of the form 2^n (n is the number of bits in the computer memory). We assume without loss of generality that the domain and codomain of f are of the same size (f is a function because one input cannot generate two or more outputs). We represent the calculation process in Fig. 1, where on the left hand side we have the value of each bit (zero or one). The process of calculation goes from left to right, and the output is stored in the same bits on the right hand side.

Usually f is given in terms of elementary blocks that can be implemented in practice using transistors and other electrical devices. The blocks are the AND, OR and NOT gates, known as universal gates (This set could be reduced further since OR can be written in terms of AND and NOT). For example, the circuit to add two one-bit numbers modulo 2 is given in Fig. 2. The possible inputs are 00, 01, 10, 11, and the corresponding outputs are 00, 01, 11, 10. The inputs are prepared creating electric potential gaps, which create electric currents that propagate through the wires towards right. The gates are activated as time goes by. The meter symbols on the right indicate that measurements of the electric potential are performed, which tell whether the output value of each bit is zero or one. The second bit gives the result of the calculation. The wire for the output of the first bit is artificial and unnecessary; at this point, it is there simply to have the codomain size of the function f equal to the domain size. This circuit, without the first bit output wire, is the circuit for the XOR (exclusive OR) gate in terms of the universal

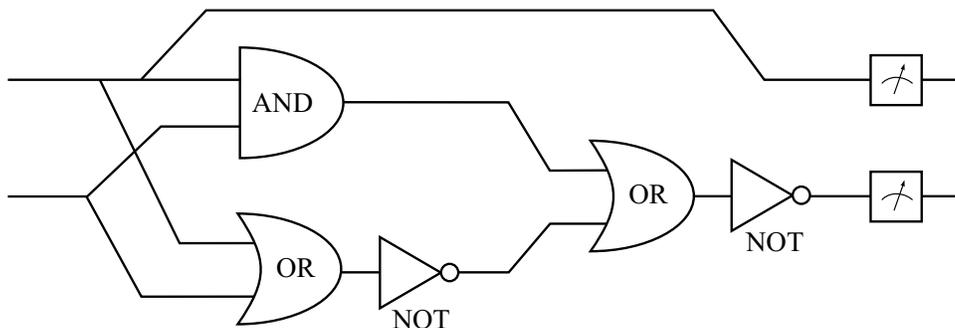


Figure 2: The circuit to add two one-bit numbers modulo 2.

gates.

The circuit of Fig. 2 is irreversible, since the gates AND and OR are irreversible. If the output of the AND gate is 0, nobody can tell what was the input, and similarly when the output of the OR gate is 1. This means that the physical theory which describes the processes in Fig. 2 must be irreversible. Then, the AND and OR gates cannot be straightforwardly generalized to quantum gates, which must be reversible ones.

However, the circuit of Fig. 2 can be made reversible. Although the following description is unnecessary from the classical point of view, it helps the quantum generalization performed in the next sections. We employ the controlled-NOT (CNOT) gate of Fig. 3. The bits a and b assume values either 0 or 1. The value of the first bit (called the control bit) never changes in this gate; the second bit (called the target bit) is flipped only if $a = 1$. If $a = 0$, nothing happens to both bits. The gate \oplus is a NOT gate controlled by the value of the first bit. Now it is easy to verify that the value of the second bit for this gate is $a + b \pmod{2}$. The CNOT gate is not a universal building block for classical circuits, but its quantum counterpart is a basic block of quantum circuits.

We have described the reversible counterpart of the XOR gate. What is the reversible counterpart of the AND gate? The answer employs the *Toffoli* gate (Fig. 4) which is a generalization of the CNOT gate with two control bits instead of one. The value of the third bit (target) is inverted only if both a and b are 1, otherwise it does not change. The following table describes all possible inputs and the corresponding outputs:

000	→	000
001	→	001
010	→	010
011	→	011
100	→	100
101	→	101
110	→	111
111	→	110

The AND gate can be replaced by the Toffoli gate simply by taking $c = 0$. The output of the third bit is then a AND b (The reversible circuit for the OR gate is a little cumbersome because it requires more than one Toffoli gate, so we will not describe it here).

Another feature implicit in Fig. 2 that cannot be performed in quantum circuits

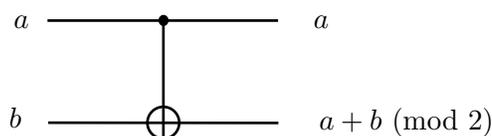


Figure 3: Classical controlled-NOT (CNOT) gate.

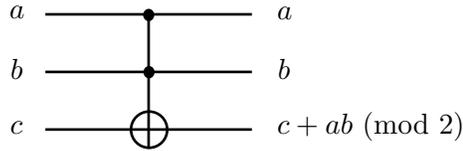


Figure 4: Classical Toffoli gate.

is FANOUT. Note that there are bifurcations of wires; there is no problem to do this classically. However, this is forbidden in quantum circuits, due to the “no cloning” theorem (see [10] p.162). Classical FANOUT can be obtained from the CNOT gate by taking $b = 0$. The value of the first bit is then duplicated.

Consider again Fig. 1. If the computer has n bits, there are 2^n possible inputs. For each input there are 2^n possible outputs, therefore the number of possible functions f that can be calculated is 2^{n2^n} . All these functions can be reduced to circuits using the universal gates. That is what a classical computer can do: calculate 2^{n2^n} functions. This number is astronomical for computers with 1 gigabyte, that is a typical memory size for good personal computers nowadays.

Another important issue is how fast can the computer calculate these functions. The answer depends on the number of gates used in the circuit for f . If the number of elementary gates increases polynomially with n , we say that the circuit is “efficient”. If the number of gates increases exponentially with n , the circuit is “inefficient”. This is a very coarse method to measure the efficiency, but it is useful for theoretical analysis when n is large. Note that we are thinking of computers of variable size, which is not the case in practice. In fact, instead of referring to actual computers, it is better to use a Turing machine, which is an abstract model for computers and softwares as a whole [14]. Similarly, quantum computers and their softwares are abstractly modeled as the quantum Turing machine [15, 6]. The classical theory of complexity classes and its quantum counterpart address this kind of problems.

All calculations that can be performed in a classical computer can also be performed in a quantum computer. One simply replaces the irreversible gates of the classical computer with their reversible counterparts. The new circuit can be implemented in a quantum computer. But there is no advantage in this procedure: why build a very expensive quantum machine which behaves classically? The appeal of quantum computation is the possibility of quantum algorithms faster than classical ones. The quantum algorithms must use quantum features not available in classical computers, such as quantum parallelism and entanglement, in order to enhance the computation. On the other hand, a naïve use of quantum features does not guarantee any improvements. So far, there are only two classes of successful quantum algorithms: the database search algorithms and the algorithms for finding the generators of a normal subgroup of a given group. Shor’s algorithms for integer factorization and discrete logarithm are special cases of this latter class.

3 The Quantum Computer

In quantum computers, one is allowed to use quantum states instead of classical ones. So, the electric potential can be replaced by some quantum state: the *quantum bit* (*qubit* for short). Just as a bit has a state 0 or 1, a qubit also has a state $|0\rangle$ or $|1\rangle$. This is called the Dirac notation and it is the standard notation for states in Quantum Mechanics. The difference between bits and qubits is that a qubit $|\psi\rangle$ can also be in a linear combination of states $|0\rangle$ and $|1\rangle$:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle. \quad (1)$$

This state is called a *superposition* of the states $|0\rangle$ and $|1\rangle$ with *amplitudes* α and β (α and β are complex numbers). Thus, the state $|\psi\rangle$ is a vector in a two-dimensional complex vector space, where the states $|0\rangle$ and $|1\rangle$ form an orthonormal basis, called the computational basis (see Fig. 5 in the real case).

The state $|0\rangle$ is not the zero vector, but simply the first vector of the basis. The matrix representations of the vectors $|0\rangle$ and $|1\rangle$ are given by

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

What is the interpretation of α and β in Eq. (1)? Quantum mechanics tells us that if one measures the state $|\psi\rangle$ one gets either $|0\rangle$, with probability $|\alpha|^2$, or $|1\rangle$, with probability $|\beta|^2$. That is, measurement changes the state of a qubit. In fact, any attempt to find out the amplitudes of the state $|\psi\rangle$ produces a nondeterministic collapse of the superposition to either $|0\rangle$ or $|1\rangle$. If $|\alpha|^2$ and $|\beta|^2$ are probabilities and there are only two possible outputs, then

$$|\alpha|^2 + |\beta|^2 = 1. \quad (2)$$

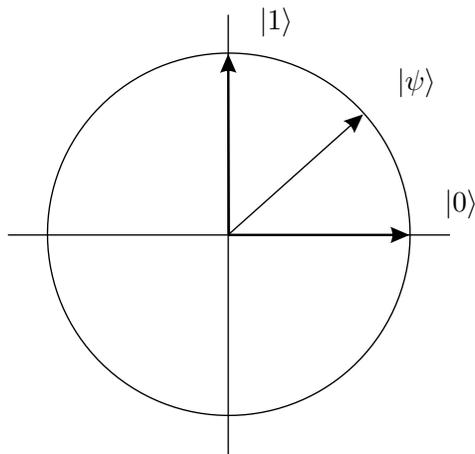


Figure 5: Computational basis for the case α, β real. In the general case (α, β complex) there is still a geometrical representation called the Bloch sphere [9].

Calculating the norm of $|\psi\rangle$, Eq. (2) gives

$$\| |\psi\rangle \| = \sqrt{|\alpha|^2 + |\beta|^2} = 1.$$

If a qubit is in state $|\psi\rangle$ given by Eq. (1), there are two ways it can interact. The first one is a measurement. This forces the state $|\psi\rangle$ to collapse to either $|0\rangle$ or $|1\rangle$ with probabilities $|\alpha|^2$ and $|\beta|^2$, respectively. Note that the measurement does not give the value of α and β . They are inaccessible via measurements unless one has many copies of the same state. The second kind of interaction does not give any information about the state. In this case, the values of α and β change keeping the constraint (2). The most general transformation of this kind is a linear transformation U that takes unit vectors into unit vectors. Such transformation is called *unitary* and can be defined by

$$U^\dagger U = U U^\dagger = I,$$

where $U^\dagger = (U^*)^T$ (* indicates complex conjugation and T indicates the transpose operation).

To consider multiple qubits it is necessary to introduce the concept of *tensor product*. Suppose V and W are complex vector spaces of dimensions m and n , respectively. The tensor product $V \otimes W$ is an mn -dimensional vector space. The elements of $V \otimes W$ are linear combinations of tensor products $|v\rangle \otimes |w\rangle$, satisfying the following properties ($z \in \mathbb{C}$, $|v\rangle, |v_1\rangle, |v_2\rangle \in V$, and $|w\rangle, |w_1\rangle, |w_2\rangle \in W$):

1. $z(|v\rangle \otimes |w\rangle) = (z|v\rangle) \otimes |w\rangle = |v\rangle \otimes (z|w\rangle)$,
2. $(|v_1\rangle + |v_2\rangle) \otimes |w\rangle = (|v_1\rangle \otimes |w\rangle) + (|v_2\rangle \otimes |w\rangle)$,
3. $|v\rangle \otimes (|w_1\rangle + |w_2\rangle) = (|v\rangle \otimes |w_1\rangle) + (|v\rangle \otimes |w_2\rangle)$.

We use also the notations $|v\rangle|w\rangle$, $|v, w\rangle$ or $|vw\rangle$ for the tensor product $|v\rangle \otimes |w\rangle$. Note that the tensor product is non-commutative, so the notation must preserve the ordering.

Given two linear operators A and B defined on the vector spaces V and W , respectively, we can define the linear operator $A \otimes B$ on $V \otimes W$ as

$$(A \otimes B)(|v\rangle \otimes |w\rangle) = A|v\rangle \otimes B|w\rangle, \quad (3)$$

where $|v\rangle \in V$ and $|w\rangle \in W$. The matrix representation of $A \otimes B$ is given by

$$A \otimes B = \begin{bmatrix} A_{11}B & \cdots & A_{1m}B \\ \vdots & \ddots & \vdots \\ A_{m1}B & \cdots & A_{mm}B \end{bmatrix}, \quad (4)$$

where A is an $m \times m$ matrix and B is a $n \times n$ matrix (We are using the same notation for the operator and its matrix representation). So the matrix $A \otimes B$ has dimension $mn \times mn$. For example, given

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

the tensor product $A \otimes B$ is

$$A \otimes B = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

The formula (4) can also be used for non square matrices, such as the tensor product of two vectors. For example, the tensor product $|0\rangle \otimes |1\rangle$ is given by

$$|0\rangle \otimes |1\rangle = |01\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

The notations $|\psi\rangle^{\otimes k}$ and $A^{\otimes k}$ mean $|\psi\rangle$ and A tensored with themselves k times, respectively.

The general state $|\psi\rangle$ of two qubits is a superposition of the states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$:

$$|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle, \quad (5)$$

with the constraint

$$|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1.$$

Regarding the zeroes and ones as constituting the binary expansion of an integer, we can replace the representations of states

$$|00\rangle, |01\rangle, |10\rangle, |11\rangle,$$

by the shorter forms

$$|0\rangle, |1\rangle, |2\rangle, |3\rangle,$$

in decimal notation.

In general, the state $|\psi\rangle$ of n qubits is a superposition of the 2^n states $|0\rangle, |1\rangle, \dots, |2^n - 1\rangle$:

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle,$$

with amplitudes α_i constrained to

$$\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1.$$

The orthonormal basis $\{|0\rangle, \dots, |2^n - 1\rangle\}$ is called *computational basis*. As before, a measurement of a generic state $|\psi\rangle$ yields the result $|i_0\rangle$ with probability $|\alpha_{i_0}|^2$, where

$0 \leq i_0 < N$. Usually, the measurement is performed qubit by qubit yielding zeroes or ones that are read together to form i_0 . We stress again a very important feature of the measurement process. The state $|\psi\rangle$ as it is before measurement is inaccessible unless it is in the computational basis. The measurement process inevitably disturbs $|\psi\rangle$ forcing it to collapse to one vector of the computational basis. This collapse is non-deterministic, with the probabilities given by the squared norms of the corresponding amplitudes in $|\psi\rangle$.

If we have two qubits, one in the state

$$|\varphi\rangle = a|0\rangle + b|1\rangle$$

and the other in the state

$$|\psi\rangle = c|0\rangle + d|1\rangle,$$

then the state of the pair $|\varphi\rangle|\psi\rangle$ is the tensor product

$$\begin{aligned} |\varphi\rangle \otimes |\psi\rangle &= (a|0\rangle + b|1\rangle) \otimes (c|0\rangle + d|1\rangle) \\ &= ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle. \end{aligned} \tag{6}$$

Note that a general 2-qubit state (5) is of the form (6) if and only if

$$\begin{aligned} \alpha &= ac, \\ \beta &= ad, \\ \gamma &= bc, \\ \delta &= bd. \end{aligned}$$

From these equalities we have that a general 2-qubit state (5) is of the form (6) if and only if

$$\alpha\delta = \beta\gamma.$$

Thus, the general 2-qubit state is not a product of two 1-qubit states. Such non-product states of two or more qubits are called *entangled* states, for example, $(|00\rangle + |11\rangle)/\sqrt{2}$.

There is an *inner product* between two n -qubit states $|\varphi\rangle$ and $|\psi\rangle$, written in the form $\langle\varphi|\psi\rangle$, which is defined by the following rules in a complex vector space V :

1. $\langle\psi|\varphi\rangle = \langle\varphi|\psi\rangle^*$,
2. $\langle\varphi|(a|u\rangle + b|v\rangle)\rangle = a\langle\varphi|u\rangle + b\langle\varphi|v\rangle$,
3. $\langle\varphi|\varphi\rangle > 0$ if $|\varphi\rangle \neq 0$,

where $a, b \in \mathbb{C}$ and $|\varphi\rangle, |\psi\rangle, |u\rangle, |v\rangle \in V$. The *norm* of a vector $|\varphi\rangle$ is given by

$$\| |\varphi\rangle \| = \sqrt{\langle\varphi|\varphi\rangle}.$$

The notation $\langle\varphi|$ is used for the *dual vector* to the vector $|\varphi\rangle$. The dual is a linear operator from the vector space V to the complex numbers, defined by

$$\langle\varphi|(|v\rangle) = \langle\varphi|v\rangle, \quad \forall|v\rangle \in V.$$

Given two vectors $|\varphi\rangle$ and $|\psi\rangle$ in a vector space V , there is also an *outer product* $|\psi\rangle\langle\varphi|$, defined as a linear operator on V satisfying

$$(|\psi\rangle\langle\varphi|)|v\rangle = |\psi\rangle\langle\varphi|v\rangle, \quad \forall|v\rangle \in V.$$

If $|\varphi\rangle = a|0\rangle + b|1\rangle$ and $|\psi\rangle = c|0\rangle + d|1\rangle$, then the matrix representations for inner and outer products are:

$$\begin{aligned} \langle\varphi|\psi\rangle &= \begin{bmatrix} a^* & b^* \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix} = a^*c + b^*d, \\ |\varphi\rangle\langle\psi| &= \begin{bmatrix} a \\ b \end{bmatrix} \begin{bmatrix} c^* & d^* \end{bmatrix} = \begin{bmatrix} ac^* & ad^* \\ bc^* & bd^* \end{bmatrix}. \end{aligned}$$

Notice the complex conjugation in the process of taking the dual.

After the above review, we are ready to outline the quantum computer. Fig. 6 is the generalization of Fig. 1 to the quantum case. The function f is replaced by a unitary operator U and classical bits are replaced by quantum bits, where each one has a state $|\psi_i\rangle$. In Fig. 6, we are taking a non-entangled input, what is quite reasonable. In fact, $|\psi_i\rangle$ is either $|0\rangle$ or $|1\rangle$ generally. $|\psi\rangle$ on the right hand side of Fig. 6 is the result of the application of U on the input. The last step is the measurement of the states of each qubit, which returns zeroes and ones that form the final result of the quantum calculation. Note that there is, in principle, an infinite number of possible operators U , which are unitary $2^n \times 2^n$ matrix, with continuous entries. In particular, one must take errors into account, which reduces the number of implementable circuits. But even in this case, the number of degrees of freedom is greater than in the classical case.

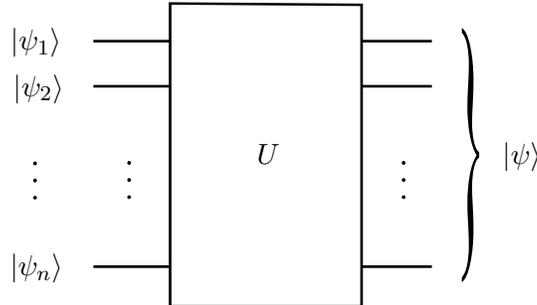


Figure 6: The sketch of the quantum computer. We consider the input non-entangled, which is reasonable in general. On the other hand, the output is entangled in general. The measurement of the state $|\psi\rangle$, not shown here, returns zeroes and ones.

Similarly to the classical case, the operator U is in general written in terms of gates forming a quantum circuit, which is the topic of the next section.

4 Quantum Circuits

Let us start with one-qubit gates. In the classical case there is only one possibility, which is the NOT gate, like the ones used in Fig. 2. The straightforward generalization to the quantum case is given in Fig. 7, where X is the unitary operator

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

So, if $|\psi\rangle$ is $|0\rangle$, the output is $|1\rangle$ and vice-versa. But now we have a situation with no classical counterpart. The state $|\psi\rangle$ can be a superposition of states $|0\rangle$ and $|1\rangle$. The general case is given in Eq. (1). The output in this case is $\alpha|1\rangle + \beta|0\rangle$.

The gate X is not the only one-qubit gate. There are infinitely many, since there are an infinite number of 2×2 unitary matrices. In principle, any unitary operation can be implemented in practice. The *Hadamard* gate is another important one-qubit gate, given by

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

It is easy to see that

$$\begin{aligned} H|0\rangle &= \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \\ H|1\rangle &= \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \end{aligned}$$

If the input is $|0\rangle$, the Hadamard gate creates a superposition of states with equal weights. This is a general feature, valid for two or more qubits. Let us analyze the 2-qubit case.

The first example of a 2-qubit gate is $H \otimes H$:

$$\begin{aligned} H^{\otimes 2}|0\rangle|0\rangle &= (H \otimes H)(|0\rangle \otimes |0\rangle) = H|0\rangle \otimes H|0\rangle \\ &= \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) \otimes \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) \\ &= \frac{1}{2}(|0\rangle|0\rangle + |0\rangle|1\rangle + |1\rangle|0\rangle + |1\rangle|1\rangle) \\ &= \frac{1}{2}(|0\rangle + |1\rangle + |2\rangle + |3\rangle). \end{aligned}$$

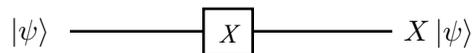


Figure 7: Quantum NOT gate.

The result is a superposition of all basis states with equal weights. More generally, the Hadamard operator applied to the n -qubit state $|0\rangle$ is

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

Thus, the tensor product of n Hadamard operators produces an equally weighted superposition of all computational basis states, when the input is the state $|0\rangle$.

Another important 2-qubit quantum gate is the CNOT gate, which is the quantum generalization of the classical gate described earlier (Fig. 3). It has two input qubits, the control and the target qubit, respectively. The target qubit is flipped only if the control qubit is set to 1, that is,

$$\begin{aligned} |00\rangle &\rightarrow |00\rangle, \\ |01\rangle &\rightarrow |01\rangle, \\ |10\rangle &\rightarrow |11\rangle, \\ |11\rangle &\rightarrow |10\rangle. \end{aligned} \tag{7}$$

The action of the CNOT gate can also be represented by

$$|a, b\rangle \rightarrow |a, a \oplus b\rangle,$$

where \oplus is addition modulo 2. Now, let us obtain its matrix representation. We know that

$$\begin{aligned} |00\rangle &= |0\rangle \otimes |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \\ |01\rangle &= |0\rangle \otimes |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \\ |10\rangle &= |1\rangle \otimes |0\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \\ |11\rangle &= |1\rangle \otimes |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \end{aligned} \tag{8}$$

Thus, from (7) and (8), the matrix representation U_{CNOT} of the CNOT gate is

$$U_{\text{CNOT}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Fig. 8 describes the CNOT gate, where $|i\rangle$ is either $|0\rangle$ or $|1\rangle$. The figure could lead one to think that the output is always non-entangled, but that is not true, since if the first qubit is in a more general state given by $a|0\rangle + b|1\rangle$, then the output will be $a|0\rangle|\sigma\rangle + b|1\rangle X|\sigma\rangle$, which is entangled in general.

CNOT and one-qubit gates form a universal set of gates. This means that any other gate, operating on 2 or more qubits can be written as compositions and direct products of CNOT and one-qubit gates [16].

We have seen two examples of 2-qubit gates. The general case is a 4×4 unitary matrix. Gates that are the direct product of other gates, such as $H \otimes H$, do not produce entanglement. If the input is non-entangled, the output is not too. On the other hand, the output of the CNOT gate can be entangled while the input is non-entangled.

The next gate we consider is the 3-qubit quantum Toffoli gate. Its action on the computational basis is given by

$$|a, b, c\rangle \rightarrow |a, b, c \oplus ab\rangle.$$

The action on a generic state

$$|\psi\rangle = \sum_{a,b,c=0}^1 \alpha_{a,b,c} |a, b, c\rangle = \begin{bmatrix} \alpha_{000} \\ \vdots \\ \alpha_{101} \\ \alpha_{110} \\ \alpha_{111} \end{bmatrix}$$

is obtained by linearity as

$$|\psi'\rangle = \sum_{a,b,c=0}^1 \alpha_{a,b,c} |a, b, c \oplus ab\rangle = \begin{bmatrix} \alpha_{000} \\ \vdots \\ \alpha_{101} \\ \alpha_{111} \\ \alpha_{110} \end{bmatrix}.$$

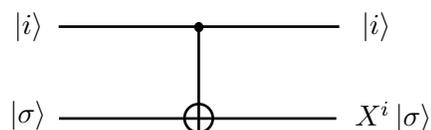


Figure 8: CNOT gate.

So, the matrix representation for the Toffoli gate becomes

$$U_{\text{Toffoli}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Further details about quantum circuits can be found in [16, 9].

5 Grover's Algorithm

Suppose we have an unstructured database with N elements. Without loss of generality, suppose that the elements are numbers from 0 to $N - 1$. The elements are not ordered. Classically, we would test each element at a time, until we hit the one searched for. This takes an average of $N/2$ attempts and N in the worst case, therefore the complexity is $O(N)$. As we will see, using Quantum Mechanics only $O(\sqrt{N})$ trials are needed. For simplicity, assume that $N = 2^n$, for some integer n .

Grover's algorithm has two registers: n qubits in the first and one qubit in the second. The first step is to create a superposition of all 2^n computational basis states $\{|0\rangle, \dots, |2^n - 1\rangle\}$ of the first register. This is achieved in the following way. Initialize the first register in the state $|0, \dots, 0\rangle$ and apply the operator $H^{\otimes n}$

$$\begin{aligned} |\psi\rangle &= H^{\otimes n} |0, \dots, 0\rangle \\ &= (H |0\rangle)^{\otimes n} \\ &= \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right)^{\otimes n} \\ &= \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle. \end{aligned} \tag{9}$$

$|\psi\rangle$ is a superposition of all basis states with equal amplitudes given by $1/\sqrt{N}$. The second register can begin with $|1\rangle$ and, after a Hadamard gate is applied, it will be in state $|-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$.

Now define $f : \{0, \dots, N - 1\} \rightarrow \{0, 1\}$ as a function which recognizes the solution:

$$f(i) = \begin{cases} 1 & \text{if } i \text{ is the searched element } (i_0) \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

This function is used in the classical algorithm. In the quantum algorithm, let us assume that it is possible to build a linear unitary operator also dependent on f , U_f , such that

$$U_f (|i\rangle |j\rangle) = |i\rangle |j \oplus f(i)\rangle. \tag{11}$$

U_f is called *oracle*. In the above equation, $|i\rangle$ stands for a state of the first register, so i is in $\{0, \dots, 2^n - 1\}$, $|j\rangle$ is a state of the second register, so j is in $\{0, 1\}$, and the sum is modulo 2. It is easy to check that

$$\begin{aligned} U_f(|i\rangle|-\rangle) &= \frac{U_f(|i\rangle|0\rangle) - U_f(|i\rangle|1\rangle)}{\sqrt{2}} \\ &= \frac{|i\rangle|f(i)\rangle - |i\rangle|1 \oplus f(i)\rangle}{\sqrt{2}} \\ &= (-1)^{f(i)} |i\rangle|-\rangle. \end{aligned} \quad (12)$$

In the last equality, we have used the fact that

$$1 \oplus f(i) = \begin{cases} 0 & \text{for } i = i_0 \\ 1 & \text{for } i \neq i_0. \end{cases} \quad (13)$$

Now look at what happens when we apply U_f to the superposition state coming from the first step, $|\psi\rangle|-\rangle$. The state of the second register does not change. Let us call $|\psi_1\rangle$ the resulting state of the first register:

$$\begin{aligned} |\psi_1\rangle|-\rangle &= U_f(|\psi\rangle|-\rangle) \\ &= \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} U_f(|i\rangle|-\rangle) \\ &= \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} (-1)^{f(i)} |i\rangle|-\rangle. \end{aligned} \quad (14)$$

$|\psi_1\rangle$ is a superposition of all basis elements, but the amplitude of the searched element is negative while all others are positive. The searched element has been marked with a minus sign. This result is obtained using a feature called *quantum parallelism*. At the quantum level, it is possible “to see” all database elements simultaneously. The position of the searched element is known: it is the value of i of the term with negative amplitude in (14). This quantum information is not fully available at the classical level. A classical information of a quantum state is obtained by practical measurements, and, at this point, it does not help if we measure the state of the first register, because it is much more likely that we obtain a non-desired element, instead of the searched one. Before we can perform a measure, the next step should be to increase the amplitude of the searched element while decreasing the amplitude of the others. This is quite general: quantum algorithms work by increasing the amplitude of the states which carry the desired result. After that, a measurement will hit the solution with high probability.

Now we shall work out the details by introducing the circuit for Grover’s algorithm (Fig. 9) and analyzing it step by step. The unitary operator G is applied $O(\sqrt{N})$ times. The exact number will be obtained later on. The circuit for one Grover iteration G is given in Fig. 10. The states $|\psi\rangle$ and $|\psi_1\rangle$ are given by Eqs. (9) and (14), respectively. The operator $2|\psi\rangle\langle\psi| - I$ is called inversion about the mean for reasons that will be clear

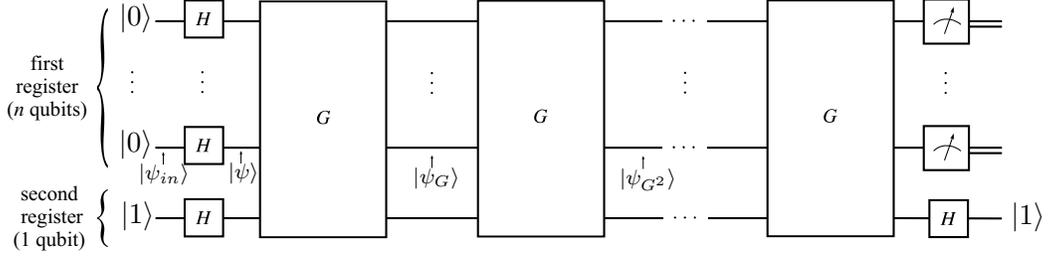


Figure 9: Outline of Grover's algorithm.

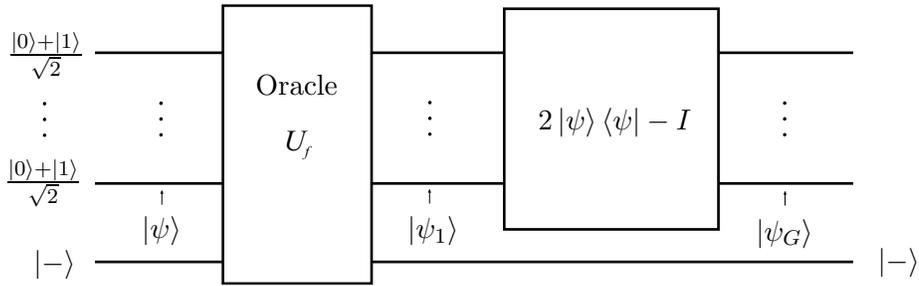


Figure 10: One Grover iteration (G). The states of the first register correspond to the first iteration.

in the next section. We will also show how each Grover operator application raises the amplitude of the searched element. $|\psi_1\rangle$ can be rewritten as

$$|\psi_1\rangle = |\psi\rangle - \frac{2}{\sqrt{2^n}} |i_0\rangle, \quad (15)$$

where $|i_0\rangle$ is the searched element. $|i_0\rangle$ is a state of the computational basis. Note that

$$\langle\psi|i_0\rangle = \frac{1}{\sqrt{2^n}}. \quad (16)$$

Let us calculate $|\psi_G\rangle$ of Fig. 9. Using Eqs. (15) and (16), we obtain

$$\begin{aligned} |\psi_G\rangle &= (2|\psi\rangle\langle\psi| - I)|\psi_1\rangle \\ &= \frac{2^{n-2} - 1}{2^{n-2}} |\psi\rangle + \frac{2}{\sqrt{2^n}} |i_0\rangle. \end{aligned} \quad (17)$$

This is the state of the first register after one application of G . The second register is in the state $|-\rangle$.

6 Geometric Representation

All the operators and amplitudes in Grover's algorithm are real. This means that all states of the quantum computer live in a real vector subspace of the Hilbert space.

This allows a nice geometrical representation taking $|i_0\rangle$ and $|\psi\rangle$ as base vectors (non-orthogonal basis).

In Fig. 11 we can see the vectors $|i_0\rangle$ and $|\psi\rangle$. They form an angle smaller than 90° as can be seen from Eq. (16), since $0 < \langle\psi|i_0\rangle < 1$. If n is large, then the angle is nearly 90° . We can think that $|\psi\rangle$ is the initial state of the first register, and the steps of the computation are the applications of the unitary operators U_f and $2|\psi\rangle\langle\psi| - I$. Then $|\psi\rangle$ will rotate in the real plane spanned by $|\psi\rangle$ and $|i_0\rangle$, keeping the unit norm. This means that the tip of $|\psi\rangle$'s vector lies in the unit circle.

From Eqs. (15) and (16) we see that $|\psi\rangle$ rotates θ degrees clockwise, where (see $|\psi_1\rangle$ in Fig. 11)

$$\cos\theta = 1 - \frac{1}{2^{n-1}}. \quad (18)$$

From Eq. (17) we see that the angle between $|\psi_G\rangle$ and $|\psi\rangle$ is

$$\cos\theta' = \langle\psi|\psi_G\rangle = 1 - \frac{1}{2^{n-1}}. \quad (19)$$

So, $\theta' = \theta$ and $|\psi_1\rangle$ rotates 2θ degrees counterclockwise (in the direction of $|i_0\rangle$). This explains the placement of $|\psi_G\rangle$ in Fig. 11. This is a remarkable result, since the resulting action of $G = (2|\psi\rangle\langle\psi| - I)U_f$ rotates $|\psi\rangle$ towards $|i_0\rangle$ by θ degrees. This means that the amplitude of $|i_0\rangle$ in $|\psi_G\rangle$ increased and the amplitudes of $|i\rangle$, $i \neq i_0$, decreased with respect to their original values in $|\psi\rangle$. A measurement, at this point, will return $|i_0\rangle$ more likely than before. But that is not enough in general, since θ is a small angle if $n \gg 1$ (see Eq. (18)). That is why we need to apply G repeatedly, ending up θ degrees closer to $|i_0\rangle$ each time, until the state of the first register be very close to $|i_0\rangle$, so we can measure.

Now we show that further applications of G also rotate the state of the first register by θ degrees towards $|i_0\rangle$. The proof is quite general: suppose that $|\sigma\rangle$ is a unit vector

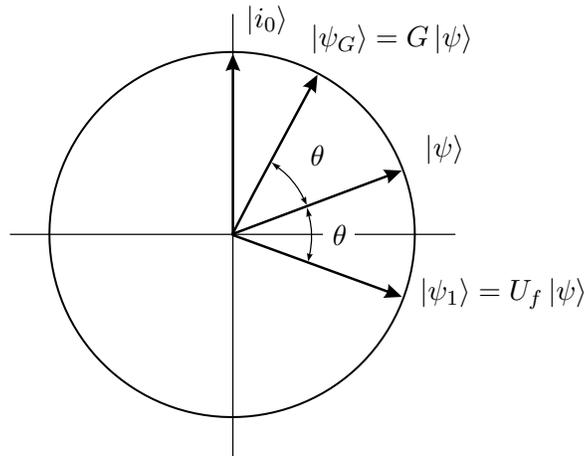


Figure 11: The state of the first register lives in the real vector space spanned by $|i_0\rangle$ and $|\psi\rangle$. We take these states as a basis to describe what happens in Grover's algorithm.

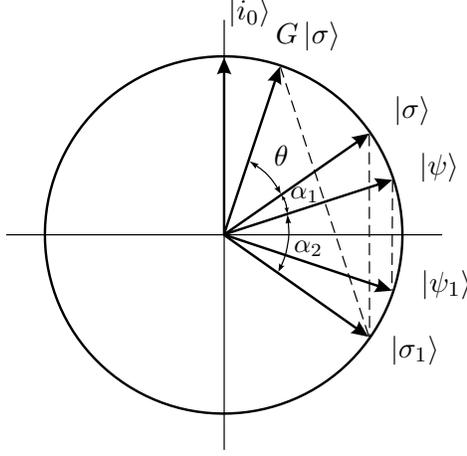


Figure 12: A generic vector $|\sigma\rangle$ is reflected around the horizontal axis by the application of U_f , yielding $|\sigma_1\rangle$. Then, the reflection of $|\sigma_1\rangle$ about the mean $|\psi\rangle$ gives $G|\sigma\rangle$, which is θ degrees closer to $|i_0\rangle$ (vertical axis).

making an angle α_1 with $|\psi\rangle$, as in Fig. 12. Let $|\sigma_1\rangle$ be the state of the first register after the application of U_f on $|\sigma\rangle|-\rangle$. U_f changes the sign of the component of $|\sigma\rangle$ in the direction of $|i_0\rangle$. So $|\sigma_1\rangle$ is the reflection of $|\sigma\rangle$ around the horizontal axis. Let α_2 be the angle between $|\psi\rangle$ and $|\sigma_1\rangle$. Let us show that $G|\sigma\rangle$ lies in the subspace spanned by $|i_0\rangle$ and $|\psi\rangle$:

$$\begin{aligned}
G|\sigma\rangle &= (2|\psi\rangle\langle\psi| - I)U_f|\sigma\rangle \\
&= 2\langle\psi|U_f|\sigma\rangle|\psi\rangle - |\sigma_1\rangle \\
&= 2\cos\alpha_2|\psi\rangle - |\sigma_1\rangle.
\end{aligned} \tag{20}$$

We have omitted the state $|-\rangle$ of the second register in the above calculation for simplicity. $|\sigma_1\rangle$ lies in the subspace spanned by $|i_0\rangle$ and $|\psi\rangle$, then $G|\psi\rangle$ also does.

Now we prove that the angle between $|\sigma\rangle$ and $G|\sigma\rangle$ is θ , which is the angle between $|\psi\rangle$ and $|\psi_1\rangle$ (see Fig. 12):

$$\begin{aligned}
\langle\sigma|G|\sigma\rangle &= 2\langle\sigma|\psi\rangle\cos\alpha_2 - \langle\sigma|\sigma_1\rangle \\
&= \cos\alpha_1\cos\alpha_2 - \cos(\alpha_1 + \alpha_2) \\
&= \cos(\alpha_2 - \alpha_1).
\end{aligned} \tag{21}$$

From Fig. 12 we see that $\alpha_2 - \alpha_1$ is θ . From Eq. (20) we see that $G|\sigma\rangle$ is a rotation of $|\sigma\rangle$ towards $|i_0\rangle$ by θ degrees.

The geometrical interpretation of the operator $2|\psi\rangle\langle\psi| - I$ is that it reflects any real vector around the axis defined by the vector $|\psi\rangle$. For example, in Fig. 12 we see that $G|\sigma\rangle = (2|\psi\rangle\langle\psi| - I)|\sigma_1\rangle$ is the reflection of $|\sigma_1\rangle$ around $|\psi\rangle$. $2|\psi\rangle\langle\psi| - I$ is called inversion about the mean for the following reason. Let $|\sigma\rangle = \sum_{i=0}^{2^n-1} \sigma_i|i\rangle$ be a generic

vector and define $\langle \sigma \rangle = \sum_{i=0}^{2^n-1} \sigma_i$ (mean of the amplitudes of $|\sigma\rangle$). Defining

$$|\sigma'\rangle = \sum_{i=0}^{2^n-1} (\sigma_i - \langle \sigma \rangle) |i\rangle, \quad (22)$$

results

$$(2|\psi\rangle\langle\psi| - I)|\sigma'\rangle = -|\sigma'\rangle. \quad (23)$$

The above equation shows that a vector with amplitudes $\sigma_i - \langle \sigma \rangle$ is transformed to a vector with amplitudes $-(\sigma_i - \langle \sigma \rangle)$. Note that $|\sigma'\rangle$ is not normalized, but this is irrelevant in the above argument because the amplitudes of $|\sigma\rangle$ and $|\sigma'\rangle$ only differ by a minus sign.

U_f also has a geometrical interpretation, which can be seen from the expression

$$U_f = I - 2|i_0\rangle\langle i_0|, \quad (24)$$

which yields

$$U_f|i\rangle = \begin{cases} |i\rangle, & \text{if } i \neq i_0 \\ -|i_0\rangle, & \text{if } i = i_0. \end{cases} \quad (25)$$

Therefore, the above representation for U_f is equivalent to Eq. (12) if we do not consider the state of the second register. The geometrical interpretation is: U_f reflects a generic vector about the plane orthogonal to $|i_0\rangle$. This is what Eq. (25) shows for vectors of the computational basis. The interpretation is valid for a generic vector because U_f is linear. We have not used Eq. (24) to define U_f before, because we do not know i_0 before running the algorithm. On the other hand, we assumed that it is possible somehow to use function f given by Eq. (10), and to build U_f as given by Eq. (11).

7 An Example: Grover for $N = 8$

We describe Grover's Algorithm for a search space of 8 elements. If $N = 8$ then $n = 3$. There are 3 qubits in the first register and 1 qubit in the second register. For $N = 8$, the operator G will be applied 2 times as we will see in Eq. (50). The circuit in this case is given in Fig. 13. The oracle is queried 2 times. Classically, an average of more than 4 queries are needed in order to have a probability of success of more than 1/2.

Let us describe the quantum computer state at each step shown in the circuit ($|\psi_0\rangle, |\psi\rangle, |\psi_1\rangle, |\psi_2\rangle, |\psi_3\rangle$, and $|\psi_f\rangle$). The initial state is

$$|\psi_0\rangle = |000\rangle. \quad (26)$$

After applying Hadamard gates,

$$|\psi\rangle = H^{\otimes 3}|000\rangle = (H|0\rangle)^{\otimes 3} = \frac{1}{2\sqrt{2}} \sum_{i=0}^7 |i\rangle. \quad (27)$$

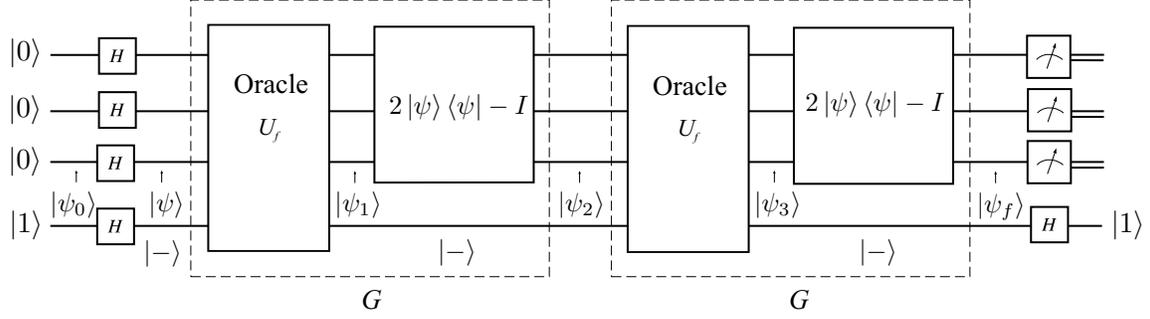


Figure 13: Grover's algorithm for $N = 8$.

Suppose that we are searching for the element with index 5. Since $|5\rangle = |101\rangle$,

$$\begin{aligned} U_f(|101\rangle|-\rangle) &= -|101\rangle|-\rangle \\ U_f(|i\rangle|-\rangle) &= |i\rangle|-\rangle, \text{ if } i \neq 5. \end{aligned} \quad (28)$$

Define $|u\rangle$ as

$$\begin{aligned} |u\rangle &= \frac{1}{\sqrt{7}} \sum_{\substack{i=0 \\ i \neq 5}}^7 |i\rangle \\ &= \frac{|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |110\rangle + |111\rangle}{\sqrt{7}}. \end{aligned} \quad (29)$$

Then

$$|\psi\rangle = \frac{\sqrt{7}}{2\sqrt{2}} |u\rangle + \frac{1}{2\sqrt{2}} |101\rangle. \quad (30)$$

With this result we can see the direction of $|\psi\rangle$ in Fig. 14. The value of θ is

$$\begin{aligned} \theta &= 2 \arccos\left(\frac{\sqrt{7}}{2\sqrt{2}}\right) \\ &= \arccos\left(\frac{3}{4}\right) \\ &\approx 41.4^\circ. \end{aligned} \quad (31)$$

The next step is

$$\begin{aligned} |\psi_1\rangle|-\rangle &= U_f(|\psi\rangle|-\rangle) \\ &= \left(\frac{|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle - |101\rangle + |110\rangle + |111\rangle}{2\sqrt{2}} \right) |-\rangle \end{aligned} \quad (32)$$

Note that $|101\rangle$ is the only state with a minus sign. We can write $|\psi_1\rangle$ as

$$|\psi_1\rangle = |\psi\rangle - \frac{1}{\sqrt{2}} |101\rangle \quad (33)$$

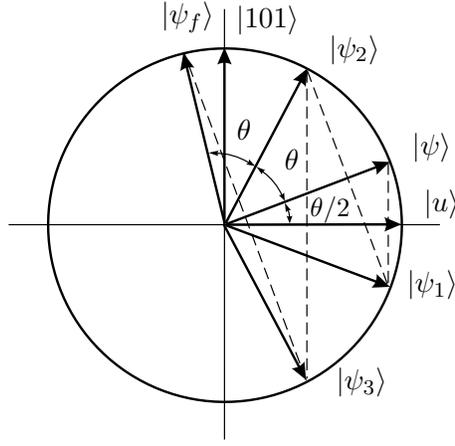


Figure 14: Intermediate states in Grover's algorithm for $N = 8$. Notice how close is $|\psi_f\rangle$ to $|101\rangle$, indicating a high probability that a measurement will give the searched element. The value of θ is around 41.4° .

or

$$|\psi_1\rangle = \frac{\sqrt{7}}{2\sqrt{2}}|u\rangle - \frac{1}{2\sqrt{2}}|101\rangle. \quad (34)$$

The form of Eq. (33) is useful in the next step of calculation, since we have to apply $2|\psi\rangle\langle\psi| - I$. The form of Eq. (34) is useful to draw $|\psi_1\rangle$ in Fig. 14. $|\psi_1\rangle$ is the reflection of $|\psi\rangle$ with respect to $|u\rangle$.

Next we calculate

$$|\psi_2\rangle = (2|\psi\rangle\langle\psi| - I)|\psi_1\rangle. \quad (35)$$

Using Eq. (33), we get

$$|\psi_2\rangle = \frac{1}{2}|\psi\rangle + \frac{1}{\sqrt{2}}|101\rangle \quad (36)$$

and, using Eq. (30),

$$|\psi_2\rangle = \frac{\sqrt{7}}{4\sqrt{2}}|u\rangle + \frac{5}{4\sqrt{2}}|101\rangle. \quad (37)$$

Let us confirm that the angle between $|\psi\rangle$ and $|\psi_2\rangle$ is θ :

$$\cos\theta = \langle\psi|\psi_2\rangle = \frac{1}{2}\langle\psi|\psi\rangle + \frac{1}{\sqrt{2}}\langle\psi|101\rangle = \frac{3}{4}, \quad (38)$$

which agrees with Eq. (31). This completes one application of G .

The second and last application of G is similar. $|\psi_3\rangle$ is given by

$$|\psi_3\rangle = \frac{\sqrt{7}}{2\sqrt{2}}|u\rangle - \frac{3}{4\sqrt{2}}|101\rangle. \quad (39)$$

Using Eq. (30), we have

$$|\psi_3\rangle = \frac{1}{2}|\psi\rangle - \frac{3}{2\sqrt{2}}|101\rangle. \quad (40)$$

$|\psi_3\rangle$ is the reflection of $|\psi_2\rangle$ with respect to $|u\rangle$.

The last step is

$$|\psi_f\rangle = (2|\psi\rangle\langle\psi| - I)|\psi_3\rangle. \quad (41)$$

Using Eqs. (30) and (40), we have

$$|\psi_f\rangle = -\frac{\sqrt{7}}{8\sqrt{2}}|u\rangle + \frac{11}{8\sqrt{2}}|101\rangle. \quad (42)$$

It is easy to confirm that $|\psi_f\rangle$ and $|\psi_2\rangle$ form an angle θ . Note that the amplitude of the state $|101\rangle$ is much bigger than the amplitude of any other state $|i\rangle$ ($i \neq 5$) in Eq. (42). This is the way most quantum algorithms work. They increase the amplitude of the states that carry the desired information. A measurement of the state $|\psi_f\rangle$ in the computational basis will project it into the state $|101\rangle$ with probability

$$p = \left| \frac{11}{8\sqrt{2}} \right|^2 \approx 0.945. \quad (43)$$

The chance of getting the result $|101\rangle$, which reads as number 5, is around 94.5%.

8 Generalization

The easiest way to calculate the output of Grover's Algorithm is to consider only the action of G instead of breaking the calculation into the action of the oracle (U_f) and the inversion about the mean. To this end, we choose $|i_0\rangle$ and $|u\rangle$ as the basis for the subspace where $|\psi\rangle$ rotates after successive applications of G . $|i_0\rangle$ is the searched state and $|u\rangle$ is defined as in Eq. (29),

$$\begin{aligned} |u\rangle &= \frac{1}{\sqrt{N-1}} \sum_{\substack{i=0 \\ i \neq i_0}}^{N-1} |i\rangle \\ &= \sqrt{\frac{N}{N-1}} |\psi\rangle - \frac{1}{\sqrt{N-1}} |i_0\rangle. \end{aligned} \quad (44)$$

From the first equation above we easily see that $\langle i_0|u\rangle = 0$, i.e., $|i_0\rangle$ and $|u\rangle$ are orthogonal. From the second equation we have

$$|\psi\rangle = \sqrt{1 - \frac{1}{N}} |u\rangle + \frac{1}{\sqrt{N}} |i_0\rangle. \quad (45)$$

The state of the quantum computer at each step is

$$G^k |\psi\rangle = \cos\left(\frac{2k+1}{2}\theta\right) |u\rangle + \sin\left(\frac{2k+1}{2}\theta\right) |i_0\rangle, \quad (46)$$

where we have dropped the state of the second register since it is $|-\rangle$ all the time. Eq.(46) is obtained from Fig. 15 after analyzing the components of $G^k |\psi\rangle$. The value of θ is obtained substituting k for 0 in Eq. (46) and comparing it with Eq. (45),

$$\theta = 2 \arccos \sqrt{1 - \frac{1}{N}}. \quad (47)$$

Eq.(46) expresses the fact we proved in section 6, that each application of G rotates the state of the first register by θ degrees towards $|i_0\rangle$. Fig. 15 shows successive applications of G .

The number of times k_0 that G must be applied obeys the equation

$$k_0 \theta + \frac{\theta}{2} = \frac{\pi}{2}. \quad (48)$$

Since k_0 must be integer, we write

$$k_0 = \text{round} \left(\frac{\pi - \theta}{2\theta} \right), \quad (49)$$

where θ is given by Eq. (47). If $N \gg 1$, by Taylor expanding Eq. (47), we get $\theta \approx 2/\sqrt{N}$ and from Eq. (49),

$$k_0 = \text{round} \left(\frac{\pi}{4} \sqrt{N} \right). \quad (50)$$

After applying G k_0 times, the probability p of finding the desired element after a measurement is

$$p = \sin^2 \left(\frac{2k_0 + 1}{2} \theta \right). \quad (51)$$

Fig. 16 shows p for n from 2 to 30. Recall that $N = 2^n$, so for $n = 30$ the search space has around 1 billion elements. For $n = 2$ the probability of getting the result is exactly 1. The reason for this is that Eq. (47) yields $\theta = \pi/3$ and $|\psi\rangle$ makes an angle $\pi/6$ with $|u\rangle$. Applying G one time rotates $|\psi\rangle$ to $|i_0\rangle$ exactly. For $n = 2$, Eq. (51) yields $p \approx 0.945$ which is the result (43) of the previous section.

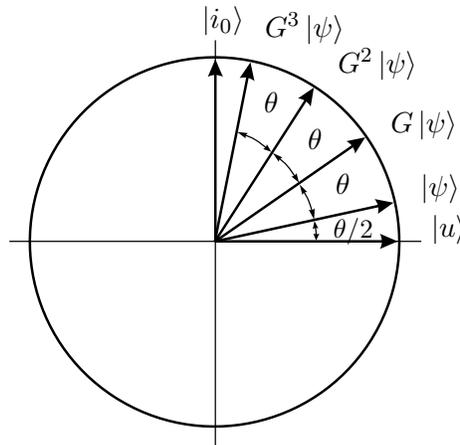


Figure 15: Effect of G on $|\psi\rangle$.

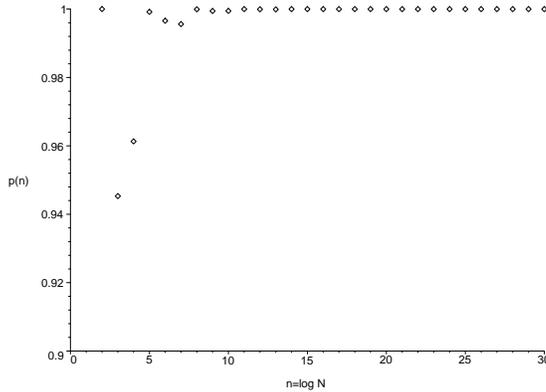


Figure 16: Probability of succeeding as a function of n .

9 Grover Operator in Terms of the Universal Gates

In this section we go in the opposite direction. We decompose G in terms of universal gates, which are CNOT and one-qubit gates. This decomposition shows how to implement G in practice. Let us begin by decomposing the inversion about the mean $2|\psi\rangle\langle\psi| - I$. Recall that

$$|\psi\rangle = H^{\otimes n} |0\rangle. \quad (52)$$

Then

$$2|\psi\rangle\langle\psi| - I = H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n}. \quad (53)$$

This equation shows that it is enough to consider the operator $2|0\rangle\langle 0| - I$, which inverts a generic vector about the vector $|0\rangle$. The circuit for it is given in Fig. 17. One can convince oneself that the circuit gives the correct output by following what happens to each state of the computational basis. The input $|0\rangle$ is the only one that does not change

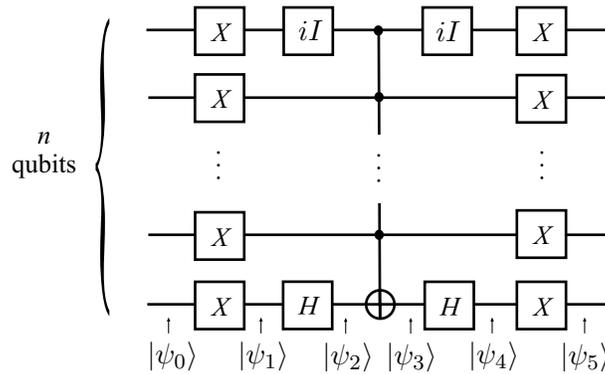


Figure 17: Circuit for $2|0\rangle\langle 0| - I$. Note the presence of the imaginary unit, which does not affect the real character of the operator.

signal. The intermediate states as shown in Fig. 17 are

$$\begin{aligned}
|\psi_0\rangle &= |0\rangle |0\rangle \dots |0\rangle |0\rangle \\
|\psi_1\rangle &= |1\rangle |1\rangle \dots |1\rangle |1\rangle \\
|\psi_2\rangle &= i |1\rangle |1\rangle \dots |1\rangle |-\rangle \\
|\psi_3\rangle &= i |1\rangle |1\rangle \dots |1\rangle (-|-\rangle) \\
|\psi_4\rangle &= -i(i |1\rangle) |1\rangle \dots |1\rangle |1\rangle \\
|\psi_5\rangle &= |0\rangle |0\rangle \dots |0\rangle |0\rangle.
\end{aligned} \tag{54}$$

The same calculations for the input $|j\rangle$, $0 < j < N$, results in $-|j\rangle$ as output.

The only operator in Fig. 17 that does not act on single qubits is the generalized Toffoli gate, which is shown alone in Fig. 18. The decomposition of the generalized Toffoli gate in terms of Toffoli gates is given in Fig. 19. The $n - 2$ work qubits are extra qubits whose input and output are known *a priori*. They are introduced in order to simplify the decomposition. A careful analysis of Fig. 19 shows that the output is the same of the generalized Toffoli gate with the extra work qubits.

The final step is the decomposition of the Toffoli gate, which is given in Fig. 20, where S is the phase gate

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \tag{55}$$

and T is the $\pi/8$ gate

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}. \tag{56}$$

This decomposition can be verified either by an exhaustive calculation of tensor products and operator compositions or by an exhaustive application of operators on basis elements.

By now one should be asking about the decomposition of U_f in terms of elementary gates. U_f has a different nature from other operators in Grover's algorithm, since its implementation depends on how data is loaded from a quantum memory of a quantum computer. On the other hand, we have pointed out that U_f can be represented by $I - 2|i_0\rangle\langle i_0|$ (Eq. (24)), if one knows the answer i_0 *a priori*. This representation is useful for simulating Grover's algorithm in a classical computer to test its efficiency. The operator $I - 2|i_0\rangle\langle i_0|$ is decomposed as a generalized Toffoli gate with n control qubits,

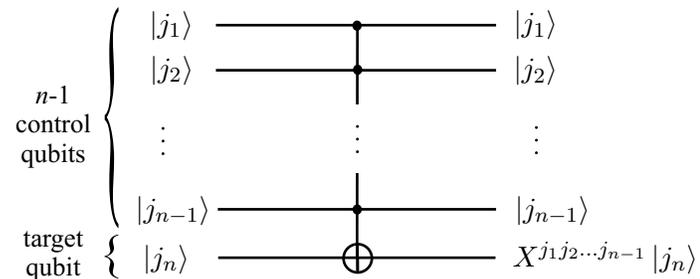


Figure 18: Generalized Toffoli gate.

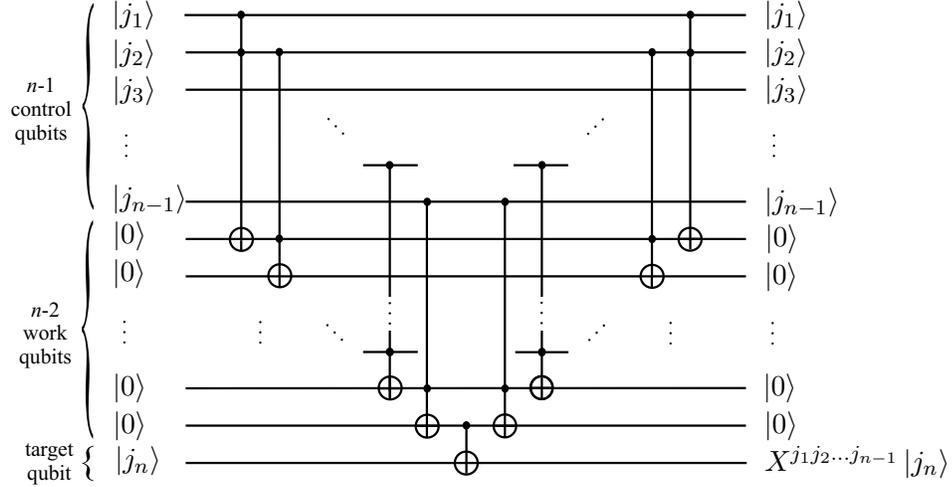


Figure 19: Decomposition of the generalized Toffoli gate in terms of Toffoli gates.

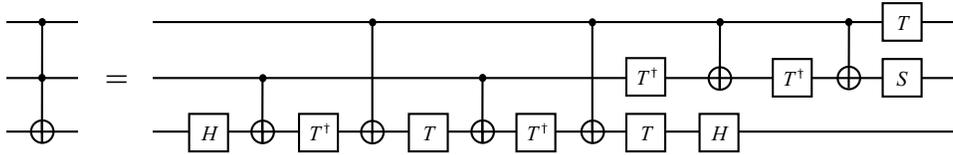


Figure 20: The Toffoli gate in terms of CNOT and one-qubit gates.

one target qubit in the state $|-\rangle$, and two symmetrical X gates in the i th qubit, if the i th binary digit of i_0 is 0. For example, the operator U_f used in section 7, for $N = 8$ (see Eq. (28)) is given in Fig. 21.

In section 1, we have pointed out that the efficiency of an algorithm is measured by how the number of elementary gates increases as a function of the number of qubits. Counting the number of elementary gates (Figs. 9, 10, 17, 19, and 20), and using Eq. (50), we get $\pi(17n - 15)\sqrt{2^n} + n + 2$, which yields complexity $O(n\sqrt{2^n})$, or equivalently $\tilde{O}(\sqrt{2^n})$. The notation $\tilde{O}(N)$ means $O(\text{poly}(\log(N))N)$.

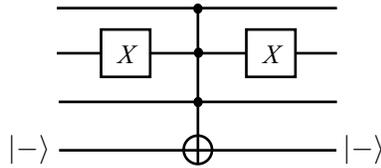


Figure 21: Decomposition of $I - 2|101\rangle\langle 101|$, which simulates U_f that searches number 5.

Acknowledgments

We thank the Group of Quantum Computation at LNCC, in particular, Drs. F. Haas and G. Girdali, and the students J.F. Abreu, D.C. Resende, and F. Marquezino. We thank also Drs. L. Davidovich and N. Zaguri for stimulating discussions on the subject.

References

- [1] P. Shor, Algorithms for Quantum Computation: Discrete Logarithm and Factoring, Proc. 35th Annual Symposium on Foundations of Computer Science (1994) 124-134.
- [2] L.M.K. Vandersypen, M. Steffen, G. Breyta, C.S. Yannoni, M.H. Sherwood, and I.L. Chuang, Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance, *Nature*, 414 (2001) 883-887.
- [3] L.K. Grover, A fast quantum mechanical algorithm for database search, Proc. 28th Annual ACM Symposium on the Theory of Computing (STOC), May (1996) 212-219.
- [4] L.K. Grover, Quantum Mechanics helps in searching for a needle in a haystack, *Phys. Rev. Lett.* **79** (1997) 325.
- [5] D. Deutsch and R. Jozsa, Rapid solution of problems by quantum computation, *Proc. R. Soc. London* **A439** (1992) 553-558.
- [6] E. Bernstein and U.V. Vazirani, Quantum Complexity Theory, Proc. 25th ACM Symp. on Theory of Computation, San Diego, CA, 1993, pp. 11-20 and *SIAM Journal on Computing* **26** (1997) 1411-1473.
- [7] D. Simon, On the power of quantum computation, Proc. 35th Annual Symposium on Foundations of Computer Science (1994) 116 and *SIAM Journal on Computing* **26** (1997) 1474-1483.
- [8] D. Aharonov, Quantum Computation, *Annual Reviews of Computational Physics*, ed. Dietrich Stauffer, World Scientific, vol. VI (1998).
- [9] M.A. Nielsen and I.L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge (2000).
- [10] J. Preskill, *Quantum Information and Computation*, Lecture Notes, California Institute of Technology (1998).
- [11] C.H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, Strengths and Weaknesses of Quantum Computing, *SIAM Journal on Computing* **26** (1997) 1510-1523.

- [12] M. Boyer, G. Brassard, P. Høyer and A. Tapp, Tight bounds on quantum searching, *Fortsch. Phys.* **46** (1998) 493-506.
- [13] G. Brassard, P. Høyer, and A. Tapp, Quantum Counting, [quant-ph/9805082](https://arxiv.org/abs/quant-ph/9805082).
- [14] C.H. Papadimitriou, *Computational Complexity*, Addison Wesley Pub. Co., Massachusetts (1994).
- [15] D. Deutsch, Quantum theory, the Church-Turing principle and the universal quantum computer, *Proc. R. Soc. London* **A400** (1985) 97-117.
- [16] A. Barenco, C.H. Bennett, R. Cleve, D.P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter, Elementary gates for quantum computation, *Phys. Rev.* **A52** (1995) 3457-3467.