1 Introduction

Quantum computing is a relatively new area of computing that has the potential to greatly speed up the solution of certain problems. However, quantum computers work in a fundamentally different way than classical computers. In this course we will study the model of computation and several algorithms in areas of interest to operations research.

1.1 Model of computation

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 by applying certain operations. The model of computation that we consider is the quantum circuit model, which works as follows:

1. The quantum computer has a state that is contained in a quantum register and is initialized in a predefined way.
2. The state evolves by applying operations specified in advance in the form of an algorithm.
3. At the end of the computation, some information on the state of the quantum register is obtained by means of a special operation, called a measurement.

All terms in italics will be the subject of assumptions, upon which our exposition will build. Note that this type of computing device is similar to a Turing machine, except for the presence of a tape. 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; fundamental results regarding universal quantum computers (i.e., the quantum equivalent of a universal Turing machine) are presented in [Deutsch, 1985, Yao, 1993, Bernstein and Vazirani, 1997].

We will use the quantum circuit model throughout this course. This model of computation closely matches that of certain quantum hardware technologies that are used by some of the major players in the field [Castelvecchi, 2017], although we should note that the hardware is affected by noise and therefore it does not provide an exact implementation of the theoretical model. To understand the effect of noise, we can give the following simple, but overall quite accurate, intuitive explanation. According to the model of computation, the state evolves by applying operations, and some information on the state can be extracted via a measurement; due to noise, the state may not evolve in the desired way (e.g., applying a certain operation on the state $s_1$ should yield the state $s_2$, but we obtain a different state $s_3$ instead), or the information extracted by a measurement may not be what it is supposed to be (e.g., a measurement should produce the output $0$ with probability $p_1$, but it produces $0$ with a different probability $p_2$ instead).
Since this course aims to be “physics-free”, we will not discuss the specifics of existing quantum hardware that follows the circuit model anymore. However, we should note that a different model for quantum computing exists, and it is the so-called adiabatic model. We do not discuss the adiabatic model for two reasons: first, the adiabatic and the circuit model are equivalent [Aharonov et al., 2008], therefore we are free to choose whatever model is more convenient; second, the circuit model is more natural for computer scientists, and is the one used in most textbooks on quantum computing.

1.2 Basic definitions and notation

A course on quantum computing requires working with the decimal and the binary representation of integers, and familiarity with the properties of the tensor product. We describe here the necessary concepts and the notation.

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 $\otimes : V \times W \rightarrow V \otimes W$. The vector space $V \otimes W$ has basis $e_i \otimes f_j \forall i = 1, \ldots, m, j = 1, \ldots, n$.

If the origin vector spaces are complex Euclidean spaces of the form $\mathbb{C}^n$, and we choose the standard basis (consisting of the orthonormal vectors that have a 1 in a single position and 0 elsewhere) 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:

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

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

In this course we always work with complex Euclidean spaces 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.

Example 1. We provide an example of the tensor product for normalized vectors, which will link this concept to probability distributions and will hopefully provide a better understanding of some of the future material. Consider two independent discrete random variables $X$ and $Y$ that describe the probability of extracting numbers from two urns. The first urn contains the numbers 0 and 1, the second urn contains the numbers 00, 01, 10, 11. Assume that the extraction mechanism is biased and therefore the outcomes do not have equal probability. The outcome probabilities are given below, and for convenience we define two vectors containing them:

$$x = \begin{pmatrix}
    \Pr(X = 0) \\
    \Pr(X = 1)
\end{pmatrix} = \begin{pmatrix}
    0.25 \\
    0.75
\end{pmatrix}, \quad
y = \begin{pmatrix}
    \Pr(Y = 00) \\
    \Pr(Y = 01) \\
    \Pr(Y = 10) \\
    \Pr(Y = 11)
\end{pmatrix} = \begin{pmatrix}
    0.2 \\
    0.2 \\
    0.2 \\
    0.4
\end{pmatrix}.$$
Notice that because each vector contains probabilities for all possible respective outcomes, the vectors are normalized so that their entries sum up to 1. Then, the joint probabilities for simultaneously extracting numbers from the two urns are given by the tensor product $x \otimes y$:

\[
x \otimes y = \begin{pmatrix} 0.25 \\ 0.75 \end{pmatrix} \otimes \begin{pmatrix} 0.2 \\ 0.2 \\ 0.4 \end{pmatrix} = \begin{pmatrix} 0.05 & 0.05 & 0.15 \\ 0.05 & 0.15 & 0.15 \end{pmatrix} = \begin{pmatrix} \Pr(X = 0, Y = 0) & \Pr(X = 0, Y = 0) & \Pr(X = 0, Y = 0) \\ \Pr(X = 0, Y = 1) & \Pr(X = 0, Y = 1) & \Pr(X = 0, Y = 1) \\ \Pr(X = 0, Y = 10) & \Pr(X = 0, Y = 10) & \Pr(X = 0, Y = 10) \\ \Pr(X = 0, Y = 11) & \Pr(X = 0, Y = 11) & \Pr(X = 0, Y = 11) \\ \Pr(X = 1, Y = 00) & \Pr(X = 1, Y = 00) & \Pr(X = 1, Y = 00) \\ \Pr(X = 1, Y = 01) & \Pr(X = 1, Y = 01) & \Pr(X = 1, Y = 01) \\ \Pr(X = 1, Y = 10) & \Pr(X = 1, Y = 10) & \Pr(X = 1, Y = 10) \\ \Pr(X = 1, Y = 11) & \Pr(X = 1, Y = 11) & \Pr(X = 1, Y = 11) \end{pmatrix},
\]

where the last equality is due to the fact that $X$ and $Y$ are independent. The vector $x \otimes y$ is also normalized, which is easy to verify algebraically.

The next proposition states some properties of the tensor product that will be useful in the rest of the course.

**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, $u, v \in \mathbb{C}^m, w, x \in \mathbb{C}^n$, 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) $(au) \otimes (bw) = ab(u \otimes w)$.

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

Above and in the following, the notation $A^\dagger$ denotes the conjugate transpose of $A$, which is the matrix defined as follows: $A^\dagger := A^\top$. Given a matrix $A$, the notation $A^{\otimes n}$ indicates the tensor product of $A$ with itself $n$ times, and the same notation will be used for vector spaces $S$:

$A^{\otimes n} := A \otimes A \cdots \otimes A, \quad S^{\otimes n} := S \otimes S \cdots \otimes S$.

The quantum computing literature refers to a Hilbert space, typically denoted $\mathcal{H}$, rather than a complex Euclidean space $\mathbb{C}^n$. However, the material discussed in this course does not require any property of Hilbert spaces that is not already present in complex Euclidean spaces, hence we stick to the more familiar concept.

We will work extensively with binary strings, using the following definitions.

**Definition 3.** For any integer $q > 0$, we denote by $\vec{j} \in \{0, 1\}^q$ a binary string on $q$ digits, where we use the arrow to emphasize that $\vec{j}$ is a string of binary digits rather than an integer. Given $\vec{j} \in \{0, 1\}^q$, we denote its $k$-th digit by $j_k$.
We use the notation $\vec{0}$ to denote the all-zero binary string, and $\vec{1}$ to denote the all-one binary string: the size of these strings will always be clear from the context. We use a little-endian convention for binary strings, i.e., the first digit is the most significant one. Given a binary string $\vec{j}$, we use the corresponding symbol without the arrow, $j$, to denote the number it corresponds to, i.e., $j = \sum_{k=1}^{q} j_k 2^{q-k}$.

In the rest of this course, as is frequent in the quantum computing literature, we use $\mathbb{C}^n$ to represent $n$-dimensional complex vector space. The standard basis for $\mathbb{C}^2$ is denoted by $|0\rangle$, $|1\rangle$, $\langle 0|$, and $\langle 1|$.

In this work will be of the form $(\mathbb{C}^2)^{\otimes q}$, where $q$ is a given integer. It is therefore convenient to specify the basis elements of such spaces.

**Definition 4.** Given a complex Euclidean space $\mathbb{S} \equiv \mathbb{C}^n$, $|\psi\rangle \in \mathbb{S}$ denotes a column vector, and $\langle \psi | \in \mathbb{S}^\dagger$ denotes a row vector that is the conjugate transpose of $|\psi\rangle$, i.e., $\langle \psi | = |\psi\rangle^\dagger$. The vector $|\psi\rangle$ is also called a ket, and the vector $\langle \psi |$ is also called a bra.

Thus, an expression such as $\langle \psi | \phi \rangle$ is an inner product. The complex Euclidean spaces used in this work will be of the form $(\mathbb{C}^2)^{\otimes q}$, where $q$ is a given integer. It is therefore convenient to specify the basis elements of such spaces.

**Definition 5.** The standard basis for $\mathbb{C}^2$ is denoted by $|0\rangle_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $|1\rangle_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. The standard basis for $(\mathbb{C}^2)^{\otimes q}$, which has $2^q$ elements, is denoted by $|\vec{j}\rangle$, $\vec{j} \in \{0, 1\}^q$.

According to our notation, for any $q$-digit binary string $\vec{j} \in \{0, 1\}^q$, $|\vec{j}\rangle$ is the $2^q$-dimensional basis vector in $(\mathbb{C}^2)^{\otimes q}$ corresponding to the binary string $\vec{j}$. Since we always use the standard basis and the most natural order for its vectors, it is easy to verify that for $\vec{j} \in \{0, 1\}^q$, $|\vec{j}\rangle$ is the basis vector with a 1 in position $\sum_{k=1}^{q} j_k 2^{q-k} + 1$, and 0 elsewhere. For example, $|101\rangle$ is the 8-dimensional basis vector $(0 0 0 0 1 0 0 \ 0)^\top$, obtained as the tensor product $|1\rangle \otimes |0\rangle \otimes |1\rangle$.

Whenever useful for clarity, we use a subscript for bras and kets to denote the dimension of the space that the vector belongs to, e.g., we write $|\vec{j}\rangle_q$ to emphasize that we are working in a $2^q$ dimensional space (or, in other words, that the basis elements of the space are associated with binary strings with $q$ digits). We typically omit the subscript if the dimension of the space is evident from the context. We provide a further example of this notation below.

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

|00\rangle_2 = |00\rangle = |0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\
|01\rangle_2 = |01\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \\
|10\rangle_2 = |10\rangle = |1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \\
|11\rangle_2 = |11\rangle = |1\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.


In the above example we made an exception to our rule and used a subscript to denote the
dimension of the basis vectors, just to emphasize that $|00\rangle_2$ and $|00\rangle$ are exactly the same. In
the remainder of this paper, we will always write $|01\rangle$ rather than $|01\rangle_2$ because it is clear that
the basis element $|01\rangle$ has two digits and therefore lives in the space $(\mathbb{C}^2)^\otimes 2$.

To improve clarity when dealing with vectors in $(\mathbb{C}^2)^\otimes q$, we always denote basis vectors using
spelled-out binary strings or Roman letters, (e.g., $|01\rangle, |\vec{h}\rangle, |\vec{x}\rangle, |\vec{y}\rangle$ all denote basis vectors),
whereas we use Greek letters to denote vectors that may not be basis vectors (e.g., $|\psi\rangle, |\phi\rangle$ all
denote vectors that may not be basis vectors). In the same spirit, single-digit binary numbers
are always denoted with Roman letters (e.g., $x, y, z$ denote a 0 or a 1).

2 Qubits and quantum states

According to our computational model, a quantum computing device has a state that is stored
in the quantum register. Qubits 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 a single quantum register that is made up of qubits. The assumption that there
is a single quantum register is without loss of generality, as one can think of multiple registers as
being placed “side-by-side” to form a single register (of course, one would then need to specify
what operations are allowed on the resulting register). The state of the quantum register, and
therefore of the quantum computing device, is defined next.

Assumption 1. The state of a $q$-qubit quantum register is a unit vector in $(\mathbb{C}^2)^\otimes q = \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2, q$ times.

Remark 1. A vector $|\psi\rangle \in \mathbb{C}^n$ is a unit vector if $|||\psi||| = \sqrt{\langle \psi | \psi \rangle} = 1$.

Remark 2. Choosing the standard basis for $\mathbb{C}^2$, the state of a 1-qubit register ($q = 1$) can be
represented as $\alpha |0\rangle + \beta |1\rangle = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ where $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$.

Remark 3. Given the standard basis for $\mathbb{C}^2$, a basis for $(\mathbb{C}^2)^\otimes q$ is given by the following $2^q$
vectors:

$$
|00\cdots00\rangle = |0\rangle \otimes \cdots \otimes |0\rangle \otimes |0\rangle,
|00\cdots01\rangle = |0\rangle \otimes \cdots \otimes |0\rangle \otimes |1\rangle,
\vdots
|11\cdots11\rangle = |1\rangle \otimes \cdots \otimes |1\rangle \otimes |1\rangle.
$$

In more compact form, the vectors are denoted by $|\vec{j}\rangle, \vec{j} \in \{0,1\}^q$. The state of a $q$-qubit quantum
register can then be represented as: $|\psi\rangle = \sum_{\vec{j} \in \{0,1\}^q} \alpha_{\vec{j}} |\vec{j}\rangle_q$, with $\alpha_{\vec{j}} \in \mathbb{C}$ and $\sum_{\vec{j} \in \{0,1\}^q} |\alpha_{\vec{j}}|^2 = 1$.

For brevity, we often write “state of $q$-qubits” or “$q$-qubit state” to refer to the state of a
$q$-qubit quantum register. This is common in the literature, where the discussion of qubits is
not necessarily limited to the context of quantum registers. By properties of the tensor product,
we will see that sometimes it is appropriate to refer to the state of just some of the qubits of a...
quantum computing device, rather than all of them, and this may still be a well-defined concept. We will revisit this in Section 2.2.

It is important to remark that $(\mathbb{C}^2)^\otimes q$ 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. Furthermore, to represent a quantum state we need complex coefficients: the state of a $q$-qubit quantum register is described by $2^q$ complex coefficients, which is an enormous amount of information compared to what is necessary to describe a $q$-bit classical register. However, later we will see that a quantum state cannot be accessed directly, therefore even if a description of the quantum state requires infinite precision in principle, we cannot access such description as easily as with classical registers. In fact, as it turns out we cannot extract more than $q$ bits of information out of a $q$-qubit register! This will be intuitively clear after stating the effect of quantum measurements in the next lecture; for a formal proof, see [Kholevo, 1973].

2.1 Basis states and superposition

We continue our study of the state of quantum registers by discussing the concept of superposition.

**Definition 6.** We say that $q$ qubits are in a basis state if the state $|\psi\rangle = \sum_{j \in \{0, 1\}^q} \alpha_j |j\rangle_q$ of the corresponding register is such that $\exists \bar{k} : |\alpha_{\bar{k}}| = 1$, $\alpha_j = 0 \ \forall j \neq \bar{k}$. Otherwise, we say that they are in a superposition.

**Remark 4.** A simpler, more intuitive definition would be to say that a basis state is such that $|\psi\rangle = |\bar{k}\rangle$ for some $\bar{k} \in \{0, 1\}^q$. It is acceptable to use the simpler definition if desired: as it turns out, even if the states $|\alpha_{\bar{k}}| |\bar{k}\rangle$ for some $\bar{k} \in \{0, 1\}^q$ and $|\alpha_{\bar{k}}|^2 = 1$ are all different in principle, they are equivalent to $|\bar{k}\rangle$ up to the multiplication factor $\alpha_{\bar{k}}$ which will be seen to be unimportant in the next lecture.

**Example 3.** Consider two 1-qubit registers and their states $|\psi\rangle, |\phi\rangle$:

\[
|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle
\]

\[
|\phi\rangle = \beta_0 |0\rangle + \beta_1 |1\rangle.
\]

If we put these 1-qubit registers side-by-side to form a 2-qubit register, then the 2-qubit register will be in state:

\[
|\psi\rangle \otimes |\phi\rangle = \alpha_0 \beta_0 |00\rangle \otimes |0\rangle + \alpha_0 \beta_1 |00\rangle \otimes |1\rangle + \alpha_1 \beta_0 |10\rangle \otimes |0\rangle + \alpha_1 \beta_1 |11\rangle \otimes |1\rangle.
\]

If both $|\psi\rangle$ and $|\phi\rangle$ are in a basis state, we have that either $\alpha_0$ or $\alpha_1$ is zero, and similarly either $\beta_0$ or $\beta_1$ is zero, while the nonzero coefficients have modulus one. Thus, only one of the coefficients in the expression of the state of $|\psi\rangle \otimes |\phi\rangle$ is nonzero, and in fact its modulus is one. This implies that if both $|\psi\rangle$ and $|\phi\rangle$ are in a basis state, $|\psi\rangle \otimes |\phi\rangle$ 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 $|\psi\rangle$ and $|\phi\rangle$ are in a superposition. Then the state of $|\psi\rangle \otimes |\phi\rangle$ 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 unit vectors is also a unit vector.

6
The example clearly generalizes to an arbitrary number of qubits. In fact the following proposition is trivially true:

**Proposition 2.** For any \( q \), a \( q \)-qubit register is in a basis state if and only if its state can be expressed as the tensor product of \( q \) 1-qubit registers, each of which 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., a \( q \)-bit classical register will always contain exactly one of the \( 2^q \) binary strings in \( \{0,1\}^q \). 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.

### 2.2 Product states and entanglement

We have seen that the state of a \( q \)-qubit register is a vector in \((\mathbb{C}^2)^\otimes q\), which is a \( 2^q \) dimensional space. Since this is a tensor product of \( \mathbb{C}^2 \), i.e., the space in which 1-qubit states 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 \( q \) 1-qubit states. We can answer this question by using the definitions given above. The state of \( q \) qubits is a unit vector in \((\mathbb{C}^2)^\otimes q\), and it can be written as:

\[
|\psi\rangle = \sum_{j \in \{0,1\}^q} \alpha_j |\vec{j}\rangle_q, \quad \sum_{j \in \{0,1\}^q} |\alpha_j|^2 = 1.
\]

Now let us consider the tensor product of \( q \) 1-qubit states, the \( k \)-th of which is given by \( \beta_{k,0}|0\rangle + \beta_{k,1}|1\rangle \), for \( k = 1, \ldots, q \) (the first qubit corresponds to the most significant bit, according to the little-endian convention). Taking the tensor product we obtain the vector:

\[
|\phi\rangle = (\beta_{1,0}|0\rangle + \beta_{1,1}|1\rangle) \otimes (\beta_{2,0}|0\rangle + \beta_{2,1}|1\rangle) \otimes \cdots \otimes (\beta_{q,0}|0\rangle + \beta_{q,1}|1\rangle)
\]

\[
= \sum_{j_1=0}^1 \sum_{j_2=0}^1 \cdots \sum_{j_q=0}^1 \prod_{k=1}^q \beta_{k,j_k} |j_1 j_2 \cdots j_q\rangle = \sum_{j \in \{0,1\}^q} \prod_{k=1}^q \beta_{k,j_k} |\vec{j}\rangle_q,
\]

satisfying \( |\beta_{k,0}|^2 + |\beta_{k,1}|^2 = 1 \) \( \forall k = 1, \ldots, q \).

The normalization condition for \( |\phi\rangle \) implies the normalization condition of \( |\psi\rangle \), but the converse is not true. That is, \( |\beta_{k,0}|^2 + |\beta_{k,1}|^2 = 1 \) \( \forall k = 1, \ldots, q \) implies \( \sum_{j_1=0}^1 \sum_{j_2=0}^1 \cdots \sum_{j_q=0}^1 \prod_{k=1}^q |\beta_{k,j_k}|^2 = 1 \), but not vice versa. This means that there exist values of \( \alpha_j \), with \( \sum_{j \in \{0,1\}^q} |\alpha_j|^2 = 1 \), that cannot be expressed as coefficients \( \beta_{k,0}, \beta_{k,1} \) (for \( k = 1, \ldots, q \)) satisfying the conditions for \( |\phi\rangle \).

This is easily clarified with an example.

**Example 4.** Consider two 1-qubit states:

\[
|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle
\]

\[
|\phi\rangle = \beta_0|0\rangle + \beta_1|1\rangle.
\]

Taking the two qubits together in a 2-qubit register, the state of the 2-qubit register is:

\[
|\psi\rangle \otimes |\phi\rangle = \alpha_0 \beta_0|00\rangle + \alpha_0 \beta_1|01\rangle + \alpha_1 \beta_0|10\rangle + \alpha_1 \beta_1|11\rangle,
\]
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 $|\xi\rangle$ is:

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

with normalization condition $|\gamma_{00}|^2 + |\gamma_{01}|^2 + |\gamma_{10}|^2 + |\gamma_{11}|^2 = 1$. Comparing equations (1) and (2), we determine that $|\xi\rangle$ is of the form $|\psi\rangle \otimes |\phi\rangle$ if and only if it satisfies the relationship:

$$\gamma_{00}\gamma_{11} = \gamma_{01}\gamma_{10}.$$  \hspace{1cm} (3)

Clearly $|\psi\rangle \otimes |\phi\rangle$ 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 (3) 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:

$$|\gamma_{00}| = \sqrt{|\gamma_{00}|^2} = \sqrt{|\gamma_{00}|^2(|\gamma_{00}|^2 + |\gamma_{01}|^2 + |\gamma_{10}|^2 + |\gamma_{11}|^2)}$$

$$= \sqrt{|\gamma_{00}|^4 + |\gamma_{00}|^2|\gamma_{01}|^2 + |\gamma_{10}|^2|\gamma_{00}|^2 + |\gamma_{01}|^2|\gamma_{10}|^2 + |\gamma_{11}|^2|\gamma_{10}|^2}$$

$$= \left(\sqrt{|\gamma_{00}|^2 + |\gamma_{01}|^2}\sqrt{|\gamma_{00}|^2 + |\gamma_{11}|^2}\right).$$

and similarly for the other coefficients:

$$|\gamma_{01}| = \left(\sqrt{|\gamma_{00}|^2 + |\gamma_{01}|^2}\sqrt{|\gamma_{01}|^2 + |\gamma_{11}|^2}\right),$$

$$|\gamma_{10}| = \left(\sqrt{|\gamma_{10}|^2 + |\gamma_{11}|^2}\sqrt{|\gamma_{00}|^2 + |\gamma_{10}|^2}\right),$$

$$|\gamma_{11}| = \left(\sqrt{|\gamma_{10}|^2 + |\gamma_{11}|^2}\sqrt{|\gamma_{01}|^2 + |\gamma_{11}|^2}\right).$$

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

$$\alpha_0 = e^{i\theta_{00}}|\alpha_0|, \quad \alpha_1 = e^{i\theta_{10}}|\alpha_1|, \quad \beta_0 = |\beta_0|, \quad \beta_1 = e^{i(\theta_{01} - \theta_{00})}|\beta_1|. \hspace{1cm} (4)$$

It is now easy to verify that the state $|\xi\rangle$ in (2) can be expressed as $|\psi\rangle \otimes |\phi\rangle$ in (1) with coefficients $\alpha_0, \alpha_1, \beta_0, \beta_1$ as given in (4).

The condition in equation (3), to verify if the coefficients of a 2-qubit state $|\xi\rangle$ can be expressed as a tensor product of two 1-qubit states, can also be written in matrix form, which makes it easier to remember. If we assign the rows of the matrix to the first qubit, and the columns to the second qubit, we can arrange the coefficients $\gamma$ as follows (notice how the first qubit has value 0 in the first row and 1 in the second row; similarly for the second qubit and the columns):

$$\begin{pmatrix}
\gamma_{00} & \gamma_{01} \\
\gamma_{10} & \gamma_{11}
\end{pmatrix}.$$

Then, $|\xi\rangle$ is a tensor product of two 1-qubit states if and only if this matrix has rank 1. This is equivalent to (3).
We formalize the concept of expressing a quantum state as a tensor product of lower-dimensional quantum states as follows.

**Definition 7.** A quantum state \( |\psi\rangle \in (\mathbb{C}^2)^\otimes q \) is a product state if it can be expressed as a tensor product \( |\psi_1\rangle \otimes \cdots \otimes |\psi_q\rangle \) of \( q \) 1-qubit 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, \( |\psi\rangle \) exhibits some entanglement, but in some sense it can still be “simplified”. Generally, according to the definition above, we call a quantum state entangled as long as it cannot be fully decomposed into a tensor product of 1-qubit states. In the case of quantum systems composed of multiple subsystems (rather than just two subsystems as in the example \( |\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \)), the concept of entanglement as discussed in the literature is not as simple as given in Def. 7 (and the rank-1 test discussed at the end of Example 4 is not well-defined). However, our simplified definition works in this course and for most of the literature on quantum algorithms, therefore we can leave other considerations aside; we refer to [Coffman et al., 2000] as an entry point for a discussion on multipartite entanglement.

**Example 5.** 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 \( \left( \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \right) \otimes \left( \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \right) \). By contrast, 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 1-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. The required property is stated below, and we treat it as an assumption.

**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 5.** A matrix \( U \) is unitary if \( U^\dagger U = UU^\dagger = I \).

A well-known property of unitary matrices is that they are norm-preserving; that is, given a unitary matrix \( U \) and a vector \( v \), \( \|Uv\| = \|v\| \). Thus, for a \( q \)-qubit system, the quantum state is a unit vector \( |\psi\rangle \in \mathbb{C}^{2^q} \), a quantum operation is a matrix \( U \in \mathbb{C}^{2^q \times 2^q} \), and the application of \( U \) onto the state \( |\psi\rangle \) is the unit vector \( U|\psi\rangle \in \mathbb{C}^{2^q} \). 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 does not appear to be reversible, because memory can be erased and, in the classical Turing machine, symbols can be erased from the tape. However, [Bennett, 1973] shows that all computations (including classical computations) can be made reversible by means of extra space. The general idea to make a function invertible is to have separate input and output registers: any output is stored in a different location than the input, so that the input does not have to be erased. This is a standard trick in quantum computing that will be discussed in future lectures, but in order to do that, we first need to introduce some notation for 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 increasing order from the top. 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.](image)

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 \( A\ B\ |\psi\rangle \).

![Figure 2: Order of the operations in a quantum circuit.](image)

Gates can also be applied to individual qubits. Because a single qubit is a vector in \( \mathbb{C}^2 \), a single-qubit gate is a unitary matrix in \( \mathbb{C}^{2\times2} \). Consider the same three-qubit device, and suppose we want to apply the gate only to the third qubit. We would write it as in Figure 3.

![Figure 3: A circuit with a single-qubit gate.](image)

From an algebraic point of view, the action of our first example in Figure 1 on the quantum state is clear: the state of the three qubits is mapped onto another three-qubit state, as \( U \) acts on all the qubits. To understand the example in Figure 3, where \( U \) is a single-qubit gate that
acts on qubit 3 only, we must imagine that an identity gate is applied to all the empty qubit lines. Therefore, Figure 3 can be thought of as indicated in Figure 4.

```
qubit 1 —[I]—
qubit 2 —[I]—
qubit 3 —[U]—
```

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

This circuit can be interpreted as applying the gate $I \otimes I \otimes U$ to the 3-qubit state. Notice that by convention the matrix $U$, which is applied to qubit 3, appears in the rightmost term of the tensor product. This is because qubit 3 is associated with the least significant digit according to our little-endian convention, see Def. 3 and the subsequent discussion. If we have a product state $|\psi\rangle \otimes |\phi\rangle \otimes |\xi\rangle$, we can write labels as indicated in Figure 5.

```
|\psi\rangle ——— |\psi\rangle
|\phi\rangle ——— |\phi\rangle
|\xi\rangle ——— U ——— U|\xi\rangle
```

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

Indeed, $(I \otimes I \otimes U)(|\psi\rangle \otimes |\phi\rangle \otimes |\xi\rangle) = |\psi\rangle \otimes |\phi\rangle \otimes U|\xi\rangle$. 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 input state, the effect of the circuit is as indicated in Figure 6. Notice that this fact is essentially the reason why

```
|\psi\rangle ——— (I \otimes I \otimes U)|\psi\rangle
```

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

simulation of quantum computations on classical computers may take exponential resources in the worst case: to simulate the effect of even a single-qubit gate on the entangled state $|\psi\rangle$, we have to explicitly compute the effect of the $2^q \times 2^q$ matrix $(I \otimes I \otimes U)$ on the state $|\psi\rangle$. This requires exponential space with a naive approach (if the matrices and vectors are stored explicitly), and even with more parsimonious approaches it may require exponential time (e.g., if we compute elements of the state vector one at a time). 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 – or time – required.

**Remarks on our notation**

In this course we use several notational devices for clarity that are usually not employed in the quantum computing literature. We list them here.

- The subscript for bra-ket vectors to indicate the dimension of the space, e.g., $|\psi\rangle_q$ for $2^q$-dimensional vectors. Typically, the dimension of the space is defined elsewhere and can
be understood from the context. Whenever subscripts for kets are used, it is normally to address registers.

- The vector arrow, e.g., $\vec{j}$, to indicate binary strings. Typically, binary strings are not distinguished from other mathematical symbols and they can be identified from the context.
- The use of Roman letters for basis vectors and Greek letters for general, i.e., possibly not basis, vectors.

Finally, the all-zero binary string of dimension $q$ is normally denoted $0^q$, rather than $\vec{0}$.

References


