An Introduction to Quantum Computing, Without the Physics

Giacomo Nannicini
IBM T.J. Watson, Yorktown Heights, NY
nannicini@us.ibm.com

Last updated: August 15, 2017.

Abstract

This paper is a gentle but rigorous introduction to quantum computing intended for computer scientists. Starting from a small set of assumptions on the behavior of quantum computing devices, we analyze their main characteristics, stressing the differences with classical computers, and finally describe two well-known algorithms (Simon's algorithm and Grover's algorithm) using the formalism developed in previous sections. This paper does not touch on the physics of the devices, and therefore does not require any notion of quantum mechanics.

1 Introduction

Quantum computing is a relatively new area of computing that has the potential to greatly speedup the solution of certain problems. However, quantum computers work in a fundamentally different way than classical computers. This introduction aims to explain the basic principles underpinning quantum computing. It assumes the reader is at ease with linear algebra, and with basic concepts in classical computing such as Turing machines, and algorithm complexity.

The literature contains many textbooks on quantum computing: a comprehensive reference is [Nielsen and Chuang, 2002], whereas more modern textbooks that are more accessible to non-physicists are [Mermin, 2007, Rieffel and Polak, 2011]. However, those books are time-consuming reads. There are not many short introductions that are truly accessible to non-physicists: [Rieffel and Polak, 2000] is noteworthy, as it actually uses very little physics.

The approach used in this work is, as far as I am aware, different from the literature in the sense that it abstracts *entirely* away from quantum physics: we study a quantum computing device starting from a small set of assumptions, and rigorously derive the remaining properties. The assumptions are verified in the real world because of the laws of quantum mechanics, but it is not necessary to understand why they hold: as long as we are willing to take a small leap of faith and believe that these assumptions are true, the rest will follow. The exposition in this work is more formal than in other surveys I am aware of, but for this reason, I like to think that it is also more mathematically precise.

The quantum computing device is, in abstract terms, similar to a classical computing device: it has a state, and the state of the device evolves according to certain operations. (It is possible to assume the presence of a tape and be more formal in defining a device that is the quantum equivalent of a Turing machine, but there is no need to do so for the purposes of this work.) This will be described in the next sections.

2 Qubits

To talk about quantum computers we must talk about qubits, that are the quantum counterpart of the bits found in classical computers: a classical computer has registers that are made up of bits, whereas a quantum computer has quantum registers that are made up of qubits.

2.1 Quantum state: basic definitions

Assumption 1 (**single-qubit version**). *The* state *of a single* qubit *is a unitary vector in* $\mathbb{C}^2$.

Remark 1. *If we pick the standard basis for* $\mathbb{C}^2$ *given by the vectors* $\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}$, *then a single qubit can be represented as* $\alpha\begin{pmatrix}1\\0\end{pmatrix} + \beta\begin{pmatrix}0\\1\end{pmatrix}$ *where* $\alpha, \beta \in \mathbb{C}$ *and* $|\alpha|^2 + |\beta|^2 = 1$.

To extend this definition to multiple qubits, it is necessary to use tensor products. We first review the basic definition and properties of the tensor product.

Definition 1. *Given two vector spaces* V *and* W *over a field* K *with bases* $e_1, \ldots, e_m$ *and* $f_1, \ldots, f_n$ *respectively, the* tensor product $V \otimes W$ *is another vector space over* K *of dimension* mn. *The tensor product space is equipped with a bilinear operation* $\otimes : V \times W \to V \otimes W$. *The vector space* $V \otimes W$ *has basis* $e_i \otimes f_j \; \forall i = 1, \ldots, m, j = 1, \ldots, n$.

If we choose the standard basis in the origin vector spaces, then the tensor product is none other than the Kronecker product, which is itself a generalization of the outer product. This is formalized next.

Definition 2. *Given* $A \in \mathbb{C}^{m\times n}, B \in \mathbb{C}^{p\times q}$, *the* Kronecker product $A \otimes B$ *is the matrix* $D \in \mathbb{C}^{mp\times nq}$ *defined as:*

$$D := A \otimes B = \begin{pmatrix} a_{11}B & \ldots & a_{1n}B \\ a_{21}B & \ldots & a_{2n}B \\ \vdots & & \vdots \\ a_{m1}B & \ldots & a_{mn}B \end{pmatrix}.$$

If we choose the standard basis over the vector spaces $\mathbb{C}^{m\times n}$ *and* $\mathbb{C}^{p\times q}$, *then the bilinar operation* $\otimes$ *of the tensor product* $\mathbb{C}^{m\times n} \otimes \mathbb{C}^{p\times q}$ *is simply the Kronecker product.*

In this paper, we always work with complex Hilbert spaces $\mathbb{H}$ of the form $\mathbb{C}^n$, using the standard basis. With a slight but common abuse of notation, we will therefore use tensor product to refer to the Kronecker and outer products. We now assume that $V \equiv \mathbb{C}^m, W \equiv \mathbb{C}^n$.

Proposition 1. *Let* $A, B : \mathbb{C}^{m\times m}, C, D \in \mathbb{C}^{n\times n}$ *be linear transformations on* V *and* W *respectively,* $u, v \in C^m, w, x \in \mathbb{C}^n$, *and* $a, b \in \mathbb{C}$. *The tensor product satisfies the following properties:*

(i) $(A \otimes C)(B \otimes D) = AB \otimes CD$.

(ii) $(A \otimes C)(u \otimes w) = Au \otimes Cw$.

(iii) $(u + v) \otimes w = u \otimes w + v \otimes w$.

(iv) $u \otimes (w + x) = u \otimes w + u \otimes x$.

(v) $au \otimes bw = abu \otimes w$.

(vi) $(A \otimes C)^* = A^* \otimes C^*$.

Above and in the following, the notation A^* denotes the conjugate transpose of A, and given a matrix A, the notation $A^{\otimes q}$ indicates the tensor product of A with itself q times: $A^{\otimes q} := \underbrace{A \otimes A \cdots \otimes A}_{q \text{ times}}$. The same notation will be used for a vector: $x^{\otimes q} := \underbrace{x \otimes x \cdots \otimes x}_{q \text{ times}}$.

When considering multiple qubits, we have to deal with bases on spaces of different dimensions, and this requires some additional notation. Quantum physics uses the *bra-ket* notation. There is an undeniable advantage in the quantum notation in that it puts the most important information in the center of the symbols, rather than relegate it to a marginal role in the subscript or superscript.

Definition 3. *For any integers $q > 0$ and $0 \leq j \leq 2^q - 1$, we denote by $jB_q \in \{0,1\}^q$ the vector containing the binary representation of j on q digits.*

Definition 4. *Given a Hilbert space $\mathbb{H} \equiv \mathbb{C}^n$, a quantity $\psi \in \mathbb{H}$ enclosed in a* ket*, denoted $|\psi\rangle$, is a vector and can be thought of as a column vector. A quantity $\phi \in \mathbb{H}^*$ enclosed in a* bra*, denoted $\langle\phi|$, is a vector in the dual space, and can be thought of as a row vector that is the conjugate transpose of $\phi \in \mathbb{H}$. The standard basis for $\mathbb{C}^2$ is denoted by $|0\rangle_1 = \begin{pmatrix}1\\0\end{pmatrix}, |1\rangle_1 = \begin{pmatrix}0\\1\end{pmatrix}$.*

Thus, an expression such as $\langle\psi|\phi\rangle$ is an inner product in the Hilbert space. We can now properly define the state of a quantum register of size q, restating Assumption 1 in more general form.

Assumption 1. *The state of q qubits is a unitary vector in $\left(\mathbb{C}^2\right)^{\otimes q} = \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2$.*

Remark 2. *Given the standard basis for each $\mathbb{C}^2$, a basis for $\left(\mathbb{C}^2\right)^{\otimes q}$ is given by:*

$$
\begin{aligned}
|0\rangle_q &= \underbrace{|0\rangle_1 \otimes \cdots \otimes |0\rangle_1}_{q \text{ times}} = |0B_q\rangle \\
|1\rangle_q &= \underbrace{|0\rangle_1 \otimes \cdots \otimes |1\rangle_1}_{q \text{ times}} = |1B_q\rangle \\
&\vdots \\
|2^q - 1\rangle_q &= \underbrace{|1\rangle_1 \otimes \cdots \otimes |1\rangle_1}_{q \text{ times}} = |(2^q - 1)B_q\rangle.
\end{aligned}
$$

The state of q qubits can be represented as: $|\psi\rangle = \sum_{j=0}^{2^q-1} \alpha_j |j\rangle_q$, with $\alpha_j \in \mathbb{C}$ and $\sum_{j=1}^{2^q-1} |\alpha_j|^2 = 1$.

It is important to notice that $\left(\mathbb{C}^2\right)^{\otimes q}$ is a 2^q-dimensional space. This is in sharp contrast with the state of classical bits: given q *classical* bits, their state is a binary string in $\{0,1\}^q$, which is a q-dimensional space. In other words, the dimension of the state space of quantum registers grows exponentially in the number of qubits, whereas the dimension of the state space of classical registers grows linearly in the number of bits.

We now define more formally our ket notation for basis vectors.

- When $x \in \{0,1\}$, $|x\rangle_1$ denotes the corresponding basis vector in $\mathbb{C}^2$, and is generally denoted simply by $|x\rangle$. In other words, if the subscript for the ket is omitted, it is intended to be 1, so that $|0\rangle = |0\rangle_1$ and $|1\rangle = |1\rangle_1$.
- When x is any integer $\leq 2^1 - 1$, $|x\rangle_q$ is the 2^q-dimensional basis vector $|xB_q\rangle$, i.e. the basis vector in which x is expressed as a binary string on q digits.
- When $x_1, \ldots, x_q$ are binary digits, $|x_1 x_2 \ldots x_q\rangle$ is the 2^q-dimensional basis vector corresponding to the binary string $x_1 x_2 \ldots x_q$. In other words, a sequence of single digits enclosed in $|\rangle$ should be interpreted as a binary string, rather than a "regular" product. If a product inside $|\rangle$ is necessary, we will denote it by $\cdot$.

To provide an example of this notation below.

Example 1. *Let us write the basis elements of* $\mathbb{C}^2 \otimes \mathbb{C}^2$*:*

$$|0\rangle_2 = |0\rangle \otimes |0\rangle = |00\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \qquad |1\rangle_2 = |0\rangle \otimes |1\rangle = |01\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}$$

$$|2\rangle_2 = |1\rangle \otimes |0\rangle = |10\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \qquad |3\rangle_2 = |1\rangle \otimes |1\rangle = |11\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}.$$

Thus, the index in the standard basis of the basis element corresponding to tensor product of basis elements of $\mathbb{C}^2$ is given simply by the decimal number corresponding to the binary string obtained by juxtaposing the index of the basis elements of $\mathbb{C}^2$.

2.2 Basis states and superposition

Definition 5. *We say that q qubits are in a* basis state *if their state* $|\psi\rangle = \sum_{j=0}^{2^q-1} \alpha_j |j\rangle_q$ *is such that* $\exists k : \alpha_k = 1$, $\alpha_j = 0\ \forall j \neq k$*. Otherwise, we say that they are in a* superposition.

Example 2. *Consider two qubits:*

$$|x\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$$
$$|y\rangle = \beta_0|0\rangle + \beta_1|1\rangle.$$

Then, the two qubits taken as a whole will be in state:

$$|x\rangle \otimes |y\rangle = \alpha_0\beta_0|0\rangle \otimes |0\rangle + \alpha_0\beta_1|0\rangle \otimes |1\rangle + \alpha_1\beta_0|1\rangle \otimes |0\rangle + \alpha_1\beta_1|1\rangle \otimes |1\rangle.$$

If both $|x\rangle$ *and* $|y\rangle$ *are in a basis state, we have that either* α_0 *or* α_1 *is zero, and similarly either* β_0 *or* β_1 *is zero, while the nonzero coefficients have modulus one. Thus, only one of the coefficients in the expression of the state of* $|x\rangle \otimes |y\rangle$ *is nonzero, and in fact its modulus is one: all other coefficients are zero. This implies that if both* $|x\rangle$ *and* $|y\rangle$ *are in a basis state,* $|x\rangle \otimes |y\rangle$ *is in a basis state as well. But now assume that* $\alpha_0 = \beta_0 = \alpha_1 = \beta_1 = \frac{1}{\sqrt{2}}$*: the qubits* $|x\rangle$ *and* $|y\rangle$ *are in a superposition. Then the state of* $|x\rangle \otimes |y\rangle$ *is* $\frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle$*, which is a superposition as well. Notice that the normalization of the coefficients works out, as one can easily check with simple algebra: the tensor product of unitary vectors is unitary.*

The example clearly generalizes to an arbitary number of qubits. In fact the following proposition is trivially true:

Proposition 2. *For any q, q qubits are in a basis state if and only if each of the individual qubits is in a basis state.*

Notice that superposition does not have a classical equivalent: q classical bits are always in a basis state, i.e., the q bits will always correspond exactly to one of the 2^q binary strings representing the numbers $0, \ldots, 2^q - 1$. Indeed, superposition is one of the main features of quantum computers that differentiates them from classical computers. The second important feature is entanglement, that will be discussed next.

2.3 Product states and entanglement

We have seen that the state of q-qubits is a vector in $\left(\mathbb{C}^2\right)^{\otimes q}$, which is a 2^q dimensional space. Since this is a tensor product of $\mathbb{C}^2$, i.e., the space in which single qubits live, it is natural to ask whether moving from single qubits to multiple qubits gained us anything at all. In other words, we want to investigate whether the quantum states that are representable on q qubits are simply the tensor product of q single qubits. We can answer this question by using the definitions given above. The state of q qubits is a unitary vector in $\left(\mathbb{C}^2\right)^{\otimes q}$, and it can be represented as:

$$|\psi\rangle = \sum_{j=0}^{2^q-1} \alpha_j |j\rangle_q, \qquad \sum_{j=0}^{2^q-1} |\alpha_j|^2 = 1.$$

Now let us consider the tensor product of q qubits, the j-th of which is in state $\beta_{j,0}|0\rangle + \beta_{j,1}|1\rangle$. Taking the tensor product we obtain the vector:

$$|\phi\rangle = \sum_{j_{q-1}=0}^{1} \sum_{j_{q-2}=0}^{1} \cdots \sum_{j_0=1}^{1} \prod_{k=0}^{q-1} \beta_{k,j_k} |j_q j_{q-1} \ldots j_0\rangle = \sum_{j=0}^{2^q-1} \prod_{k=1}^{q} \beta_{k,(jB_q)_k} |jB_q\rangle,$$
$$|\beta_{j,0}|^2 + |\beta_{j,1}|^2 = 1 \; \forall j = 1, \ldots, q.$$

The normalization condition for $|\phi\rangle$ implies that $\sum_{j_{q-1}=0}^{1} \sum_{j_{q-2}=0}^{1} \cdots \sum_{j_0=1}^{1} \prod_{k=0}^{q-1} |\beta_{k,j_k}|^2 = 1$, but it is more restrictive than that of $|\psi\rangle$. That is, there are values for α_j with $\sum_{j=1}^{2^q-1} |\alpha_j|^2 = 1$ that cannot be expressed as coefficients satisfying the conditions for $|\phi\rangle$.

This is easily clarified with an example using two qubits:

$$|x\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$$
$$|y\rangle = \beta_0|0\rangle + \beta_1|1\rangle.$$

that taken as a whole will be in state:

$$|x\rangle \otimes |y\rangle = \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle,$$

with the normalization conditions $|\alpha_0|^2 + |\alpha_1|^2 = 1$ and $|\beta_0|^2 + |\beta_1|^2 = 1$. The general state of a 2-qubit register $|\psi\rangle$ is:

$$|\psi\rangle = \gamma_{00}|00\rangle + \gamma_{01}|01\rangle + \gamma_{10}|10\rangle + \gamma_{11}|11\rangle,$$

with normalization condition $|\gamma_{00}|^2 + |\gamma_{01}|^2 + |\gamma_{10}|^2 + |\gamma_{11}|^2 = 1$. Comparing the two expressions, we see that a $|\psi\rangle$ is of the form $|x\rangle \otimes |y\rangle$ if and only if it satisfies the relationship:

$$\gamma_{00}\gamma_{11} = \gamma_{01}\gamma_{10}.$$

Clearly $|x\rangle \otimes |y\rangle$ yields coefficients that satisfy this condition. To see the converse, let $\theta_{00}, \theta_{01}, \theta_{10}, \theta_{11}$ be the phases of $\gamma_{00}, \gamma_{01}, \gamma_{10}, \gamma_{11}$. Notice that $\gamma_{00}\gamma_{11} = \gamma_{01}\gamma_{10}$ implies:

$$|\gamma_{00}|^2|\gamma_{11}|^2 = |\gamma_{01}|^2|\gamma_{10}|^2$$
$$\theta_{00} + \theta_{11} = \theta_{01} + \theta_{10}.$$

Then we can write:

$$\begin{aligned} |\gamma_{00}| &= \sqrt{|\gamma_{00}|}^2 = \sqrt{|\gamma_{00}|^2(|\gamma_{00}|^2 + |\gamma_{01}|^2 + |\gamma_{10}|^2 + |\gamma_{11}|^2)} \\ &= \sqrt{|\gamma_{00}|^4 + |\gamma_{00}|^2|\gamma_{01}|^2 + |\gamma_{00}|^2|\gamma_{10}|^2 + |\gamma_{01}|^2|\gamma_{10}|^2} \\ &= \underbrace{\sqrt{|\gamma_{00}|^2 + |\gamma_{01}|^2}}_{|\alpha_0|} \underbrace{\sqrt{|\gamma_{00}|^2 + |\gamma_{10}|^2}}_{|\beta_0|}, \end{aligned}$$

and similarly for other coefficients:

$$|\gamma_{01}| = \underbrace{\sqrt{|\gamma_{00}|^2 + |\gamma_{01}|^2}}_{|\alpha_0|}\underbrace{\sqrt{|\gamma_{01}|^2 + |\gamma_{11}|^2}}_{|\beta_1|}$$
$$|\gamma_{10}| = \underbrace{\sqrt{|\gamma_{10}|^2 + |\gamma_{11}|^2}}_{|\alpha_1|}\underbrace{\sqrt{|\gamma_{00}|^2 + |\gamma_{10}|^2}}_{|\beta_0|}$$
$$|\gamma_{11}| = \underbrace{\sqrt{|\gamma_{10}|^2 + |\gamma_{11}|^2}}_{|\alpha_1|}\underbrace{\sqrt{|\gamma_{01}|^2 + |\gamma_{11}|^2}}_{|\beta_1|}.$$

To fully define the coefficients $\alpha_0, \alpha_1, \beta_0, \beta_1$ we must determine their phases. We can assign:

$$\alpha_0 = e^{i\theta_{00}}|\alpha_0|, \qquad \alpha_1 = e^{i\theta_{10}}|\alpha_1|, \qquad \beta_0 = |\beta_0|, \qquad \beta_1 = e^{i(\theta_{01}-\theta_{00})}|\beta_1|.$$

It is now easy to verify that the state $|\psi\rangle$ can be expressed as $|x\rangle \otimes |y\rangle$ with coefficients $\alpha_0, \alpha_1, \beta_0, \beta_1$ as derived from the expressions above.

Definition 6. *A quantum state $|\psi\rangle \in \left(\mathbb{C}^2\right)^{\otimes q}$ is a* decomposable *if it can be expressed as a tensor product $|\psi_1\rangle \otimes \cdots \otimes |\psi_k\rangle$ of $k > 2$ quantum states on $q_1, \ldots, q_k$ qubits respectively, with the property that $q_1 + \cdots + q_k = q$.*

Definition 7. *A quantum state $|\psi\rangle \in \left(\mathbb{C}^2\right)^{\otimes q}$ is a* product state *if it is decomposable into the tensor product of q single-qubit quantum states. Otherwise, it is* entangled.

Notice that a general quantum state $|\psi\rangle$ could be the product of two or more lower-dimensional quantum state, e.g., $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$, with $|\psi_1\rangle$ and $|\psi_2\rangle$ being entangled states. In such a situation, $|\psi\rangle$ still exhibits some entanglement, but in some sense it can still be "simplified". Generally, according to the definition above, a quantum state is called entangled as long as it cannot be fully decomposed.

Example 3. *Consider the following 2-qubit state:*

$$\frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle.$$

This is a product state because it is equal to $\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)$. On the other hand, the following 2-qubit state:

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$$

is an entangled state, because it cannot be expressed as a product of two single-qubit states.

3 Operations on qubits

Operations on quantum states must satisfy certain conditions, to ensure that applying an operation does not break the basic properties of the quantum state.

Assumption 2. *An* operation *applied by a quantum computer with q qubits, also called a* gate, *is a unitary matrix in $\mathbb{C}^{2^q \times 2^q}$.*

Remark 3. *A matrix U is unitary if $U^*U = UU^* = I$.*

A well-known property of unitary matrices is that they are norm-preserving; that is, given a unitary matrix U and a vector x, $\|Ux\| = \|x\|$. Thus, for a q-qubit system, the quantum state is a unitary vector $\psi \in \mathbb{C}^{2^q}$, a quantum operation is a matrix $U \in \mathbb{C}^{2^q \times 2^q}$, and the application of U onto the state ψ is the unitary vector $U\psi \in \mathbb{C}^{2^q}$. This leads to the following remarks:

- Quantum operations are *linear*.
- Quantum operations are *reversible*.

While these properties may initially seem to be extremely restrictive, [Deutsch, 1985] shows that a universal quantum computer is Turing-complete, implying that it can simulate any Turing-computable function with an additional polynomial amount of space, given sufficient time. Out of the two properties indicated above, the most counterintuitive is perhaps reversibility: the classical notion of computation is typically not reversible, because memory can be erased and, in the classical Turing machine model of computation, symbols are erased from the tape. However, [Bennett, 1973] shows that computations can be made reversible by means of extra space. We will see how this is performed in quantum computers, but in order to do that, we need to introduce some notation about quantum circuits.

3.1 Notation for quantum circuits

A quantum circuit is represented by indicating which operations are performed on each qubit, or group of qubits. For a quantum computer with q qubits, we represent q qubit lines, where the top line indicates qubit 1 and the rest are given in ascending order. Operations are represented as gates; from now, the two terms are used interchangeably. Gates take qubit lines as input, have the same number of qubit lines as output, and apply the unitary matrix indicated on the gate to the quantum state of those qubits. Figure 1 is a simple example.

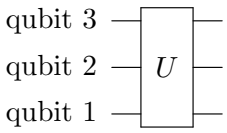


Figure 1: A simple quantum circuit.

Note that circuit diagrams are read from left to right, but because each gate corresponds to applying a matrix to the quantum state, the matrices corresponding to the gates should be written from right to left in the mathematical expression describing the circuit. For example, in the circuit in Figure 2, the outcome of the circuit is the state $BA|\psi\rangle$, because we start with

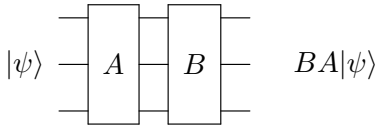


Figure 2: Order of the operations in a quantum circuit.

state $|\psi\rangle$, and we first apply the gate with unitary matrix A, and then B.

Gates can also be applied to individual gates. Because a single qubit is a vector in $\mathbb{C}|^2$, a single-qubit gate is a unitary matrix in $\mathbb{C}^{2\times 2}$. Consider the same 3-qubit device, and suppose we want to apply the gate only to the first qubit. We would write it as in Figure 3
From an algebraic point of view, the action of first example on the quantum state is clear: the state of the three qubits is mapped onto another 3-qubit state, as U acts on all the qubits. To

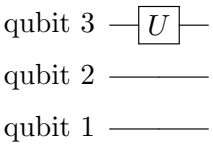


Figure 3: A circuit with a single-qubit gate.

understand the second example, we must imagine that an identity gate is applied to all the empty qubit lines. Therefore, the second example can be thought of as indicated in Figure 4.

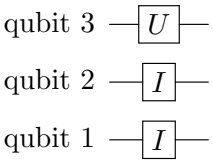


Figure 4: Equivalent representation of a circuit with a single-qubit gate.

This circuit can be interpreted as applying the $U \otimes I \otimes I$ to the 3-qubit state $|\psi\rangle$. Notice that by convention U, applied to qubit 3, appears in the leftmost term of the tensor product, because the basis for $\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2$ has elements $|000\rangle, |001\rangle, \ldots, |111\rangle$ where by convention we label qubit 3 the leftmost qubit. In particular, if we have a product state $|x\rangle \otimes |y\rangle \otimes |z\rangle$, we can write labels as indicated in Figure 5.

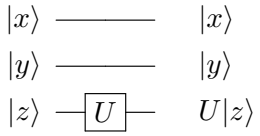


Figure 5: Effect of a single-qubit gate on a product state.

This is because $(I \otimes I \otimes U)(|x\rangle \otimes |y\rangle \otimes |z\rangle) = |x\rangle \otimes |y\rangle \otimes U|z\rangle$. If the system is in an entangled state, however, the action of $(I \otimes I \otimes U)$ cannot be determined in such a simple way, because the state cannot be factored as a product state. Thus, for a general entangled state, the effect of the circuit is as indicated in Figure 6. Notice that this fact is essentially the reason why simulation

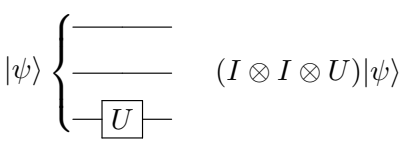


Figure 6: Effect of a single-qubit gate on an entangled state.

of quantum computations on classical computers takes exponential space: to simulate the effect even of a single-qubit gate on the entangled state $|\psi\rangle$, we have to explicitly compute the $2^q \times 2^q$ matrix $(I \otimes I \otimes U)$, and then apply it to $|\psi\rangle$. As long as the quantum state is not entangled computations can be carried out on each qubit independently, but entanglement requires us to keep track of the full quantum state in 2^q-dimensional complex space, leading to large amounts of memory required.

3.2 Input-output, and measurement gates

We will first note that by convention, the initial quantum state of the quantum computing device is $|0\rangle_q$. Any algorithm will have to start from this state. Of course, this does not prevent

the algorithm to modify the state and transform it into a more suitable one. Examples of how this can be done will be seen in subsequent sections. The important part is that if there is any data that has to be fed to the algorithm, this data will take the form of a circuit that performs some operation on the quantum state. Therefore, the *input* to a quantum computing device is a circuit, or a set of circuits, which are then combined in an algorithm: the algorithm may be self-contained in the quantum computer, or it may involve an external, classical computing device that uses quantum computations as a subtroutine. But what is the *output* of the quantum computer?

So far we characterized properties of quantum states and quantum gates. Remarkably, the state of a q-qubit quantum device has dimension 2^q, exponentially larger than the dimension of q classical bits. However, there is a catch: in a classical computer we can simply read the state of the bits, whereas in a quantum computer we do not have direct, unrestricted access to the quantum state. Information on the quantum state is only gathered through a measurement gate, indicated in the circuit diagram in Figure 7.

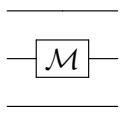


Figure 7: Single-qubit measurement.

We now formally define the effect of a single-bit measurement.

Assumption 3. *Information on the state of a quantum computing device can only be obtained through a* measurement. *Given a q-qubit quantum state $|\psi\rangle = \sum_{j=0}^{2^q-1} \alpha_j |j\rangle_q$, a* measurement gate *on qubit k outputs 0 with probability $\sum_{j:(jB_q)_k=0} |\alpha_j|^2$, and 1 with probability $\sum_{j:(jB_q)_k=1} |\alpha_j|^2$. Let $x \in \{0,1\}$ be the measured value. After the measurement, the quantum state becomes:*

$$\sum_{j:(jB_q)_k=x} \frac{\alpha_j}{\sqrt{\sum_{j:(jB_q)_k=x} |\alpha_j|^2}} |j\rangle_q.$$

The original quantum state is no longer recoverable.

Remark 4. *The state of the quantum system after a measurement collapses to a linear combination of only those basis states that are consistent with the outcome of the measurement.*

We can now show the following.

Proposition 3. *Given a q-qubit quantum state $|\psi\rangle = \sum_{j=0}^{2^q-1} \alpha_j |j\rangle_q$, measuring the q qubits yields jB_q with probability $|\alpha_j|^2$, for $j = 0, \ldots, 2^q - 1$.*

Proof. We need to show that the probability of observing jB_q after q single-qubit measurements is equal to $|\alpha_j|^2$. We can do this by induction on q. The case $q = 1$ is trivial. We now show how to go from $q-1$ to q. In terms of notation, we will write $\Pr(i \overset{M}{=} x)$ to denote the probability that the measurement of qubit i yields $x \in \{0,1\}$. If it is important to indicate the quantum state on which the measurement is performed, we denote it as $\Pr_{|\psi\rangle}(i \overset{M}{=} x)$.

Suppose the qubits are measured in an abitrary order $\pi(1), \ldots, \pi(q)$. After measuring the first qubit, the quantum state becomes:

$$|\phi\rangle = \sum_{k:(kB_q)_{\pi(1)}=(jB_q)_{\pi(1)}} \frac{\alpha_k}{\sqrt{\sum_{k:(kB_q)_{\pi(1)}=(jB_q)_{\pi(1)}} |\alpha_k|^2}} |k\rangle_q := \sum_{k:(kB_q)_{\pi(1)}=(jB_q)_{\pi(1)}} \beta_k |k\rangle_q,$$

where the coefficients β_k, as given above, are only defined for $k : (kB_q)_{\pi(1)} = (jB_q)_{\pi(1)}$. Regarding the probability of measuring the outcomes, we can write:

$$\Pr_{|\psi\rangle}\left(\pi(1) \stackrel{M}{=} (jB_q)_{\pi(1)}, \ldots, \pi(q) \stackrel{M}{=} (jB_q)_{\pi(q)}\right) =$$

$$\Pr_{|\psi\rangle}\left(\pi(2) \stackrel{M}{=} (jB_q)_{\pi(2)}, \ldots, \pi(q) \stackrel{M}{=} (jB_q)_{\pi(q)} | \pi(1) \stackrel{M}{=} (jB_q)_{\pi(1)}\right) \Pr_{|\psi\rangle}\left(\pi(1) \stackrel{M}{=} (jB_q)_{\pi(1)}\right) =$$

$$\Pr_{|\phi\rangle}\left(\pi(2) \stackrel{M}{=} (jB_q)_{\pi(2)}, \ldots, \pi(q) \stackrel{M}{=} (jB_q)_{\pi(q)}\right) \Pr_{|\psi\rangle}\left(\pi(1) \stackrel{M}{=} (jB_q)_{\pi(1)}\right).$$

By the definition of single-qubit measurement, we have:

$$\Pr_{|\psi\rangle}\left(\pi(1) \stackrel{M}{=} (jB_q)_{\pi(1)}\right) = \sum_{k:(kB_q)_{\pi(1)}=(jB_q)_{\pi(1)}} |\alpha_k|^2.$$

By the induction hypothesis:

$$\Pr_{|\phi\rangle}\left(\pi(2) \stackrel{M}{=} (jB_q)_{\pi(2)}, \ldots, \pi(q) \stackrel{M}{=} (jB_q)_{\pi(q)}\right) =$$

$$\sum_{k:(kB_q)_{\pi(h)}=(jB_q)_{\pi(h)}\ \forall h=1,\ldots,q} |\beta_k|^2 = \sum_{k:(kB_q)_{\pi(h)}=(jB_q)_{\pi(h)}\ \forall h=1,\ldots,q} \frac{|\alpha_k|^2}{\sum_{k:(kB_q)_{\pi(1)}=(jB_q)_{\pi(1)}} |\alpha_k|^2},$$

where the summation is only over indices for which $(kB_q)_{\pi(1)} = (jB_q)_1$ by definition of $|\phi\rangle$. Carrying out the multiplication, we obtain:

$$\Pr_{|\psi\rangle}\left(\pi(1) \stackrel{M}{=} (jB_q)_{\pi(1)}, \ldots, \pi(q) \stackrel{M}{=} (jB_q)_{\pi(q)}\right) =$$

$$\sum_{k:(kB_q)_{\pi(h)}=(jB_q)_{\pi(h)}\ \forall h=1,\ldots,q} \frac{|\alpha_k|^2}{\sum_{k:(kB_q)_{\pi(1)}=(jB_q)_{\pi(1)}} |\alpha_k|^2} \sum_{k:(kB_q)_{\pi(1)}=(jB_q)_{\pi(1)}} |\alpha_k|^2 =$$

$$\sum_{k:(kB_q)_h=(jB_q)_h\ \forall h=1,\ldots,q} |\alpha_k|^2 = |\alpha_j|^2.$$

□

The proposition above shows that the two circuits in Figure 8 are equivalent.

Figure 8: Multiple-qubit measurement.

In other words, the single-qubit measurement gate is sufficient to measure any number of qubits in the most natural way, i.e., the measurement outcomes on the q qubits occur with probability that is exactly equal to the square of the state coefficients α_j.

Example 4. *Consider again the following 2-qubit state:*

$$\alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle = \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle.$$

We remarked that this is a product state. Let qubit QR be the qubit on the right (i.e., the second digit in the two-digit binary strings), and qubit QL the qubit on the left (i.e., the first digit in the two-digit binary strings). Then:

$$\Pr(QL \stackrel{M}{=} 0) = |\alpha_{00}|^2 + |\alpha_{01}|^2 = \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 = \frac{1}{2}$$
$$\Pr(QL \stackrel{M}{=} 1) = |\alpha_{10}|^2 + |\alpha_{11}|^2 = \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 = \frac{1}{2}$$
$$\Pr(QR \stackrel{M}{=} 0) = |\alpha_{00}|^2 + |\alpha_{10}|^2 = \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 = \frac{1}{2}$$
$$\Pr(QR \stackrel{M}{=} 1) = |\alpha_{01}|^2 + |\alpha_{11}|^2 = \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 = \frac{1}{2}.$$

Suppose we measure QR and we obtain 1 as the outcome of the measurement. Then the state of the 2-qubit systems collapses to:

$$\frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|11\rangle.$$

If we measure QL from this modified state, we obtain:

$$\Pr(QL \stackrel{M}{=} 0) = \frac{1}{2} \qquad \Pr(QL \stackrel{M}{=} 1) = \frac{1}{2}.$$

Hence, the probability of measuring 0 or 1 from qubit QL did not change after the measurement.

Consider now the followin entangled 2-qubit state:

$$\beta_{00}|00\rangle + \beta_{11}|11\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle.$$

Doing the calculations, we still have:

$$\Pr(QL \stackrel{M}{=} 0) = |\beta_{00}|^2 = \frac{1}{2} \qquad \Pr(QL \stackrel{M}{=} 1) = |\beta_{11}|^2 = \frac{1}{2}$$
$$\Pr(QR \stackrel{M}{=} 0) = |\beta_{00}|^2 = \frac{1}{2} \qquad \Pr(QR \stackrel{M}{=} 1) = |\beta_{11}|^2 = \frac{1}{2}.$$

Suppose we measure qubit QR and we obtain 1 as the outcome of the measurement. Then the state of the 2-qubit system collapses to:

$$|11\rangle.$$

If we measure QL from this state, we obtain:

$$\Pr(QL \stackrel{M}{=} 0) = 0 \qquad \Pr(QL \stackrel{M}{=} 1) = 1.$$

The situation is now very different: the probability of the outcomes from a measurement on QL have changed after measuring QR. This is exactly the concept of entanglement: when two or more qubits are entangled, they affect each other, and measuring one qubit changes the probability distribution for the other qubits.

3.3 The no-cloning principle

Because measurement destroys the quantum state, it is natural to look for a way to create a copy of a quantum state, so that several measurements can be carried out without having to repeat the algorithm. However, it turns out that this cannot be done: a direct consequence of the properties of quantum gates is that a quantum state cannot be cloned.

Proposition 4. *Let $|\psi\rangle$ be an arbitrary quantum state on n qubits. There does not exist a unitary matrix that maps $|\psi\rangle \otimes |0\rangle_n$ to $|\psi\rangle \otimes |\psi\rangle$.*

Proof. Suppose there exists such a unitary U. Then for any two quantum states $|\psi\rangle, |\phi\rangle$ on n qubits, we have:

$$U(|\psi\rangle \otimes |0\rangle_n) = |\psi\rangle \otimes |\psi\rangle$$
$$U(|\phi\rangle \otimes |0\rangle_n) = |\phi\rangle \otimes |\phi\rangle.$$

Using these equalities, we can write:

$$\langle\phi|\psi\rangle = \langle\phi|\psi\rangle\langle 0|0\rangle_n == \langle\phi|\psi\rangle \otimes \langle 0|0\rangle_n = (\langle\phi| \otimes \langle 0|_n)(|\psi\rangle \otimes |0\rangle_n) = (\langle\phi| \otimes \langle 0|_n)U^*U(|\psi\rangle \otimes |0\rangle_n)$$
$$= (\langle\phi| \otimes \langle\phi|)(|\psi\rangle \otimes |\psi\rangle) = \langle\phi|\psi\rangle^2.$$

But $\langle\phi|\psi\rangle = \langle\phi|\psi\rangle^2$ is only true if $\langle\phi|\psi\rangle$ is equal to 0 or to 1, contradicting the fact that $|\phi\rangle, |\psi\rangle$ are arbitrary quantum states. □

The above proposition shows that we cannot copy a quantum state. This establishes that we cannot "cheat" the destructive effect of a measurement by simply cloning the state before the measurement. Hence, whenever we run an algorithm that produces an output quantum state, in general we can only reproduce the output quantum state only by repeating all the steps of the algorithm.

3.4 Basic operations and universality

Existing quantum computing hardware does not allow the user to specify just any unitary matrix in the code. Quantum gates have to be constructed out of a set of basic gates. We will now discuss what these basic gates are, and how they can be combined to form other operations.

The first operations that we discuss are the *Pauli gates*.

Definition 8. *The four Pauli gates are the following single-qubit gates:*

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$
$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Proposition 5. *The Pauli gates form a basis for $\mathbb{C}^{2\times 2}$, they are Hermitian, and they satisfy the relationship $XYZ = iI$.*

The X, Y, Z gates all correspond to 90° rotations, around different axes. The X gate flips a qubit:

$$X|0\rangle = |1\rangle \qquad X|1\rangle = |0\rangle.$$

This is the equivalent of a NOT gate in classical computers. At the same time, the Z gate is also called a phase-flip gate: it leaves $|0\rangle$ unchanged, and maps $|1\rangle$ to $-|1\rangle$.

$$Z|0\rangle = |0\rangle \qquad Z|1\rangle = -|1\rangle.$$

A single-qubit gate that is used in many quantum algorithms is the *Hadamard* gate:

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

The action of H is as follows:

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

Proposition 6. *Given a q-qubit quantum device initially in the state $|0\rangle_q$, applying the Hadamard gate to all qubits, equivalent to the matrix $H^{\otimes q}$, yields the uniform superposition of basis states $\frac{1}{\sqrt{2^q}} \sum_{j=0}^{2^q-1} |j\rangle_q$.*

Proof. The state $|0\rangle_q$ can also be written as $|0\rangle^{\otimes q}$. Therefore we have:

$$H^{\otimes q}|0\rangle^{\otimes q} = (H|0\rangle)^{\otimes q} = \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)^{\otimes q} = \frac{1}{\sqrt{2^q}} \sum_{j=0}^{2^q-1} |j\rangle_q.$$

□

The multiple Hadamard can be represented by one of the equivalent circuits given in Figure 9

Figure 9: Two representations for multiple Hadamard gates.

In general, since single-qubit gates are unitary matrices, they can be represented by the following parameterized matrix:

$$U(\theta, \phi, \lambda) = \begin{pmatrix} e^{-i(\phi+\lambda)/2} \cos(\theta/2) & -e^{-i(\phi-\lambda)/2} \sin(\theta/2) \\ e^{i(\phi-\lambda)/2} \sin(\theta/2) & e^{i(\phi+\lambda)/2} \cos(\theta/2) \end{pmatrix}$$

All single-qubit gates can be obtained by an appropriate choice of parameters θ, ϕ, λ.

Another fundamental gate is the CNOT gate, also called "controlled NOT". The CNOT gate is a two-qubit gate that has a control bit and a target bit, and acts as follows: if the control bit is 0, nothing happens, whereas if the control bit is 1, the target bit is flipped. The corresponding circuit is given in Figure 10
Here, the $\oplus$ operation indicates sum modulo 2. The matrix description of the gate with control qubit 2 and target qubit 1 is as follows:

$$\text{CNOT}_{21} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Figure 10: The CNOT, or controlled-NOT, gate.

We can easily see that the effect of CNOT is as follows:

$$\begin{aligned} \mathrm{CNOT}_{21}|00\rangle &= |00\rangle & \mathrm{CNOT}_{21}|01\rangle &= |01\rangle \\ \mathrm{CNOT}_{21}|10\rangle &= |11\rangle & \mathrm{CNOT}_{21}|11\rangle &= |10\rangle. \end{aligned}$$

An interesting feature of the CNOT gate is that it can be used to swap two qubits. Considering that CNOT, as all quantum gates, is a linear map, this may sound surprising. The SWAP can be constructed as depicted in Figure 11.

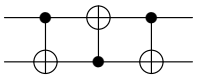


Figure 11: A circuit that swaps two qubits.

Proposition 7. *The circuit above, constructed with three CNOTs, swaps qubits 1 and 2.*

Proof. By linearity, it suffices to show that the circuit above maps $|00\rangle \to |00\rangle, |01\rangle \to |10\rangle, |10\rangle \to |01\rangle$, and $|11\rangle \to |11\rangle$. We have:

$$\begin{aligned} \mathrm{CNOT}_{21}\mathrm{CNOT}_{12}\mathrm{CNOT}_{21}|00\rangle &= \mathrm{CNOT}_{21}\mathrm{CNOT}_{12}|00\rangle = \mathrm{CNOT}_{21}|00\rangle = |00\rangle. \\ \mathrm{CNOT}_{21}\mathrm{CNOT}_{12}\mathrm{CNOT}_{21}|01\rangle &= \mathrm{CNOT}_{21}\mathrm{CNOT}_{12}|01\rangle = \mathrm{CNOT}_{21}|11\rangle = |10\rangle. \\ \mathrm{CNOT}_{21}\mathrm{CNOT}_{12}\mathrm{CNOT}_{21}|10\rangle &= \mathrm{CNOT}_{21}\mathrm{CNOT}_{12}|11\rangle = \mathrm{CNOT}_{21}|01\rangle = |01\rangle. \\ \mathrm{CNOT}_{21}\mathrm{CNOT}_{12}\mathrm{CNOT}_{21}|11\rangle &= \mathrm{CNOT}_{21}\mathrm{CNOT}_{12}|10\rangle = \mathrm{CNOT}_{21}|10\rangle = |11\rangle. \end{aligned}$$

Therefore, the SWAP circuit maps:

$$\alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \to \alpha_{00}|00\rangle + \alpha_{01}|10\rangle + \alpha_{10}|01\rangle + \alpha_{11}|11\rangle.$$

□

The types of gates that we presented here can be shown to be sufficient to construct any unitary matrix with arbitrary precision.

Definition 9. *An ordered set of gates S that yields the unitary matrix A is an ϵ-approximation of a unitary matrix U if $\sup_{\psi:\|\psi\|=1} \|U - A\| < \epsilon$. A set of gates that yields an ϵ-approximation of any unitary matrix on any given number of qubits is called a* universal *set of gates.*

Theorem 1. *(Solovay-Kitaev [Kitaev, 1997, Nielsen and Chuang, 2002]) Let $U \in \mathbb{C}^{2\times 2}$ be an arbitrary unitary matrix. Then there exists a constant c such that there exists a sequence S of gates of length $O(\log^c \frac{1}{\epsilon})$ that is an ϵ-approximation of U and consists only of H, $R_{\pi/4} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$ and CNOT gates.*

The theorem implies that just two single-qubit gates together with CNOT allow us to build any single-qubit gate with arbitrary precision. [Dawson and Nielsen, 2005] gives a proof with $c \approx 3.98$. More recent work gives improved algorithm with smaller c, in fact even $c = 1$ (but

different constants), see [Selinger, 2012, Kliuchnikov et al., 2016]. To go from single-qubit gates to general q-qubit gates, one needs at most $O(4^q)$ gates (since each gate on q qubits has $2^q \times 2^q$ elements), and the decomposition can be done in terms of single-qubit gates and CNOTs. In other words, the set of gates consisting of single-qubit gates and CNOT is universal, and in fact even a restricted set of single-qubit gates (the two gates indicated in the above theorem) is universal. This shows that with a very small set of basic gates, we can construct any unitary matrix in any dimension, although this may require many operations.

3.5 Can we solve NP-hard problems?

It is important to remark that even if we can easily create a uniform superposition of all basis states, the rules of measurement prevent us from using such a superposition to trivially solve NP-complete problems such as, for example, SAT (the satisfiability problem). Indeed, suppose we have a quantum circuit U_f that encodes a SAT formula on n boolean variables; in other words, a unitary $U_f : |j\rangle_n \otimes |0\rangle_1 \to |j\rangle_n \otimes |f(j)\rangle_1$, where $f(j)$ is 1 if the binary string jB_n satisfies the formula, and 0 if not. We might be tempted to apply $H^{\otimes n}$ to the initial state $|0\rangle_n$ to create the uniform superposition $\frac{1}{\sqrt{2^q}} \sum_{j=0}^{2^q-1} |j\rangle_q$, apply U_f to this superposition (which evaluates the truth assignment of all possible binary strings), and then perform a measurement on all qubits. But measuring the state:

$$U_f \left(\frac{1}{\sqrt{2^q}} \sum_{j=0}^{2^q-1} |j\rangle_q \otimes |0\rangle_1 \right) = \frac{1}{\sqrt{2^q}} \sum_{j=0}^{2^q-1} |j\rangle_q \otimes |f(j)\rangle_1$$

will return a binary string that satisfies the formula if and only if the last qubit has value 1 after the measurement, and this happens with a probability that depends on the number of binary assignments that satisfy the formula. If, for example, the SAT problem at hand is solved by exactly ρ assignment out of 2^n possible assignment, then the probability of finding the solution after one measurement is $\frac{\rho}{2^n}$, and we have done nothing better than randomly sampling a binary string. Clearly, this is not a good algorithm for SAT. In fact, in general solving NP-hard problems in polynomial time with quantum computers is not believed to be possible. However, we will see in the next sections some examples of quantum algorithms that are faster than any known classical algorithm.

4 A simple period finding problem: Simon's algorithm

In this section we describe a quantum algorithm that gives an expected exponential speedup with respect to classical algorithms. Admittedly, the problem that this algorithm solves is not very useful, but the ideas shown here give us a flavor of what quantum computation can do.

Let $\oplus$ denote bitwise modulo 2 addition, i.e., bitwise XOR. The problem is as follows. We are given a function $f : \{0, \ldots, 2^n - 1\} \to \{0, \ldots, 2^n - 1\}$ with the property that $f(x) = f(z)$ if and only if $xBn = zB_n \oplus aB_n$, for some unknown $a \in \{0, \ldots, 2^n - 1\}$. We do not know anything else about the function, and the goal is to find a. Notice that if $a = 0$ then the function is one-to-one, whereas if $a \neq 0$ the function is two-to-one, because for every x, there is exactly another number in domain for which the function has the same value. The function f is assumed to be given as a quantum circuit on $q = 2n$ qubits, depicted in Figure 12
This particular form of the function, that maps $|x\rangle_n \otimes |y\rangle_n$ to $|x\rangle_n \otimes |y \oplus f(x)\rangle_n$, is typical of the quantum world. Notice that if $y = 0$, then $|y \oplus f(x)\rangle_n = |f(x)\rangle_n$ so the circuit computes the

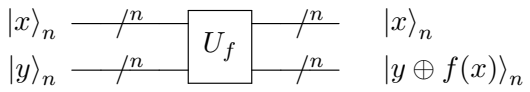


Figure 12: The circuit implementing U_f for Simon's problem.

desired function. Furthermore, this is a reversible function, because applying the same circuit U_f goes back to the initial state:

$$U_f U_f(|x\rangle_n \otimes |y\rangle_n) = U_f(|x\rangle_n \otimes |y \oplus f(x)\rangle_n) = |x\rangle_n \otimes |y \oplus f(x) \oplus f(x)\rangle_n = |x\rangle_n \otimes |y\rangle_n.$$

4.1 Classical algorithm

Because we do not know anything about the number a, the best we can do is to feed inputs to the function, and try to extract information from the output. The number a is determined once we find two distinct inputs x, z such that $f(x) = f(z)$, because then $xB_n = zB_n \oplus aB_n$ which implies $xB_n \oplus zB_n = aB_n$.

Suppose we have evaluated m distinct input values and we did not find a match. Then $aB_n \neq xB_n \oplus zB_n$ for all x, z previously evaluated, therefore we have eliminated at most $m(m-1)/2$ values of a. (Fewer values may have been eliminated if we test inputs for which $xB_n \oplus yB_n \oplus zB_n$ for any three input values x, y, z already tested. In fact, if we test w such that $wB_n = xB_n \oplus yB_n \oplus zB_n$, we have that $wB_n \oplus zB_n = xB_n \oplus yB_n$, therefore the value $wB_n \oplus zB_n$ had already been eliminated from the list of possible valus of a.) Since $m(m-1)/2$ is small compared to 2^n, the probability of success $\frac{m(m-1)}{2^{n+1}}$ is very small until we have evaluated a number of inputs that is in the order of 2^n. In particular, to guarantee a probability of success of at least ρ, we need $\frac{m(m-1)}{2^{n+1}} \geq \rho 2^n$, which implies that $m \in O(\sqrt{\rho 2^n})$. Hence, for any positive constant ρ, the number of required iterations is exponential. After evaluating $\sqrt{2^{n+1}+1} \in O(2^{n/2})$ distinct input values satisfying the condition outlined above for non-matching triplets, we are guaranteed that a matching pair has been found, or we can safely determine that $a = 0$.

4.2 Simon's algorithm: quantum computation

Using a quantum computer, we can determine a much faster. The idea, first described in [Simon, 1997], is to apply the circuit in Figure 13.

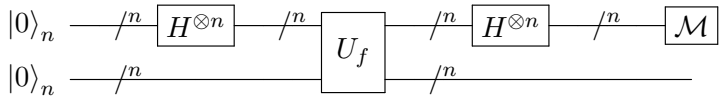


Figure 13: Quantum circuit used in Simon's algorithm.

From an algebraic point of view, this corresponds to the following operation:

$$(H^{\otimes n} \otimes I^{\otimes n}) U_f (H^{\otimes n} \otimes I^{\otimes n})(|0\rangle_n \otimes |0\rangle_n).$$

We now analyze the output of the quantum circuit, by looking at the quantum states at intermediate steps of the circuit. Let $|\psi\rangle$ be the state just before the U_f gate, $|\phi\rangle$ the state just after U_f, and $|\chi\rangle$ the final state. In other words:

$$\begin{aligned}
|\psi\rangle &= (H^{\otimes n} \otimes I^{\otimes n})(|0\rangle_n \otimes |0\rangle_n) \\
|\phi\rangle &= U_f(H^{\otimes n} \otimes I^{\otimes n})(|0\rangle_n \otimes |0\rangle_n) \\
|\chi\rangle &= (H^{\otimes n} \otimes I^{\otimes n}) U_f (H^{\otimes n} \otimes I^{\otimes n})(|0\rangle_n \otimes |0\rangle_n).
\end{aligned}$$

For $|\psi\rangle$, we know that $H^{\otimes n}$ creates a uniform superposition of $|j\rangle_n, j = 0, \ldots, 2^n - 1$ over the first n quantum bits. Therefore we can write:

$$|\psi\rangle = (H^{\otimes n} \otimes I^{\otimes n})(|0\rangle_n \otimes |0\rangle_n) = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle_n \otimes |0\rangle_n.$$

By linearity, applying U_f to this state yields:

$$|\phi\rangle = U_f|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle_n \otimes |0 \oplus f(j)\rangle_n = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle_n \otimes |f(j)\rangle_n.$$

We now need to analyze the effect of applying further Hadamard gates on the top lines of the circuit. To do so, we need an algebraic expression for the action of Hadamard gates on basis states. The effect of H on a single qubit can be summarized as follows:

$$H|z\rangle_1 = \frac{1}{\sqrt{2}}(|0\rangle_1 + (-1)^z|1\rangle_1) = \frac{1}{\sqrt{2}} \sum_{k=0}^{1} (-1)^{k \cdot z} |k\rangle_1.$$

This is consistent with our previous definition. If we apply $H^{\otimes n}$ on an n-qubit basis state, we obtain:

$$\begin{aligned} H^{\otimes n}|z\rangle_n &= \frac{1}{\sqrt{2^n}} \sum_{k_{n-1}=0}^{1} \cdots \sum_{k_0=0}^{1} (-1)^{\sum_{j=0}^{n-1} k_j \cdot (zB_n)_{j+1}} |k_{n-1}\rangle_1 \otimes \cdots \otimes |k_0\rangle_1 \\ &= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} (-1)^{kB_n \bullet zB_n} |k\rangle_n, \end{aligned}$$

where $\bullet$ is the bitwise dot product. This formula can be directly applied to our computation. The next step in the circuit is given by:

$$\begin{aligned} |\chi\rangle &= (H^{\otimes n} \otimes I^{\otimes n}) \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle_n \otimes |f(j)\rangle_n = \\ &= \frac{1}{2^n} \sum_{j=0}^{2^n-1} \sum_{k=0}^{2^n-1} (-1)^{kB_n \bullet jB_n} |k\rangle_n \otimes |f(j)\rangle_n. \end{aligned}$$

When we make a measurement on the top n qubit lines of $|\chi\rangle$, we obtain one of the states $|k\rangle_n$ with probability equal to the the sum of the square of the coefficient of the states $|k\rangle_n \otimes |f(j)\rangle$. Notice that by assumption, $|k\rangle_n \otimes |f(j)\rangle_n = |k\rangle_n \otimes |f(jB_n \oplus aB_n)\rangle_n$. Therefore, the coefficient of the basis state $|k\rangle_n \otimes |f(j)\rangle_n$ is the sum of the coefficient of the basis state $|k\rangle_n \otimes |f(j)\rangle_n$ and the basis state $|k\rangle_n \otimes |f(jB_n \oplus aB_n)\rangle_n$. Such sum of two coefficients can be expressed as:

$$\begin{aligned} \frac{(-1)^{kB_n \bullet jB_n} + (-1)^{kB_n \bullet (jB_n \oplus aB_n)}}{2^n} &= \frac{(-1)^{kB_n \bullet jB_n} + (-1)^{kB_n \bullet jB_n}(-1)^{kB_n \bullet aB_n}}{2^n} \\ &= \frac{(-1)^{kB_n \bullet jB_n} \left(1 + (-1)^{kB_n \bullet aB_n}\right)}{2^n}. \end{aligned}$$

Clearly if $kB_n \bullet aB_n$ is odd, the numerator cancels out and the sum is equal to 0. Otherwise, if $kB_n \bullet aB_n$ is even, the sum is equal to $\pm\frac{2}{2^n}$. Performing a measurement on the top n qubit lines,

we observe the binary string kB_n as output with probability $2^{n-1}\left(\pm\frac{2}{2^n}\right)^2 = \frac{1}{2^{n-1}}$, where the multiplication factor 2^{n-1} comes from the fact that there are $\frac{2^n}{2}$ distinct values for $f(j)$. Thus, the only basis states that have positive probability to be measured are those states $|k\rangle_n \otimes |f(j)\rangle_n$ for which $kB_n \bullet aB_n \equiv 0 \mod 2$. Notice that unless $k = 0$, then there is a nonempty set of bits for which the modulo 2 sum of aB_n must vanish. In this case, unless we are unlucky and we obtain the vector $kB_n = 0B_n$ or some other unlucky cases that will be specified later, we can express one of such bits as a modulo 2 sum of the others, and we eliminate half of the possible values for a.

Our discussion shows that with a single quantum computation, with high probability we learn very valuable information about a, and we can approximately halve the search space for a. It now remains to fully specify in a more precise manner how this information can be used.

4.3 Simon's algorithm: description and analysis

The quantum algorithm described in the previous section yields information on a, but it does not output a directly. To recover a, further calculations have to be performed. This is a typical situation in quantum algorithms: a quantum computation measures some properties of the desired answer; then, classical computations are used to analyze these properties and obtain the desired answer. Thus, even if the quantum algorithm does not explicitly output the desired answer, it allows us to get closer to our goal.

In the specific case of the problem discussed here, the quantum computation allows us to learn k such that $kB_n \bullet aB_n \equiv 0 \mod 2$. We embed this equation into an algorithm as follows: we initialize $E \leftarrow \emptyset$; then, while the system of equations E does not have a unique solution, we apply the circuit described in the previous section to obtain k, and add the equation $kB_n \bullet aB_n \equiv 0 \mod 2$ to E.

Because at every iteration we obtain a random k for which $kB_n \bullet aB_n \equiv 0 \mod 2$, we just have to obtain n linearly independent vectors kB_n (where independence is intended modulo 2) to ensure that the system has a unique solution. In continuous space, uniform random sampling of vectors yields linearly independent vectors with probability 1. In this case we are considering linear independence among vectors that have coefficients 0 or 1, and independence is in terms of the modulo 2 sum, so the argument is less clear; however, it is possible to show that the probability of obtaining n such linearly independent vectors after sampling $n+t$ times is bounded below by $1-\frac{1}{2^{t+1}}$ [Mermin, 2007, Apx. G]. Therefore, with overwhelming probability after slightly more than n executions of the quantum circuit, and therefore $O(n)$ queries to the function f, we determine the solution to the problem with a classical computation that can be performed in polynomial time (i.e., $O(n^2)$ to determine a solution to the system of linear equations modulo 2). Compare this with the $O(2^{n/2})$ queries that are required by a classical algorithm, and we have shown an exponential speedup.

This algorithm shows a typical feature of quantum algorithms: there is a classical computation to verify that the correct solution to the problem has been found. Indeed, quantum algorithms are probabilistic algorithm, and we can only try to increase the probability that the correct answer is returned. For this reason, we need a way to deterministically (i.e., classically) verify correctness. In other words, the quantum algorithm is applied to a problem for which it is difficult to compute the solution, but once the solution is obtained, it is easy to verify that it is correct.

5 Black-box search: Grover's algorithm

Simon's algorithm gives an exponential speedup with respect to a classical algorithm, but it solves a very narrow (and not practically useful) problem. We now describe an algorithm that gives only a polynomial (more specifically, quadratic) speedup with respect to classical, but it applies to a very large class of (practically useful) problems. The algorithm is known as Grover's search [Grover, 1996].

The problem solved by the algorithm can be described as black-box search: we are given a circuit that computes an unknown function on binary variables, and we want to determine for which value(s) of the input the function gives output 1. In other words, we are trying to determine the unique binary string that satisfies a property encoded by a circuit. The original paper [Grover, 1996] describes this as looking for a certain element in a database.

The basic idea of the algorithm is to start with the uniform superposition of all basis states, and then, iteratively, increase the amplitudes (i.e., the coefficients in the superposition of basis states) of states that correspond to binary strings for which the unknown function gives output 1.

We need some definitions. Let $f : \{0, \ldots, 2^n - 1\} \to \{0, 1\}$, and assume that there exists a unique $\ell \in \{0, \ldots, 2^n - 1\} : f(\ell) = 1$, i.e., there is a unique element in the domain of the function that yields output 1. We want to determine ℓ. (We remark that Grover's search can also be applied to the case in which there are multiple input values that yield output 1, and we want to retrieve any of them; however, the analysis in that case is slightly more convoluted, and is not pursued here.)

5.1 Classical algorithm

Given the problem definition, classical search cannot do better than $O(2^n)$ operations. Indeed, any deterministic classical algorithm may need to explore all 2^n possible input values before finding ℓ: given any deterministic classical algorithm, there exists a permutation π of $\{0, \ldots, 2^n - 1\}$ that represents the longest execution path (i.e., sequence of values at which f is queried) of such algorithm. Then, if $\ell = \pi(2^n - 1)$ the algorithm will require $O(2^n)$ queries to determine the answer, which is clearly the worst case.

At the same time, a randomized algorithm requires $O(2^n)$ function calls to have at least a constant positive probability to determine ℓ, and the expected number of function calls to determine the answer is 2^{n-1}, i.e., the expected number of flips of a biased coin with probability of heads equal to 2^{-n} until we obtain the first heads.

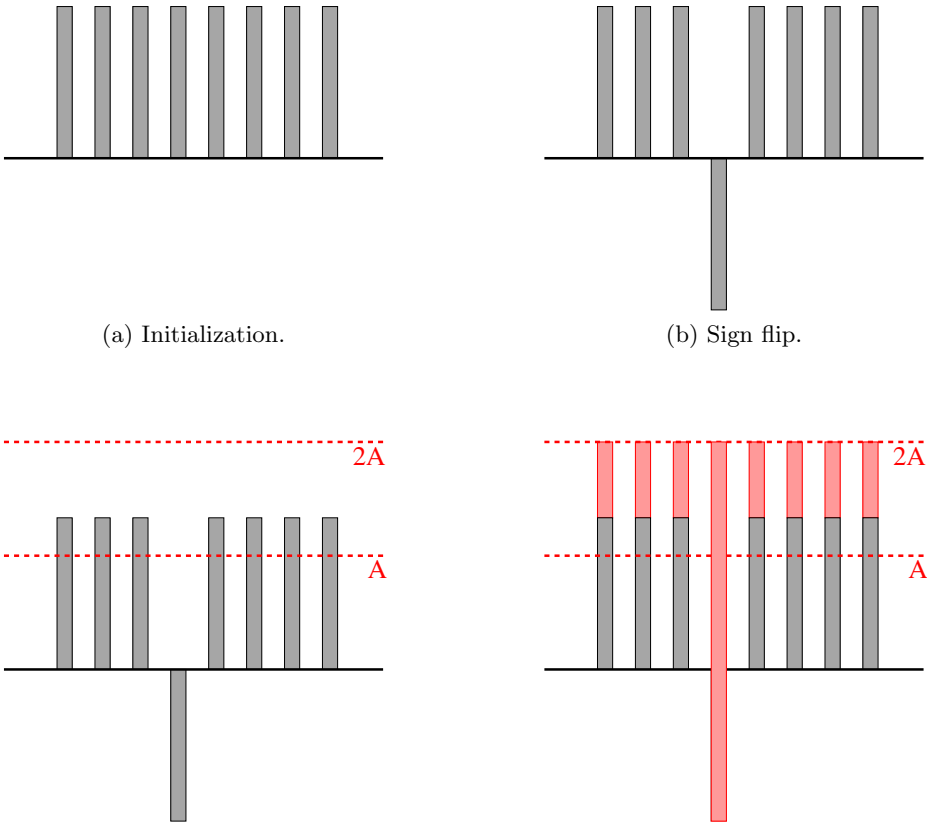
5.2 Grover's search: algorithm description

The quantum search algorithm proposed in [Grover, 1996] requires $q = n + 1$ qubits. The function f is encoded by a unitary $U_f : |j\rangle_n \otimes |y\rangle_1 \to |j\rangle_n \otimes |y \oplus f(j)\rangle_1$.

The outline of the algorithm is as follows. The algorithm starts with the uniform superposition of all basis states on n qubits. The last qubit $(n + 1)$ is used as an auxiliary qubit, and it is initialized to $H|1\rangle$. We obtain the quantum state $|\psi\rangle$. Then, these operations are repeated several times:

(i) Flip the sign of the vectors for which U_f gives output 1.

(ii) Invert all the coefficients of the quantum state around the average coefficient.



(a) Initialization.

(b) Sign flip.

(c) Computation of the average.

(d) Inversion about the average.

Figure 14: Sketch of Grover's algorithm. The bars represent the coefficients of the basis states.

A full cycle of the two operations above increases the coefficients of $|\ell\rangle_n \otimes (|0\rangle - |1\rangle)$, and after a certain number of cycles (to be specified later), the coefficient of the state $|\ell\rangle_n \otimes (|0\rangle - |1\rangle)$ is large enough that it can be obtained from a measurement with probability close to 1.

A sketch of the ideas for the algorithm is depicted in Figure 14: we have eight basis states, and suppose the fourth basis state is the target basis state $|\ell\rangle$. The representation is purely meant to convey intuition, and does not geometrically represent the vectors encoding the quantum state, but solely the amplitude of the coefficients. In Figure 14a, all basis states have the same coefficient. In Figure 14b, the coefficient of the target basis state has its sign flipped. In Figure 14c, we can see that the average value of the coefficients is slightly below the coefficient for the undesired states. Taking twice the average and subtracting each coefficient now yields the red bars in Figure 14d, where the target basis state $|\ell\rangle$ has a coefficient with much larger value than the rest, and will therefore be measured with higher probability.

We now describe the steps above more in detail.

5.2.1 Initialization

The algorithm is initialized applying $H^{\otimes(n+1)}(I^{\otimes n} \otimes X)|0\rangle_{n+1}$. We have:

$$\begin{aligned}(I^{\otimes n} \otimes X)|0\rangle_{n+1} &= |0\rangle_n \otimes |1\rangle \\ H^{\otimes(n+1)}(I^{\otimes n} \otimes X)|0\rangle_{n+1} &= \sum_{j=1}^{2^n-1} \frac{1}{\sqrt{2^n}} |j\rangle_n \otimes (|0\rangle - |1\rangle) = \sum_{j=1}^{2^n-1} \alpha_j |j\rangle_n \otimes (|0\rangle - |1\rangle) = |\psi\rangle.\end{aligned}$$

Note that the initial coefficients are real numbers. Since the steps below will map real numbers to real numbers, we only need to consider real numbers through the course of the algorithm.

5.2.2 Sign flip: step (i)

To flip the sign of the target state $|\ell\rangle_n \otimes (|0\rangle - |1\rangle)$, we apply U_f to $|\psi\rangle$. We now show why this flips the sign of $|\ell\rangle_n \otimes |0\rangle$.

$$\begin{aligned}U_f|\psi\rangle &= U_f \left(\sum_{j=1}^{2^n-1} \alpha_j |j\rangle_n \otimes (|0\rangle - |1\rangle) \right) \\ &= \alpha_\ell |\ell\rangle_n \otimes (|1\rangle - |0\rangle) + \sum_{\substack{j=1 \\ j\neq\ell}}^{2^n-1} \alpha_j |j\rangle_n \otimes (|0\rangle - |1\rangle) \\ &= \left(-\alpha_\ell |\ell\rangle_n + \sum_{\substack{j=1 \\ j\neq\ell}}^{2^n-1} \alpha_j |j\rangle_n \right) \otimes (|0\rangle - |1\rangle).\end{aligned}$$

As the expression above suggests, we can always think of the last qubit as being in the unentangled state $(|0\rangle - |1\rangle)$, with the sign flip affecting only the first n qubits. Therefore, the state that we obtain by applying U_f to $|\psi\rangle$ is the same as $|\psi\rangle$ except that the sign of the basis states $|\ell\rangle_n \otimes |0\rangle$ and $|\ell\rangle_n \otimes |1\rangle$ has been flipped.

5.2.3 Inversion about the average: step (ii)

To perform the inversion about the average, we want to perform the following operation:

$$\sum_{j=0}^{2^n-1} \alpha_j |j\rangle_n \to \sum_{j=0}^{2^n-1} \left(2 \left(\sum_{k=0}^{2^n-1} \frac{\alpha_k}{2^n} \right) - \alpha_j \right) |j\rangle_n,$$

where $\sum_{k=0}^{2^n-1} \frac{\alpha_k}{2^n}$ is the average, and therefore we are taking twice the average and subtracting each coefficient from it. This is realized by the following matrix:

$$T = \begin{pmatrix} \frac{2}{2^n} - 1 & \frac{2}{2^n} & \cdots & \frac{2}{2^n} \\ \frac{2}{2^n} & \frac{2}{2^n} - 1 & \cdots & \frac{2}{2^n} \\ \vdots & & \ddots & \vdots \\ \frac{2}{2^n} & \frac{2}{2^n} & \cdots & \frac{2}{2^n} - 1 \end{pmatrix} = \begin{pmatrix} \frac{2}{2^n} & \frac{2}{2^n} & \cdots & \frac{2}{2^n} \\ \frac{2}{2^n} & \frac{2}{2^n} & \cdots & \frac{2}{2^n} \\ \vdots & & \ddots & \vdots \\ \frac{2}{2^n} & \frac{2}{2^n} & \cdots & \frac{2}{2^n} \end{pmatrix} - I^{\otimes n},$$

where the equal elements $\frac{1}{2^n}$ over each row compute the average coefficient, the numberator 2 of the fraction takes twice the average, and finally we subtract the identity to subtract each

individual coefficient from twice the average. Define the matrix:

$$W := H^{\otimes n} \qquad W_{jk} = \frac{1}{\sqrt{2^n}}(-1)^{jB_n \bullet kB_n}.$$

Indeed, the expression above can be easily verified using the following fact:

$$H^{\otimes n} = \frac{1}{\sqrt{2}} \begin{pmatrix} H^{\otimes n-1} & H^{\otimes n-1} \\ H^{\otimes n-1} & -H^{\otimes n-1} \end{pmatrix}.$$

But now if we let:

$$R = \begin{pmatrix} 2 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} \in \mathbb{R}^{2^n \times 2^n},$$

then we can write $(WRW)_{jk} = W_{j0}R_{00}W_{0k} = \frac{2}{2^n}$, because $R_{jk} = 0$ for $j \neq 0, k \neq 0$. Therefore, using the fact that $WW = I^{\otimes n}$, we have:

$$T = WRW - I^{\otimes n} = W(R - I^{\otimes n})W = -W(I^{\otimes n} - R)W = -W\,\mathrm{diag}(\underbrace{-1, 1, \dots, 1}_{2^n})W := -WSW.$$

We must find a way to construct the matrix $S := \mathrm{diag}(-1, 1, \dots, 1)$. This will be discussed in the next section. For now, we summarize our analysis of the inversion about the average by concluding that it can be performed applying $T = -WSW = -H^{\otimes n}SH^{\otimes n}$ to the n qubits of interest (i.e., all qubits except the auxiliary qubit that we used for the sign flip of step (i)).

5.2.4 Constructing the matrix S

We give a sketch of the idea of how to construct $S = \mathrm{diag}(-1, 1, \dots, 1)$. Notice that the effect of this quantum operation is to flip the sign of the coefficient of the basis state $|0\rangle_n$, and leave each other coefficient untouched.

Instead of flipping the sign of $|0\rangle_n$, let us start by seeing how to flip the sign of $|1\rangle_n$ while leaving all other coefficients untouched. Let $\mathrm{C}^{n-1}Z$ be the gate that applies Z to qubit 1 if qubits $2, \dots, n$ are 1, and does nothing otherwise. This is a generalization of the CNOT gate, and it is called "controlled Z". $\mathrm{C}^{n-1}Z$ in the case of two qubits ($n = 2$) is given by the following matrix:

$$\mathrm{C}Z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}.$$

Notice that in the 2-qubit case, the two circuits depicted in Figure 15 are equivalent.

Figure 15: Controlled-Z gate on two qubits: two possible representations.

Carrying out the matrix multiplications will confirm that the circuit on the right in Figure 15 implements exactly the $\mathrm{C}Z$ matrix as defined above. Thus, the controlled Z gate can be easily realized with available gates.

If we have access to the $\mathrm{C}^{n-1}Z$ gate, we can write:

$$S = X^{\otimes n}(\mathrm{C}^{n-1}Z)X^{\otimes n},$$

because this operations flips the sign of the coefficient of a basis state if and only if if all qubits have value 0 in the basis state. In circuit form, it can be written as depicted in Figure 16.

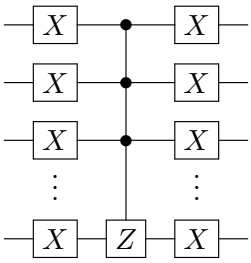


Figure 16: Quantum circuit implementing the S operation described for Grover's algorithm.

Of course, one has to construct the operation $\mathrm{C}^{n-1}Z$. There are several ways to do so. One way, suggested by [Barenco et al., 1995], is a recursive scheme that we show in Figure 17 for $n = 5$ qubits, but that clearly can be generalized to arbitary qubits.

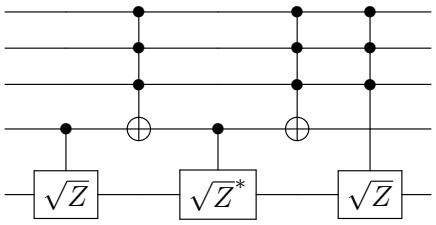


Figure 17: Decomposition of $\mathrm{C}^{n-1}Z$.

The idea here is to exploit the fact that Z has a square root, i.e., a matrix $\sqrt{Z}$ such that $\sqrt{Z}\sqrt{Z} = Z$. Notice how the scheme decomposes $\mathrm{C}^{n-1}Z$ into two copies of $\mathrm{C}^{n-2}Z$ plus a constant number of gates. This yields a decomposition with $O(n^2)$ gates, see [Barenco et al., 1995] for details. To conclude, the construction of S, and therefore of the whole circuit implementing step (ii) of Grover's search, can be performed in $O(n^2)$ gates.

5.3 Determining the number of iterations

Let Q be the matrix that applies a single iteration of Grover's search, consisting of steps (i) and (ii) above. It is paramount to determine how many iterations should be performed, so that the coefficient of the desired basis state $|\ell\rangle \otimes (|0\rangle - |1\rangle)$ is as large as possible, and the state can be measured. This is what we attempt to do in this section.

Since the last, auxiliary qubit is always in state $|0\rangle - |1\rangle$ and is unentangled, we can ignore it. Let

$$|\psi_D\rangle = |\ell\rangle_n, \qquad |\psi_U\rangle = \left(\sum_{\substack{j=1 \\ j\neq\ell}}^{2^n-1} \frac{1}{\sqrt{2^n-1}} |j\rangle_n \right)$$

be the desirable and undesirable quantum states, respectively. At iteration k of the algorithm let us denote the quantum state by $|\psi_k\rangle = d_k|\psi_D\rangle + u_k|\psi_U\rangle$. Initially, $d_0 = \frac{1}{\sqrt{2^n}}$ and $u_0 = \sqrt{\frac{2^n-1}{2^n}}$,

where notice that to obtain u_0 from the value of an individual coefficient in $|\psi_U\rangle$ we have multiplied by $\sqrt{2^n-1}$ for normalization.

At step (i), the algorithm flips $d_k|\psi_D\rangle+u_k|\psi_U\rangle \to -d_k|\psi_D\rangle+u_k|\psi_U\rangle$. There is only one state with coefficient d_k, and there are 2^n-1 states with coefficient u_k, so the value of the coefficient for these states is, respectively, d_k and $\frac{u_k}{\sqrt{2^n-1}}$. Their average at iteration k is therefore:

$$A_k = \frac{(2^n-1)\frac{1}{\sqrt{2^n-1}}u_k - d_k}{2^n} = \frac{\sqrt{2^n-1}u_k - d_k}{2^n}.$$

At step (ii), the algorithm maps $\alpha_h \to 2A_k - \alpha_h$ for each coefficient α_h. Therefore:

$$\begin{aligned} -\alpha_\ell &\to 2A_k + \alpha_\ell \\ \alpha_h &\to 2A_k - \alpha_h \quad \forall h \neq \ell, \end{aligned}$$

and to obtain u_k from α_h we need to multiply by $\sqrt{2^n-1}$, so the mapping of step (ii) can be written, overall, as:

$$-d_k|\psi_D\rangle + u_k|\psi_U\rangle \to (2A_k+d_k)|\psi_D\rangle + (2A_k\sqrt{2^n-1}-u_k)|\psi_U\rangle = d_{k+1}|\psi_D\rangle + u_{k+1}|\psi_U\rangle,$$

where we defined:

$$\begin{aligned} d_{k+1} &= 2A_k + d_k \\ u_{k+1} &= 2A_k\sqrt{2^n-1} - u_k. \end{aligned}$$

Performing the substitution of A_k, we obtain:

$$\begin{aligned} d_{k+1} &= 2\frac{\sqrt{2^n-1}u_k - d_k}{2^n} + d_k = \left(1 - \frac{1}{2^{n-1}}\right)d_k + \frac{2\sqrt{2^n-1}}{2^n}u_k \\ u_{k+1} &= 2\frac{\sqrt{2^n-1}u_k - d_k}{2^n}\sqrt{2^n-1} - u_k = -\frac{2\sqrt{2^n-1}}{2^n}d_k + \left(1 - \frac{1}{2^{n-1}}\right)u_k. \end{aligned}$$

Now this transformation is exactly a clockwise rotation by a certain angle θ, which has exactly the form $\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$, and in our case we have:

$$\sin\theta = \frac{2\sqrt{2^n-1}}{2^n}.$$

Notice that because this value of the sine is very small (for large n), then we can use the approximation $\sin x \approx x$ (when x is close to 0) to write $\theta = \frac{2\sqrt{2^n-1}}{2^n}$.

Overall, the above analysis shows that each iteration performs a rotation by an angle θ of the vectors $|\psi_D\rangle$ and $|\psi_U\rangle$. So after k iterations we obtain the following coefficients:

$$\begin{aligned} d_k &= \cos k\theta d_0 + \sin k\theta u_0 \\ u_k &= -\sin k\theta d_0 + \cos k\theta u_0. \end{aligned}$$

In order to maximize the probability of obtaining $|\psi_D\rangle$ after a measurement, remember that $|u_0| \gg |d_0|$, so the best choice is to pick $k\theta = \frac{\pi}{2}$ which yields the largest value of $|d_k|$. Hence, the optimal number of iterations of Grover's search algorithm is:

$$k \approx \frac{2^n\pi}{4\sqrt{2^n-1}} \approx \frac{\pi}{4}\sqrt{2^n}.$$

After these many iterations, we have a probability close to 1 of measuring $|\psi_D\rangle$ and obtaining the sought state $|\ell\rangle$. Comparing this with a classical algorithm, that requires $O(2^n)$ iterations, we obtained a quadratic speedup. Notice that if we perform more iterations of Grover's algorithm then the probability of measuring the desired state actually goes down, and reduces our chances of success. Therefore, it is important to choose the right number of iterations.

References

[Barenco et al., 1995] Barenco, A., Bennett, C. H., Cleve, R., DiVincenzo, D. P., Margolus, N., Shor, P., Sleator, T., Smolin, J. A., and Weinfurter, H. (1995). Elementary gates for quantum computation. *Physical review A*, 52(5):3457.

[Bennett, 1973] Bennett, C. H. (1973). Logical reversibility of computation. *IBM journal of Research and Development*, 17(6):525–532.

[Dawson and Nielsen, 2005] Dawson, C. M. and Nielsen, M. A. (2005). The Solovay-Kitaev algorithm. Technical Report quant-ph/0505030, arXiv.

[Deutsch, 1985] Deutsch, D. (1985). Quantum theory, the Church-Turing principle and the universal quantum computer. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 400, pages 97–117. The Royal Society.

[Grover, 1996] Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM Symposium on Theory of Computing*, pages 212–219. ACM.

[Kitaev, 1997] Kitaev, A. Y. (1997). Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191–1249.

[Kliuchnikov et al., 2016] Kliuchnikov, V., Maslov, D., and Mosca, M. (2016). Practical approximation of single-qubit unitaries by single-qubit quantum clifford and t circuits. *IEEE Transactions on Computers*, 65(1):161–172.

[Mermin, 2007] Mermin, N. D. (2007). *Quantum computer science: an introduction*. Cambridge University Press.

[Nielsen and Chuang, 2002] Nielsen, M. A. and Chuang, I. (2002). *Quantum computation and quantum information*. Cambridge University Press, Cambridge.

[Rieffel and Polak, 2000] Rieffel, E. and Polak, W. (2000). An introduction to quantum computing for non-physicists. *ACM Computing surveys*, 32(3):300–335.

[Rieffel and Polak, 2011] Rieffel, E. G. and Polak, W. H. (2011). *Quantum computing: A gentle introduction*. MIT Press.

[Selinger, 2012] Selinger, P. (2012). Efficient Clifford+T approximation of single-qubit operators. *arXiv preprint arXiv:1212.6253*.

[Simon, 1997] Simon, D. R. (1997). On the power of quantum computation. *SIAM journal on computing*, 26(5):1474–1483.