CHAPTER

Preliminaries 2

This chapter introduces the basic concepts and notations from quantum mechanics and quantum computation used throughout the book.

- Of course, quantum programming theory is built based on quantum mechanics. So, Section 2.1 introduces the Hilbert space formalism of quantum mechanics, which is exactly the mathematical knowledge base of this book.
- Quantum circuits are introduced in Section 2.2. Historically, several major quantum algorithms appeared before any quantum programming language was defined. So, quantum circuits usually serve as the computational model in which quantum algorithms are described.
- Section 2.3 introduces several basic quantum algorithms. The aim of this section is to provide examples for quantum programming rather than a systematic exposition of quantum algorithms. Thus, I decided not to include more sophisticated quantum algorithms.

In order to allow the reader to enter the core of this book – quantum programming – as quickly as possible, I tried to make this chapter minimal. Thus, the materials in this chapter are presented very briefly. Total newcomers to quantum computation can start with this chapter, but at the same time I suggest that they read the corresponding parts of Chapters 2, 4, 5, 6 and 8 of book [174] for more detailed explanations and examples of the notions introduced in this chapter. On the other hand, for the reader who is familiar with these materials from a standard textbook such as [174], I suggest moving directly to the next chapter, using this chapter only for fixing notations.

2.1 QUANTUM MECHANICS

Quantum mechanics is a fundamental physics subject that studies phenomena at the atomic and subatomic scales. A general formalism of quantum mechanics can be elucidated based on several basic postulates. We choose to introduce the basic postulates of quantum mechanics by presenting the mathematical framework in which these postulates can be properly formulated. The physics interpretations of

Foundations of Quantum Programming.http://dx.doi.org/10.1016/B978-0-12-802306-8.00002-1
Copyright © 2016 Elsevier Inc. All rights reserved.

Copyright © 2016. Elsevier Science & Technology. All rights reserved.
Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

these postulates are only very briefly discussed. I hope this provides the reader a short cut towards a grasp of quantum programming.

2.1.1 HILBERT SPACES

A Hilbert space usually serves as the state space of a quantum system. It is defined based on the notion of vector space. We write $\mathbb{C}$ for the set of complex numbers. For each complex number $\lambda = a + bi \in \mathbb{C}$, its conjugate is $\lambda^* = a - bi$. We adopt the Dirac notation which is standard in quantum mechanics: $|\varphi\rangle, |\psi\rangle, \ldots$ stands for vectors.

Definition 2.1.1. *A (complex) vector space is a nonempty set $\mathcal{H}$ together with two operations:*

- *vector addition* $+ : \mathcal{H} \times \mathcal{H} \to \mathcal{H}$
- *scalar multiplication* $\cdot : \mathbb{C} \times \mathcal{H} \to \mathcal{H}$

satisfying the following conditions:

(i) $+$ *is commutative:* $|\varphi\rangle + |\psi\rangle = |\psi\rangle + |\varphi\rangle$ *for any* $|\varphi\rangle, |\psi\rangle \in \mathcal{H}$.
(ii) $+$ *is associative:* $|\varphi\rangle + (|\psi\rangle + |\chi\rangle) = (|\varphi\rangle + |\psi\rangle) + |\chi\rangle$ *for any* $|\varphi\rangle, |\psi\rangle, |\chi\rangle \in \mathcal{H}$.
(iii) $+$ *has the zero element* 0*, called the zero vector, such that* $0 + |\varphi\rangle = |\varphi\rangle$ *for any* $|\varphi\rangle \in \mathcal{H}$.
(iv) *each* $|\varphi\rangle \in \mathcal{H}$ *has its negative vector* $-|\varphi\rangle$ *such that* $|\varphi\rangle + (-|\varphi\rangle) = 0$.
(v) $1|\varphi\rangle = |\varphi\rangle$ *for any* $|\varphi\rangle \in \mathcal{H}$.
(vi) $\lambda(\mu|\varphi\rangle) = \lambda\mu|\varphi\rangle$ *for any* $|\varphi\rangle \in \mathcal{H}$ *and* $\lambda, \mu \in \mathbb{C}$.
(vii) $(\lambda + \mu)|\varphi\rangle = \lambda|\varphi\rangle + \mu|\varphi\rangle$ *for any* $|\varphi\rangle \in \mathcal{H}$ *and* $\lambda, \mu \in \mathbb{C}$.
(viii) $\lambda(|\varphi\rangle + |\psi\rangle) = \lambda|\varphi\rangle + \lambda|\psi\rangle$ *for any* $|\varphi\rangle, |\psi\rangle \in \mathcal{H}$ *and* $\lambda \in \mathbb{C}$.

To define the notion of Hilbert space, we also need the following:

Definition 2.1.2. *An inner product space is a vector space $\mathcal{H}$ equipped with an inner product; that is, a mapping:*

$$\langle \cdot | \cdot \rangle : \mathcal{H} \times \mathcal{H} \to \mathbb{C}$$

satisfying the following properties:

(i) $\langle \varphi | \varphi \rangle \geq 0$ *with equality if and only if* $|\varphi\rangle = 0$*;*
(ii) $\langle \varphi | \psi \rangle = \langle \psi | \varphi \rangle^*$*;*
(iii) $\langle \varphi | \lambda_1 \psi_1 + \lambda_2 \psi_2 \rangle = \lambda_1 \langle \varphi | \psi_1 \rangle + \lambda_2 \langle \varphi | \psi_2 \rangle$

for any $|\varphi\rangle, |\psi\rangle, |\psi_1\rangle, |\psi_2\rangle \in \mathcal{H}$ *and for any* $\lambda_1, \lambda_2 \in \mathbb{C}$.

For any vectors $|\varphi\rangle, |\psi\rangle \in \mathcal{H}$, the complex number $\langle \varphi | \psi \rangle$ is called the inner product of $|\varphi\rangle$ and $|\psi\rangle$. Sometimes, we write $(|\varphi\rangle, |\psi\rangle)$ for $\langle \varphi | \psi \rangle$. If $\langle \varphi | \psi \rangle = 0$, then we say that $|\varphi\rangle$ and $|\psi\rangle$ are orthogonal and write $|\varphi\rangle \perp |\psi\rangle$. The length of a vector $|\psi\rangle \in \mathcal{H}$ is defined to be

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

A vector $|\psi\rangle$ is called a unit vector if $||\psi|| = 1$.

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

The notion of limit can be defined in terms of the length of a vector.

Definition 2.1.3. *Let $\{|\psi_n\rangle\}$ be a sequence of vectors in $\mathcal{H}$ and $|\psi\rangle \in \mathcal{H}$.*

(i) *If for any $\epsilon > 0$, there exists a positive integer N such that $||\psi_m - \psi_n|| < \epsilon$ for all $m, n \geq N$, then $\{|\psi_n\rangle\}$ is called a Cauchy sequence.*

(ii) *If for any $\epsilon > 0$, there exists a positive integer N such that $||\psi_n - \psi|| < \epsilon$ for all $n \geq N$, then $|\psi\rangle$ is called a limit of $\{|\psi_n\rangle\}$ and we write $|\psi\rangle = \lim_{n\to\infty} |\psi_n\rangle$.*

Now we are ready to present the definition of Hilbert space.

Definition 2.1.4. *A Hilbert space is a complete inner product space: that is, an inner product space in which each Cauchy sequence of vectors has a limit.*

A notion that helps us to understand the structure of a Hilbert space is its basis. In this book, we only consider finite-dimensional or countably infinite-dimensional (separable) Hilbert space.

Definition 2.1.5. *A finite or countably infinite family $\{|\psi_i\rangle\}$ of unit vectors is called an orthonormal basis of $\mathcal{H}$ if*

(i) *$\{|\psi_i\rangle\}$ are pairwise orthogonal: $|\psi_i\rangle \perp |\psi_j\rangle$ for any i, j with $i \neq j$;*

(ii) *$\{|\psi_i\rangle\}$ span the whole space $\mathcal{H}$: each $|\psi\rangle \in \mathcal{H}$ can be written as a linear combination $|\psi\rangle = \sum_i \lambda_i |\psi_i\rangle$ for some $\lambda_i \in \mathbb{C}$ and a finite number of $|\psi_i\rangle$.*

The numbers of vectors in any two orthonormal bases are the same. This is called the dimension of $\mathcal{H}$ and written as $\dim \mathcal{H}$; in particular, if an orthonormal basis contains infinitely many vectors, then $\mathcal{H}$ is infinite-dimensional and we write $\dim \mathcal{H} = \infty$.

Infinite-dimensional Hilbert spaces are required in quantum programming theory only when a data type is infinite, e.g., integers. If it is hard for the reader to understand infinite-dimensional Hilbert spaces and associated concepts (e.g., limits in Definition 2.1.3, closed subspaces in Definition 2.1.6 following), she/he can simply focus on finite-dimensional Hilbert spaces, which are exactly the vector spaces that were learned in elementary linear algebra; in this way, the reader can still grasp an essential part of this book.

Whenever $\mathcal{H}$ is finite-dimensional, say $\dim \mathcal{H} = n$, and we consider a *fixed* orthonormal basis $\{|\psi_1\rangle, |\psi_2\rangle, \ldots, |\psi_n\rangle\}$, then each vector $|\psi\rangle = \sum_{i=1}^{n} \lambda_i |\psi_i\rangle \in \mathcal{H}$ can be represented by the vector in $\mathbb{C}^n$:

$$\begin{pmatrix} \lambda_1 \\ \cdots \\ \lambda_n \end{pmatrix}$$

The notion of subspace is also important for understanding the structure of a Hilbert space.

Definition 2.1.6. *Let $\mathcal{H}$ be a Hilbert space.*

(i) *If $X \subseteq \mathcal{H}$, and for any $|\varphi\rangle, |\psi\rangle \in X$ and $\lambda \in \mathbb{C}$,*
  (a) *$|\varphi\rangle + |\psi\rangle \in X$;*
  (b) *$\lambda|\varphi\rangle \in X$,*
  then X is called a subspace of $\mathcal{H}$.

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

(ii) *For each $X \subseteq \mathcal{H}$, its closure $\overline{X}$ is the set of limits $\lim_{n\to\infty} |\psi_n\rangle$ of sequences $\{|\psi_n\rangle\}$ in X.*

(iii) *A subspace X of $\mathcal{H}$ is closed if $\overline{X} = X$.*

For any subset $X \subseteq \mathcal{H}$, the space spanned by X:

$$span X = \left\{ \sum_{i=1}^{n} \lambda_i |\psi_i\rangle : n \geq 0, \lambda_i \in \mathbb{C} \text{ and } |\psi_i\rangle \in X \ (i = 1, \ldots, n) \right\} \tag{2.1}$$

is the smallest subspace of $\mathcal{H}$ containing X. In other words, *spanX* is the subspace of $\mathcal{H}$ generated by X. Moreover, $\overline{span X}$ is the closed subspace generated by X.

We defined orthogonality between two vectors previously. It can be further defined between two sets of vectors.

Definition 2.1.7. *Let $\mathcal{H}$ be a Hilbert space.*

(i) *For any $X, Y \subseteq \mathcal{H}$, we say that X and Y are orthogonal, written $X \perp Y$, if $|\varphi\rangle \perp |\psi\rangle$ for all $|\varphi\rangle \in X$ and $|\psi\rangle \in Y$. In particular, we simply write $|\varphi\rangle \perp Y$ if X is the singleton $\{|\varphi\rangle\}$.*

(ii) *The orthocomplement of a closed subspace X of $\mathcal{H}$ is*

$$X^{\perp} = \{|\varphi\rangle \in \mathcal{H} : |\varphi\rangle \perp X\}.$$

The orthocomplement $X^{\perp}$ is also a closed subspace of $\mathcal{H}$, and we have $(X^{\perp})^{\perp} = X$ for every closed subspace X of $\mathcal{H}$.

Definition 2.1.8. *Let $\mathcal{H}$ be a Hilbert space, and let X, Y be two subspaces of $\mathcal{H}$. Then*

$$X \oplus Y = \{|\varphi\rangle + |\psi\rangle : |\varphi\rangle \in X \text{ and } |\psi\rangle \in Y\}$$

is called the sum of X and Y.

This definition can be straightforwardly generalized to the sum $\bigoplus_{i=1}^{n} X_i$ of more than two subspaces X_i of $\mathcal{H}$. In particular, if X_i $(1 \leq i \leq n)$ are orthogonal to each other, then $\bigoplus_{i=1}^{n} X_i$ is called an orthogonal sum.

With the above preparation, we can present:

- **Postulate of quantum mechanics 1:** The state space of a closed (i.e., an isolated) quantum system is represented by a Hilbert space, and a pure state of the system is described by a unit vector in its state space.

A linear combination $|\psi\rangle = \sum_{i=1}^{n} \lambda_i |\psi_i\rangle$ of states $|\psi_1\rangle, \ldots, |\psi_n\rangle$ is often called their *superposition*, and the complex coefficients λ_i are called probability amplitudes.

Example 2.1.1. *A qubit – quantum bit – is the quantum counterpart of a bit. Its state space is the two-dimensional Hilbert space:*

$$\mathcal{H}_2 = \mathbb{C}^2 = \{\alpha|0\rangle + \beta|1\rangle : \alpha, \beta \in \mathbb{C}\}.$$

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

The inner product in $\mathcal{H}_2$ is defined by

$$(\alpha|0\rangle + \beta|1\rangle, \alpha'|0\rangle + \beta'|1\rangle) = \alpha^*\alpha' + \beta^*\beta'$$

for all $\alpha, \alpha', \beta, \beta' \in \mathbb{C}$. Then $\{|0\rangle, |1\rangle\}$ is an orthonormal basis of $\mathcal{H}_2$, called the computational basis. The vectors $|0\rangle, |1\rangle$ themselves are represented as

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

in this basis. A state of a qubit is described by a unit vector $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ with $|\alpha|^2 + |\beta|^2 = 1$. The two vectors:

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

form another orthonormal basis. Both of them are superpositions of $|0\rangle$ and $|1\rangle$. The two-dimensional Hilbert space $\mathcal{H}_2$ can also be seen as the quantum counterpart of the classical Boolean data type.

Example 2.1.2. *Another Hilbert space often used in this book is the space of square summable sequences:*

$$\mathcal{H}_\infty = \left\{ \sum_{n=-\infty}^{\infty} \alpha_n |n\rangle : \alpha_n \in \mathbb{C} \text{ for all } n \in \mathbb{Z} \text{ and } \sum_{n=-\infty}^{\infty} |\alpha_n|^2 < \infty \right\},$$

where $\mathbb{Z}$ is the set of integers. The inner product in $\mathcal{H}_\infty$ is defined by

$$\left(\sum_{n=-\infty}^{\infty} \alpha_n |n\rangle, \sum_{n=-\infty}^{\infty} \alpha' |n\rangle \right) = \sum_{n=-\infty}^{\infty} \alpha_n^* \alpha_n'$$

for all $\alpha_n, \alpha_n' \in \mathbb{C}$ $(-\infty < n < \infty)$. Then $\{|n\rangle : n \in \mathbb{Z}\}$ is an orthonormal basis, and $\mathcal{H}_\infty$ is infinite-dimensional. This Hilbert space can be seen as the quantum counterpart of the classical integer data type.

Exercise 2.1.1. *Verify that the inner products defined in the previous two examples satisfy conditions (i)–(iii) in Definition 2.1.2.*

2.1.2 LINEAR OPERATORS

We studied the static description of a quantum system, namely its state space as a Hilbert space, in the previous subsection. Now we turn to learning how to describe the dynamics of a quantum system. The evolution of and all operations on a quantum system can be depicted by linear operators in its state Hilbert space. So, in this subsection, we study linear operators and their matrix representations.

Definition 2.1.9. *Let $\mathcal{H}$ and $\mathcal{K}$ be Hilbert spaces. A mapping*

$$A : \mathcal{H} \to \mathcal{K}$$

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

is called an (a linear) operator if it satisfies the following conditions:

(i) $A(|\varphi\rangle + |\psi\rangle) = A|\varphi\rangle + A|\psi\rangle$;
(ii) $A(\lambda|\psi\rangle) = \lambda A|\psi\rangle$

for all $|\varphi\rangle, |\psi\rangle \in \mathcal{H}$ *and* $\lambda \in \mathbb{C}$.

An operator from $\mathcal{H}$ to itself is called an operator in $\mathcal{H}$. The identity operator in $\mathcal{H}$ that maps each vector in $\mathcal{H}$ to itself is denoted $I_{\mathcal{H}}$, and the zero operator in $\mathcal{H}$ that maps every vector in $\mathcal{H}$ to the zero vector is denoted $0_{\mathcal{H}}$. For any vectors $|\varphi\rangle, |\psi\rangle \in \mathcal{H}$, their outer product is the operator $|\varphi\rangle\langle\psi|$ in $\mathcal{H}$ defined by

$$(|\varphi\rangle\langle\psi|)|\chi\rangle = \langle\psi|\chi\rangle|\varphi\rangle$$

for every $|\chi\rangle \in \mathcal{H}$. A class of simple but useful operators are projectors. Let X be a closed subspace of $\mathcal{H}$ and $|\psi\rangle \in \mathcal{H}$. Then there exist uniquely $|\psi_0\rangle \in X$ and $|\psi_1\rangle \in X^{\perp}$ such that

$$|\psi\rangle = |\psi_0\rangle + |\psi_1\rangle.$$

The vector $|\psi_0\rangle$ is called the projection of $|\psi\rangle$ onto X and written $|\psi_0\rangle = P_X|\psi\rangle$.

Definition 2.1.10. *For each closed subspace* X *of* $\mathcal{H}$*, the operator*

$$P_X : \mathcal{H} \to X, \quad |\psi\rangle \mapsto P_X|\psi\rangle$$

is called the projector onto X.

Exercise 2.1.2. *Show that* $P_X = \sum_i |\psi_i\rangle\langle\psi_i|$ *if* $\{|\psi_i\rangle\}$ *is an orthonormal basis of* X.

Throughout this book, we only consider bounded operators, as defined in the following:

Definition 2.1.11. *An operator* A *in* $\mathcal{H}$ *is said to be bounded if there is a constant* $C \geq 0$ *such that*

$$\|A|\psi\rangle\| \leq C \cdot \|\psi\|$$

for all $|\psi\rangle \in \mathcal{H}$. *The norm of* A *is defined to be the nonnegative number:*

$$\|A\| = \inf\{C \geq 0 : ||A|\psi\rangle|| \leq C \cdot ||\psi|| \text{ for all } \psi \in \mathcal{H}\}.$$

We write $\mathcal{L}(\mathcal{H})$ *for the set of bounded operators in* $\mathcal{H}$.

All operators in a finite-dimensional Hilbert space are bounded.

Various operations of operators are very useful in order to combine several operators to produce a new operator. The addition, scalar multiplication and composition of operators can be defined in a natural way: for any $A, B \in \mathcal{L}(\mathcal{H})$, $\lambda \in \mathbb{C}$ and $|\psi\rangle \in \mathcal{H}$,

$$\begin{aligned}(A + B)|\psi\rangle &= A|\psi\rangle + B|\psi\rangle,\\ (\lambda A)|\psi\rangle &= \lambda(A|\psi\rangle),\\ (BA)|\psi\rangle &= B(A|\psi\rangle).\end{aligned}$$

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

Exercise 2.1.3. *Show that $\mathcal{L}(\mathcal{H})$ with addition and scalar multiplication forms a vector space.*

We can also define positivity of an operator as well as an order and a distance between operators.

Definition 2.1.12. *An operator $A \in \mathcal{L}(\mathcal{H})$ is positive if for all states $|\psi\rangle \in \mathcal{H}$, $\langle\psi|A|\psi\rangle$ is a nonnegative real number: $\langle\psi|A|\psi\rangle \geq 0$.*

Definition 2.1.13. *The Löwner order $\sqsubseteq$ is defined as follows: for any $A, B \in \mathcal{L}(\mathcal{H})$, $A \sqsubseteq B$ if and only if $B - A = B + (-1)A$ is positive.*

Definition 2.1.14. *Let $A, B \in \mathcal{L}(\mathcal{H})$. Then their distance is*

$$d(A,B) = \sup_{|\psi\rangle} ||A|\psi\rangle - B|\psi\rangle|| \tag{2.2}$$

where $|\psi\rangle$ traverses all pure states (i.e., unit vectors) in $\mathcal{H}$.

Matrix Representation of Operators:

Operators in a finite-dimensional Hilbert space have a matrix representation, which is very convenient in applications. After reading this part, the reader should have a better understanding of those abstract notions defined previously through a connection from them to the corresponding notions that she/he learned in elementary linear algebra.

If $\{|\psi_i\rangle\}$ is an orthonormal basis of $\mathcal{H}$, then an operator A is uniquely determined by the images $A|\psi_i\rangle$ of the basis vectors $|\psi_i\rangle$ under A. In particular, when $\dim \mathcal{H} = n$ is finite and we consider a *fixed* orthonormal basis $\{|\psi_1\rangle, \ldots, |\psi_n\rangle\}$, A can be represented by the $n \times n$ complex matrix:

$$A = \left(a_{ij}\right)_{n\times n} = \begin{pmatrix} a_{11} & \ldots & a_{1n} \\ & \ldots & \\ a_{n1} & \ldots & a_{nn} \end{pmatrix}$$

where

$$a_{ij} = \langle\psi_i|A|\psi_j\rangle = (|\psi_i\rangle, A|\psi_j\rangle)$$

for every $i, j = 1, \ldots, n$. Moreover, the image of a vector $|\psi\rangle = \sum_{i=1}^{n} \alpha_i|\psi_i\rangle \in \mathcal{H}$ under operator A is represented by the product of matrix $A = \left(a_{ij}\right)_{n\times n}$ and vector $|\psi\rangle$:

$$A|\psi\rangle = A \begin{pmatrix} \alpha_1 \\ \ldots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \ldots \\ \beta_n \end{pmatrix}$$

where $\beta_i = \sum_{j=1}^{n} a_{ij}\alpha_j$ for every $i = 1, \ldots, n$. For example, $I_{\mathcal{H}}$ is the unit matrix, and $0_{\mathcal{H}}$ is the zero matrix. If

$$|\varphi\rangle = \begin{pmatrix} \alpha_1 \\ \ldots \\ \alpha_n \end{pmatrix}, \qquad |\psi\rangle = \begin{pmatrix} \beta_1 \\ \ldots \\ \beta_n \end{pmatrix},$$

then their outer product is the matrix $|\varphi\rangle\langle\psi| = \left(a_{ij}\right)_{n\times n}$ with $a_{ij} = \alpha_i\beta_j^*$ for every $i, j = 1, \ldots, n$. Throughout this book, we do not distinguish an operator in a finite-dimensional Hilbert space from its matrix representation.

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

Exercise 2.1.4. *Show that in a finite-dimensional Hilbert space, addition, scalar multiplication and composition of operators correspond to addition, scalar multiplication and multiplication of their matrix representations, respectively.*

2.1.3 UNITARY TRANSFORMATIONS

The postulate of quantum mechanics 1 introduced in Subsection 2.1.1 provides the static description of a quantum system. In this subsection, we give a description of the dynamics of a quantum system, with the mathematical tool prepared in the last subsection.

The continuous-time dynamics of a quantum system are given by a differential equation, called the Schrödinger equation. But in quantum computation, we usually consider the discrete-time evolution of a system – a unitary transformation. For any operator $A \in \mathcal{L}(\mathcal{H})$, it turns out that there exists a unique (linear) operator $A^\dagger$ in $\mathcal{H}$ such that

$$(A|\varphi\rangle, |\psi\rangle) = \left(|\varphi\rangle, A^\dagger|\psi\rangle\right)$$

for all $|\varphi\rangle, |\psi\rangle \in \mathcal{H}$. The operator $A^\dagger$ is called the adjoint of A. In particular, if an operator in an n-dimensional Hilbert space is represented by the matrix $A = \left(a_{ij}\right)_{n\times n}$, then its adjoint is represented by the transpose conjugate of A:

$$A^\dagger = \left(b_{ij}\right)_{n\times n}$$

with $b_{ij} = a_{ji}^*$ for every $i, j = 1, \ldots, n$.

Definition 2.1.15. *An (bounded) operator $U \in \mathcal{L}(\mathcal{H})$ is called a unitary transformation if the adjoint of U is its inverse:*

$$U^\dagger U = UU^\dagger = I_\mathcal{H}.$$

All unitary transformations U preserve the inner product:

$$(U|\varphi\rangle, U|\psi\rangle) = \langle\varphi|\psi\rangle$$

for any $|\varphi\rangle, |\psi\rangle \in \mathcal{H}$. The condition $U^\dagger U = I_\mathcal{H}$ is equivalent to $UU^\dagger = I_\mathcal{H}$ when $\mathcal{H}$ is finite-dimensional. If $\dim \mathcal{H} = n$, then a unitary operator in $\mathcal{H}$ is represented by an $n \times n$ unitary matrix U; i.e., a matrix U with $U^\dagger U = I_n$, where I_n is the n-dimensional unit matrix.

A useful technique for defining a unitary operator is given in the following:

Lemma 2.1.1. *Suppose that $\mathcal{H}$ is a (finite-dimensional) Hilbert space and $\mathcal{K}$ is a closed subspace of $\mathcal{H}$. If linear operator $U : \mathcal{K} \to \mathcal{H}$ preserves the inner product:*

$$(U|\varphi\rangle, U|\psi\rangle) = \langle\varphi|\psi\rangle$$

for any $|\varphi\rangle, |\psi\rangle \in \mathcal{K}$, then there exists a unitary operator V in $\mathcal{H}$ which extends U; i.e., $V|\psi\rangle = U|\psi\rangle$ for all $|\psi\rangle \in \mathcal{K}$.

Exercise 2.1.5. *Prove Lemma 2.1.1.*

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

Now we are ready to present:

- **Postulate of quantum mechanics 2:** Suppose that the states of a closed quantum system (i.e., a system without interactions with its environment) at times t_0 and t are $|\psi_0\rangle$ and $|\psi\rangle$, respectively. Then they are related to each other by a unitary operator U which depends only on the times t_0 and t,

$$|\psi\rangle = U|\psi_0\rangle.$$

To help the reader understand this postulate, let us consider two simple examples.

Example 2.1.3. *One frequently used unitary operator on a qubit is the Hadamard transformation in the two-dimensional Hilbert space $\mathcal{H}_2$:*

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

It transforms a qubit in the computational basis states $|0\rangle$ and $|1\rangle$ into their superpositions:

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

Example 2.1.4. *Let k be an integer. Then the k-translation operator T_k in the infinite-dimensional Hilbert space $\mathcal{H}_\infty$ is defined by*

$$T_k|n\rangle = |n+k\rangle$$

for all $n \in \mathbb{Z}$. It is easy to verify that T_k is a unitary operator. In particular, we write $T_L = T_{-1}$ and $T_R = T_1$. They move a particle on the line one position to the left and to the right, respectively.

More examples will be seen in Section 2.2, where unitary transformations are used as quantum logic gates in a quantum circuit.

2.1.4 QUANTUM MEASUREMENTS

Now that we understand both the static and dynamic descriptions of a quantum system, observation of a quantum system is carried out through a quantum measurement, which is defined by:

- **Postulate of quantum mechanics 3:** A quantum measurement on a system with state Hilbert space $\mathcal{H}$ is described by a collection $\{M_m\} \subseteq \mathcal{L}(\mathcal{H})$ of operators satisfying the normalization condition:

$$\sum_m M_m^\dagger M_m = I_{\mathcal{H}}, \tag{2.3}$$

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

where M_m are called measurement operators, and the index m stands for the measurement outcomes that may occur in the experiment. If the state of a quantum system is $|\psi\rangle$ immediately before the measurement, then for each m, the probability that the result m occurs in the measurement is

$$p(m) = ||M_m|\psi\rangle||^2 = \langle\psi|M_m^\dagger M_m|\psi\rangle \qquad \text{(Born rule)}$$

and the state of the system after the measurement with outcome m is

$$|\psi_m\rangle = \frac{M_m|\psi\rangle}{\sqrt{p(m)}}.$$

It is easy to see that the normalization condition (2.3) implies that the probabilities for all outcomes sum up to $\sum_m p(m) = 1$.

The following simple example should help the reader to understand this postulate.

Example 2.1.5. *The measurement of a qubit in the computational basis has two outcomes defined by measurement operators:*

$$M_0 = |0\rangle\langle 0|, \qquad M_1 = |1\rangle\langle 1|.$$

If the qubit was in state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ *before the measurement, then the probability of obtaining outcome* 0 *is*

$$p(0) = \langle\psi|M_0^\dagger M_0|\psi\rangle = \langle\psi|M_0|\psi\rangle = |\alpha|^2,$$

and in this case the state after the measurement is

$$\frac{M_0|\psi\rangle}{\sqrt{p(0)}} = |0\rangle.$$

Similarly, the probability of outcome 1 *is* $p(1) = |\beta|^2$ *and in this case the state after the measurement is* $|1\rangle$.

Projective Measurements:

A specially useful class of measurements is defined in terms of Hermitian operators and their spectral decomposition.

Definition 2.1.16. *An operator* $M \in \mathcal{L}(\mathcal{H})$ *is said to be Hermitian if it is self-adjoint:*

$$M^\dagger = M.$$

In physics, a Hermitian operator is also called an observable.

It turns out that an operator P is a projector; that is, $P = P_X$ for some closed subspace X of $\mathcal{H}$, if and only if P is Hermitian and $P^2 = P$.

A quantum measurement can be constructed from an observable based on the mathematical concept of spectral decomposition of a Hermitian operator. Due to the limit of space, we only consider spectral decomposition in a finite-dimensional Hilbert space $\mathcal{H}$. (The infinite-dimensional case requires a much heavier

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

mathematical mechanism; see [182], Chapter III.5. In this book, it will be used only in Section 3.6 as a tool for the proof of a technical lemma.)

Definition 2.1.17

(i) *An eigenvector of an operator $A \in \mathcal{L}(\mathcal{H})$ is a non-zero vector $|\psi\rangle \in \mathcal{H}$ such that $A|\psi\rangle = \lambda|\psi\rangle$ for some $\lambda \in \mathbb{C}$, where λ is called the eigenvalue of A corresponding to $|\psi\rangle$.*

(ii) *The set of eigenvalues of A is called the (point) spectrum of A and denoted $spec(A)$.*

(iii) *For each eigenvalue $\lambda \in spec(A)$, the set*

$$\{|\psi\rangle \in \mathcal{H} : A|\psi\rangle = \lambda|\psi\rangle\}$$

is a closed subspace of $\mathcal{H}$ and it is called the eigenspace of A corresponding to λ.

The eigenspaces corresponding to different eigenvalues $\lambda_1 \neq \lambda_2$ are orthogonal. All eigenvalues of an observable (i.e., a Hermitian operator) M are real numbers. Moreover, it has the spectral decomposition:

$$M = \sum_{\lambda \in spec(M)} \lambda P_\lambda$$

where P_λ is the projector onto the eigenspace corresponding to λ. Then it defines a measurement $\{P_\lambda : \lambda \in spec(M)\}$, called a projective measurement because all measurement operators P_λ are projectors. Using the Postulate of quantum mechanics 3 introduced earlier, we obtain: upon measuring a system in state $|\psi\rangle$, the probability of getting result λ is

$$p(\lambda) = \langle\psi|P_\lambda^\dagger P_\lambda|\psi\rangle = \langle\psi|P_\lambda^2|\psi\rangle = \langle\psi|P_\lambda|\psi\rangle \tag{2.4}$$

and in this case the state of the system after the measurement is

$$\frac{P_\lambda|\psi\rangle}{\sqrt{p(\lambda)}}. \tag{2.5}$$

Since all possible outcomes $\lambda \in spec(M)$ are real numbers, we can compute the expectation – average value – of M in state $|\psi\rangle$:

$$\begin{aligned}\langle M\rangle_\psi &= \sum_{\lambda \in spec(M)} p(\lambda)\cdot\lambda \\ &= \sum_{\lambda \in spec(M)} \lambda\langle\psi|P_\lambda|\psi\rangle \\ &= \langle\psi| \sum_{\lambda \in spec(M)} \lambda P_\lambda|\psi\rangle \\ &= \langle\psi|M|\psi\rangle.\end{aligned}$$

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

We observe that, given the state $|\psi\rangle$, probability (2.4) and post-measurement state (2.5) are determined only by the projectors $\{P_\lambda\}$ (rather than M itself). It is easy to see that $\{P_\lambda\}$ is a complete set of orthogonal projectors; that is, a set of operators satisfying the conditions:

(i) $P_\lambda P_\delta = \begin{cases} P_\lambda & \text{if } \lambda = \delta, \\ 0_{\mathcal{H}} & \text{otherwise;} \end{cases}$

(ii) $\sum_\lambda P_\lambda = I_{\mathcal{H}}$.

Sometimes, we simply call a complete set of orthogonal projectors a projective measurement. A special case is the measurement in an orthonormal basis $\{|i\rangle\}$ of the state Hilbert space, where $P_i = |i\rangle\langle i|$ for every i. Example 2.1.5 is such a measurement for a qubit.

2.1.5 TENSOR PRODUCTS OF HILBERT SPACES

Up to now we have only considered a single quantum system. In this section, we further show how a large composite system can be made up of two or more subsystems. The description of a composite system is based on the notion of tensor product. We mainly consider the tensor product of a finite family of Hilbert spaces.

Definition 2.1.18. *Let $\mathcal{H}_i$ be a Hilbert space with $\{|\psi_{ij_i}\rangle\}$ as an orthonormal basis for $i = 1, \ldots, n$. We write $\mathcal{B}$ for the set having elements of the form:*

$$|\psi_{1j_1}, \ldots, \psi_{nj_n}\rangle = |\psi_{1j_1} \otimes \ldots \otimes \psi_{nj_n}\rangle = |\psi_{1j_1}\rangle \otimes \ldots \otimes |\psi_{nj_n}\rangle.$$

Then the tensor product of $\mathcal{H}_i$ $(i = 1, \ldots, n)$ is the Hilbert space with $\mathcal{B}$ as an orthonormal basis:

$$\bigotimes_i \mathcal{H}_i = span\mathcal{B}.$$

It follows from equation (2.1) that each element in $\bigotimes_i \mathcal{H}_i$ can be written in the form of

$$\sum_{j_1,\ldots,j_n} \alpha_{j_1,\ldots,j_n} |\varphi_{1j_1}, \ldots, \varphi_{nj_n}\rangle$$

where $|\varphi_{1j_1}\rangle \in \mathcal{H}_1, \ldots, |\varphi_{nj_n}\rangle \in \mathcal{H}_n$ and $\alpha_{j_1,\ldots,j_n} \in \mathbb{C}$ for all $j_1, \ldots, j_n$. Furthermore, it can be shown by linearity that the choice of basis $\{|\psi_{ij_i}\rangle\}$ of each factor space $\mathcal{H}_i$ is not essential in the previous definition: for example, if $|\varphi_i\rangle = \sum_{j_i} \alpha_{j_i} |\varphi_{ij_i}\rangle \in \mathcal{H}_i$ $(i = 1, \ldots, n)$, then

$$|\varphi_1\rangle \otimes \ldots |\varphi_n\rangle = \sum_{j_1,\ldots,j_n} \alpha_{1j_1} \ldots \alpha_{nj_n} |\varphi_{1j_1}, \ldots, \varphi_{nj_n}\rangle.$$

The vector addition, scalar multiplication and inner product in $\bigotimes_i \mathcal{H}_i$ can be naturally defined based on the fact that $\mathcal{B}$ is an orthonormal basis.

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

We will need to consider the tensor product of a countably infinite family of Hilbert spaces occasionally in this book. Let $\{\mathcal{H}_i\}$ be a countably infinite family of Hilbert spaces, and let $\{|\psi_{ij_i}\rangle\}$ be an orthonormal basis of $\mathcal{H}_i$ for each i. We write $\mathcal{B}$ for the set of tensor products of basis vectors of all $\mathcal{H}_i$:

$$\mathcal{B} = \left\{ \bigotimes_i |\psi_{ij_i}\rangle \right\}.$$

Then $\mathcal{B}$ is a finite or countably infinite set, and it can be written in the form of a sequence of vectors: $\mathcal{B} = \{|\varphi_n\rangle : n = 0, 1, \ldots\}$. The tensor product of $\{\mathcal{H}_i\}$ can be properly defined to be the Hilbert space with $\mathcal{B}$ as an orthonormal basis:

$$\bigotimes_i \mathcal{H}_i = \left\{ \sum_n \alpha_n |\varphi_n\rangle : \alpha_n \in \mathbb{C} \text{ for all } n \geq 0 \text{ and } \sum_n |\alpha_n|^2 < \infty \right\}.$$

Now we are able to present:

- **Postulate of quantum mechanics 4:** The state space of a composite quantum system is the tensor product of the state spaces of its components.

Suppose that S is a quantum system composed of subsystems $S_1, \ldots, S_n$ with state Hilbert space $\mathcal{H}_1, \ldots, \mathcal{H}_n$. If for each $1 \leq i \leq n$, S_i is in state $|\psi_i\rangle \in \mathcal{H}_i$, then S is in the product state $|\psi_1, \ldots, \psi_n\rangle$. Furthermore, S can be in a superposition (i.e., linear combination) of several product states. One of the most interesting and puzzling phenomenon in quantum mechanics – *entanglement* – occurs in a composite system: a state of the composite system is said to be entangled if it is not a product of states of its component systems. The existence of entanglement is one of the major differences between the classical world and the quantum world.

Example 2.1.6. *The state space of the system of n qubits is:*

$$\mathcal{H}_2^{\otimes n} = \mathbb{C}^{2^n} = \left\{ \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle : \alpha_x \in \mathbb{C} \text{ for all } x \in \{0,1\}^n \right\}.$$

In particular, a two-qubit system can be in a product state such as $|00\rangle, |1\rangle|+\rangle$ *but also in an entangled state such as the Bell states or the EPR (Einstein-Podolsky-Rosen) pairs:*

$$|\beta_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \quad |\beta_{01}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle),$$
$$|\beta_{10}\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle), \quad |\beta_{11}\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle).$$

Of course, we can talk about (linear) operators, unitary transformations and measurements in the tensor product of Hilbert spaces since it is a Hilbert space too. A special class of operators in the tensor product of Hilbert spaces is defined as follows:

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

Definition 2.1.19. *Let $A_i \in \mathcal{L}(\mathcal{H}_i)$ for $i = 1, \ldots, n$. Then their tensor product is the operator $\bigotimes_{i=1}^{n} A_i = A_1 \otimes \ldots \otimes A_n \in \mathcal{L}\left(\bigotimes_{i=1}^{n} \mathcal{H}_i\right)$ defined by*

$$(A_1 \otimes \ldots \otimes A_n)|\varphi_1, \ldots, \varphi_n\rangle = A_1|\varphi_1\rangle \otimes \ldots \otimes A_n|\varphi_n\rangle$$

for all $|\varphi_i\rangle \in \mathcal{H}_i$ $(i = 1, \ldots, n)$ together with linearity.

But other operators rather than tensor products are indispensable in quantum computation because they can create entanglement.

Example 2.1.7. *The controlled-NOT or CNOT operator C in the state Hilbert space $\mathcal{H}_2^{\otimes 2} = \mathbb{C}^4$ of a two-qubit system is defined by*

$$C|00\rangle = |00\rangle, \quad C|01\rangle = |01\rangle, \quad C|10\rangle = |11\rangle, \quad C|11\rangle = |10\rangle$$

or equivalently as the 4×4 matrix

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

It can transform product states into entangled states:

$$C|+\rangle|0\rangle = \beta_{00}, \quad C|+\rangle|1\rangle = \beta_{01}, \quad C|-\rangle|0\rangle = \beta_{10}, \quad C|-\rangle|1\rangle = \beta_{11}.$$

Implementing a General Measurement by a Projective Measurement:

Projective measurements are introduced in subsection 2.1.4 as a special class of quantum measurements. The notion of tensor product enables us to show that an arbitrary quantum measurement can be implemented by a projective measurement together with a unitary transformation if we are allowed to introduce an *ancilla* system. Let $M = \{M_m\}$ be a quantum measurement in Hilbert space $\mathcal{H}$.

- We introduce a new Hilbert space $\mathcal{H}_M = span\{|m\rangle\}$, which is used to record the possible outcomes of M.
- We arbitrarily choose a fixed state $|0\rangle \in \mathcal{H}_M$. Define operator

$$U_M(|0\rangle|\psi\rangle) = \sum_m |m\rangle M_m|\psi\rangle$$

  for every $|\psi\rangle \in \mathcal{H}$. It is easy to check that U_M preserves the inner product, and by Lemma 2.1.1 it can be extended to a unitary operator in $\mathcal{H}_M \otimes \mathcal{H}$, which is denoted by U_M too.
- We define a projective measurement $\overline{M} = \{\overline{M}_m\}$ in $\mathcal{H}_M \otimes \mathcal{H}$ with $\overline{M}_m = |m\rangle\langle m| \otimes I_{\mathcal{H}}$ for every m.

Then the measurement M is realized by the projective measurement $\overline{M}$ together with the unitary operator U_M, as shown in the following:

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

Proposition 2.1.1. *Let $|\psi\rangle \in \mathcal{H}$ be a pure state.*

- *When we perform measurement M on $|\psi\rangle$, the probability of outcome m is denoted $p_M(m)$ and the post-measurement state corresponding to m is $|\psi_m\rangle$.*
- *When we perform measurement $\overline{M}$ on $|\overline{\psi}\rangle = U_M(|0\rangle|\psi\rangle)$, the probability of outcome m is denoted $p_{\overline{M}}(m)$ and the post-measurement state corresponding to m is $|\overline{\psi}_m\rangle$.*

Then for each m, we have: $p_{\overline{M}}(m) = p_M(m)$ and $|\overline{\psi}_m\rangle = |m\rangle|\psi_m\rangle$. A similar result holds when we consider a mixed state in $\mathcal{H}$ introduced in the next subsection.

Exercise 2.1.6. *Prove Proposition 2.1.1.*

2.1.6 DENSITY OPERATORS

We have already learned all of the four basic postulates of quantum mechanics. But they were only formulated in the case of pure states. In this section, we extend these postulates so that they can be used to deal with mixed states.

Sometimes, the state of a quantum system is not completely known, but we know that it is in one of a number of pure states $|\psi_i\rangle$, with respective probabilities p_i, where $|\psi_i\rangle \in \mathcal{H}$, $p_i \geq 0$ for each i, and $\sum_i p_i = 1$. A convenient notion for coping with this situation is the density operator. We call $\{(|\psi_i\rangle, p_i)\}$ an ensemble of pure states or a mixed state, whose density operator is defined to be

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|. \tag{2.6}$$

In particular, a pure state $|\psi\rangle$ may be seen as a special mixed state $\{(|\psi\rangle, 1)\}$ and its density operator is $\rho = |\psi\rangle\langle\psi|$.

Density operators can be described in a different but equivalent way.

Definition 2.1.20. *The trace $tr(A)$ of operator $A \in \mathcal{L}(\mathcal{H})$ is defined to be*

$$tr(A) = \sum_i \langle\psi_i|A|\psi_i\rangle$$

where $\{|\psi_i\rangle\}$ is an orthonormal basis of $\mathcal{H}$.

It can be shown that $tr(A)$ is independent of the choice of basis $\{|\psi_i\rangle\}$.

Definition 2.1.21. *A density operator ρ in a Hilbert space $\mathcal{H}$ is a positive operator (see Definition 2.1.12) with $tr(\rho) = 1$.*

It turns out that for any mixed state $\{(|\psi_i\rangle, p_i)\}$, operator ρ defined by equation (2.6) is a density operator according to Definition 2.1.21. Conversely, for any density operator ρ, there exists a (but not necessarily unique) mixed state $\{(|\psi_i\rangle, p_i)\}$ such that equation (2.6) holds.

The evolution of and a measurement on a quantum system in mixed states can be elegantly formulated in the language of density operators:

- Suppose that the evolution of a closed quantum system from time t_0 to t is described by unitary operator U depending on t_0 and t: $|\psi\rangle = U|\psi_0\rangle$, where

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

$|\psi_0\rangle, |\psi\rangle$ are the states of the system at times t_0 and t, respectively. If the system is in mixed states ρ_0, ρ at times t_0 and t, respectively, then

$$\rho = U\rho_0 U^\dagger. \tag{2.7}$$

- If the state of a quantum system was ρ immediately before measurement $\{M_m\}$ is performed on it, then the probability that result m occurs is

$$p(m) = tr\left(M_m^\dagger M_m \rho\right), \tag{2.8}$$

and in this case the state of the system after the measurement is

$$\rho_m = \frac{M_m \rho M_m^\dagger}{p(m)}. \tag{2.9}$$

Exercise 2.1.7. *Derive equations (2.7), (2.8) and (2.9) from equation (2.6) and Postulates of quantum mechanics 1 and 2.*

Exercise 2.1.8. *Let M be an observable (a Hermitian operator) and $\{P_\lambda : \lambda \in spec(M)\}$ the projective measurement defined by M. Show that the expectation of M in a mixed state ρ is*

$$\langle M\rangle_\rho = \sum_{\lambda \in spec(M)} p(\lambda)\cdot\lambda = tr(M\rho).$$

Reduced Density Operators:

Postulate of quantum mechanics 4 introduced in the last subsection enables us to construct composite quantum systems. Of course, we can talk about a mixed state of a composite system and its density operator because the state space of the composite system is the tensor product of the state Hilbert spaces of its subsystems, which is a Hilbert space too. Conversely, we often need to characterize the state of a subsystem of a quantum system. However, it is possible that a composite system is in a pure state, but some of its subsystems must be seen as in a mixed state. This phenomenon is another major difference between the classical world and the quantum world. Consequently, a proper description of the state of a subsystem of a composite quantum system can be achieved only after introducing the notion of density operator.

Definition 2.1.22. *Let S and T be quantum systems whose state Hilbert spaces are $\mathcal{H}_S$ and $\mathcal{H}_T$, respectively. The partial trace over system T*

$$tr_T : \mathcal{L}(\mathcal{H}_S \otimes \mathcal{H}_T) \to \mathcal{L}(\mathcal{H}_S)$$

is defined by

$$tr_{\mathcal{T}}(|\varphi\rangle\langle\psi| \otimes |\theta\rangle\langle\zeta|) = \langle\zeta|\theta\rangle \cdot |\varphi\rangle\langle\psi|$$

for all $|\varphi\rangle, |\psi\rangle \in \mathcal{H}_S$ and $|\theta\rangle, |\zeta\rangle \in \mathcal{H}_T$ together with linearity.

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

Definition 2.1.23. *Let ρ be a density operator in $\mathcal{H}_S \otimes \mathcal{H}_T$. Its reduced density operator for system S is*

$$\rho_S = tr_T(\rho).$$

Intuitively, the reduced density operator ρ_S properly describes the state of subsystem S when the composite system ST is in state ρ. For a more detailed explanation, we refer to [174], Section 2.4.3.

Exercise 2.1.9

(i) *When is the reduced density operator $\rho_A = tr_B(|\psi\rangle\langle\psi|)$ of a pure state $|\psi\rangle$ in $\mathcal{H}_A \otimes \mathcal{H}_B$ not a pure state?*

(ii) *Let ρ be a density operator in $\mathcal{H}_A \otimes \mathcal{H}_B \otimes \mathcal{H}_C$. Does it hold that $tr_C(tr_B(\rho)) = tr_{BC}(\rho)$?*

2.1.7 QUANTUM OPERATIONS

Unitary transformations defined in Section 2.1.3 are suited to describe the dynamics of closed quantum systems. For open quantum systems that interact with the outside world through, for example, measurements, we need the much more general notion of quantum operation to depict their state transformations.

A linear operator in vector space $\mathcal{L}(\mathcal{H})$ – the space of (bounded) operators in a Hilbert space $\mathcal{H}$ – is called a *super-operator* in $\mathcal{H}$. To define a quantum operation, we first introduce the notion of tensor product of super-operators.

Definition 2.1.24. *Let $\mathcal{H}$ and $\mathcal{K}$ be Hilbert spaces. For any super-operator $\mathcal{E}$ in $\mathcal{H}$ and super-operator $\mathcal{F}$ in $\mathcal{K}$, their tensor product $\mathcal{E} \otimes \mathcal{F}$ is the super-operator in $\mathcal{H} \otimes \mathcal{K}$ defined as follows: for each $C \in \mathcal{L}(\mathcal{H} \otimes \mathcal{K})$, we can write:*

$$C = \sum_k \alpha_k (A_k \otimes B_k) \tag{2.10}$$

where $A_k \in \mathcal{L}(\mathcal{H})$ and $B_k \in \mathcal{L}(\mathcal{K})$ for all k. Then we define:

$$(\mathcal{E} \otimes \mathcal{F})(C) = \sum_k \alpha_k (\mathcal{E}(A_k) \otimes \mathcal{F}(B_k)).$$

The linearity of $\mathcal{E}$ and $\mathcal{F}$ guarantees that $\mathcal{E} \otimes \mathcal{F}$ is well-defined: $(\mathcal{E} \otimes \mathcal{F})(C)$ is independent of the choice of A_k and B_k in equation (2.10).

Now we are ready to consider the dynamics of an open quantum system. As a generalization of the Postulate of quantum mechanics 2, suppose that the states of a system at times t_0 and t are ρ and ρ', respectively. Then they must be related to each other by a super-operator $\mathcal{E}$ which depends only on the times t_0 and t,

$$\rho' = \mathcal{E}(\rho).$$

The dynamics between times t_0 and t can be seen as a physical process: ρ is the initial state before the process, and $\rho' = \mathcal{E}(\rho)$ is the final state after the process happens. The following definition identifies those super-operators that are suited to model such a process.

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

Definition 2.1.25. *A quantum operation in a Hilbert space $\mathcal{H}$ is a super-operator in $\mathcal{H}$ satisfying the following conditions:*

(i) *$tr[\mathcal{E}(\rho)] \leq tr(\rho) = 1$ for each density operator ρ in $\mathcal{H}$;*

(ii) *(Complete positivity) For any extra Hilbert space $\mathcal{H}_R$, $(\mathcal{I}_R \otimes \mathcal{E})(A)$ is positive provided A is a positive operator in $\mathcal{H}_R \otimes \mathcal{H}$, where $\mathcal{I}_R$ is the identity operator in $\mathcal{L}(\mathcal{H}_R)$; that is, $\mathcal{I}_R(A) = A$ for each operator $A \in \mathcal{L}(\mathcal{H}_R)$.*

For an argument that quantum operations are an appropriate mathematical model of state transformation of an open quantum system, we refer to [174], Section 8.2.4. Here are two examples showing how unitary transformations and quantum measurements can be treated as special quantum operations:

Example 2.1.8. *Let U be a unitary transformation in a Hilbert space $\mathcal{H}$. We define:*

$$\mathcal{E}(\rho) = U\rho U^\dagger$$

for every density operator ρ. Then $\mathcal{E}$ is a quantum operation in $\mathcal{H}$.

Example 2.1.9. *Let $M = \{M_m\}$ be a quantum measurement in $\mathcal{H}$.*

(i) *For each m, if for any system state ρ before measurement, we define*

$$\mathcal{E}_m(\rho) = p_m\rho_m = M_m\rho M^\dagger$$

where p_m is the probability of outcome m and ρ_m is the post-measurement state corresponding to m, then $\mathcal{E}_m$ is a quantum operation.

(ii) *For any system state ρ before measurement, the post-measurement state is*

$$\mathcal{E}(\rho) = \sum_m \mathcal{E}_m(\rho) = \sum_m M_m\rho M_m^\dagger$$

whenever the measurement outcomes are ignored. Then $\mathcal{E}$ is a quantum operation.

Quantum operations have been widely used in quantum information theory as a mathematical model of communication channels. In this book, quantum operations are adopted as the main mathematical tool for defining semantics of quantum programs, because a quantum program may contain not only unitary transformations but also quantum measurements in order to read the middle or final computational results, and thus are better treated as an open quantum system.

The abstract definition of quantum operations given here is hard to use in applications. Fortunately, the following theorem offers a helpful insight into a quantum operation as an interaction between the system and an environment as well as calculation convenience in terms of operators rather than super-operators.

Theorem 2.1.1. *The following statements are equivalent:*

(i) *$\mathcal{E}$ is a quantum operation in a Hilbert space $\mathcal{H}$;*

(ii) *(System-environment model) There is an environment system E with state Hilbert space $\mathcal{H}_E$, and a unitary transformation U in $\mathcal{H}_E \otimes \mathcal{H}$ and a projector P onto some closed subspace of $\mathcal{H}_E \otimes \mathcal{H}$ such that*

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

$$\mathcal{E}(\rho) = tr_E\left[PU(|e_0\rangle\langle e_0| \otimes \rho)U^\dagger P\right]$$

for all density operators ρ in $\mathcal{H}$, where $|e_0\rangle$ is a fixed state in $\mathcal{H}_E$;

(iii) *(Kraus operator-sum representation) There exists a finite or countably infinite set of operators $\{E_i\}$ in $\mathcal{H}$ such that $\sum_i E_i^\dagger E_i \sqsubseteq I$ and*

$$\mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger$$

for all density operators ρ in $\mathcal{H}$. In this case, we often write:

$$\mathcal{E} = \sum_i E_i \circ E_i^\dagger.$$

The proof of this theorem is quite involved and omitted here, and the reader can find it in [174], Chapter 8.

2.2 QUANTUM CIRCUITS

A general framework of quantum mechanics was developed in the previous section. From this section on, we consider how to harness the power of quantum systems to do computation. We start from a lower-level model of quantum computers – quantum circuits.

2.2.1 BASIC DEFINITIONS

Digital circuits for classical computation are made from logic gates acting on Boolean variables. Quantum circuits are the quantum counterparts of digital circuits. Roughly speaking, they are made up of quantum (logic) gates, which are modelled by unitary transformations defined in Subsection 2.1.3.

We use $p, q, q_1, q_2, \ldots$ to denote qubit variables. Graphically, they can be thought of as wires in quantum circuits. A sequence $\overline{q}$ of distinct qubit variables is called a quantum register. Sometimes, the order of variables in the register is not essential. Then the register is identified with the set of qubit variables in it. So, we can use set-theoretic notations for registers:

$$p \in \overline{q}, \quad \overline{p} \subseteq \overline{q}, \quad \overline{p} \cap \overline{q}, \quad \overline{p} \cup \overline{q}, \quad \overline{p} \setminus \overline{q}.$$

For each qubit variable q, we write $\mathcal{H}_q$ for its state Hilbert space, which is isomorphic to the two-dimensional $\mathcal{H}_2$ (see Example 2.1.1). Furthermore, for a set $V = \{q_1, \ldots, q_n\}$ of qubit variables or a quantum register $\overline{q} = q_1, \ldots, q_n$, we write:

$$\mathcal{H}_V = \bigotimes_{q \in V} \mathcal{H}_q = \bigotimes_{i=1}^{n} \mathcal{H}_{q_i} = \mathcal{H}_{\overline{q}}$$

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

for the state space of the composite system consisting of qubits $q_1, \ldots, q_n$. Obviously, $\mathcal{H}_V$ is 2^n-dimensional. Recall that an integer $0 \leq x < 2^n$ can be represented by a string $x_1 \ldots x_n \in \{0,1\}^n$ of n bits:

$$x = \sum_{i=1}^{n} x_i \cdot 2^{i-1}.$$

We shall not distinguish integer x from its binary representation. Thus, each pure state in $\mathcal{H}_V$ can be written as

$$|\psi\rangle = \sum_{x=0}^{2^n-1} \alpha_x |x\rangle$$

where $\{|x\rangle\}$ is called the computational basis of $\mathcal{H}_2^{\otimes n}$.

Definition 2.2.1. *For any positive integer n, if U is a $2^n \times 2^n$ unitary matrix, and $\overline{q} = q_1, \ldots, q_n$ is a quantum register, then*

$$G \equiv U[\overline{q}] \text{ or } G \equiv U[q_1, \ldots, q_n]$$

is called an n-qubit gate and we write $qvar(G) = \{q_1, \ldots, q_n\}$ for the set of (quantum) variables in G.

The gate $G \equiv U[\overline{q}]$ is a unitary transformation in the state Hilbert space $\mathcal{H}_{\overline{q}}$ of $\overline{q}$. We often call unitary matrix U a quantum gate without mentioning the quantum register $\overline{q}$.

Definition 2.2.2. *A quantum circuit is a sequence of quantum gates:*

$$C \equiv G_1 \ldots G_m$$

where $m \geq 1$, and $G_1, \ldots, G_m$ are quantum gates. The set of variables of C is

$$qvar(C) = \bigcup_{i=1}^{m} qvar(G_i).$$

The presentations of quantum gates and quantum circuits in the previous two definitions are somehow similar to the Boolean expressions of classical circuits and convenient for algebraic manipulations. However, they are not illustrative. Indeed, quantum circuits can be represented graphically as is commonly done for classical circuits; the reader can find graphic illustrations of various quantum circuits in Chapter 4 of book [174], and a macro package for drawing quantum circuit diagrams can be found at http://physics.unm.edu/CQuIC//Qcircuit/.

Let us see how a quantum circuit $C \equiv G_1 \ldots G_m$ computes. Suppose that $qvar(C) = \{q_1, \ldots, q_n\}$, and each gate $G_i = U_i[\overline{r}_i]$, where register $\overline{r}_i$ is a subsequence of $\overline{q} = q_1, \ldots, q_n$, and U_i is a unitary transformation in the space $\mathcal{H}_{\overline{r}_i}$.

- If a state $|\psi\rangle \in \mathcal{H}_{qvar(C)}$ is input to the circuit C, then the output is

$$C|\psi\rangle = \overline{U}_m \ldots \overline{U}_1 |\psi\rangle \tag{2.11}$$

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

where for each i, $\overline{U_i} = U_i \otimes I_i$ is the cylindrical extension of U_i in $\mathcal{H}_C$, and I_i is the identity operator in the space $\mathcal{H}_{\overline{q}\setminus\overline{r}_i}$. Note that the applications of unitary operators $U_1, \ldots, U_m$ in equation (2.11) are in the reverse order of $G_1, \ldots, G_m$ in the circuit C.

- More generally, if $qvar(C) \subsetneqq V$ is a set of qubit variables, then each state $|\psi\rangle \in \mathcal{H}_V$ can be written in the form of

$$|\psi\rangle = \sum_i \alpha_i |\varphi_i\rangle|\zeta_i\rangle$$

with $|\varphi_i\rangle \in \mathcal{H}_{qvar(C)}$ and $|\zeta_i\rangle \in \mathcal{H}_{V\setminus qvar(C)}$. Whenever we input $|\psi\rangle$ to the circuit C, the output is

$$C|\psi\rangle = \sum_i \alpha_i (C|\varphi_i\rangle)|\zeta_i\rangle.$$

The linearity of C guarantees that this output is well-defined.

Now we can define equivalence of quantum circuits whenever their outputs are the same upon the same input.

Definition 2.2.3. *Let C_1, C_2 be quantum circuits and $V = qvar(C_1) \cup qvar(C_2)$. If for any $|\psi\rangle \in \mathcal{H}_V$, we have:*

$$C_1|\psi\rangle = C_2|\psi\rangle, \tag{2.12}$$

then C_1 and C_2 are equivalent and we write $C_1 = C_2$.

A classical circuit with n input wires and m output wires is actually a Boolean function

$$f : \{0,1\}^n \to \{0,1\}^m.$$

Similarly, a quantum circuit C with $qvar(C) = \{q_1, \ldots, q_n\}$ is always equivalent to a unitary transformation in $\mathcal{H}_{qvar(C)}$ or a $2^n \times 2^n$ unitary matrix. This can be clearly seen from equation (2.11).

Finally, we introduce composition of quantum circuits in order to construct a large quantum circuit from small ones.

Definition 2.2.4. *Let $C_1 \equiv G_1 \ldots G_m$ and $C_2 \equiv H_1 \ldots H_n$ be quantum circuits, where $G_1, \ldots, G_m$ and $H_1, \ldots, H_n$ are quantum gates. Then their composition is the concatenation:*

$$C_1 C_2 \equiv G_1 \ldots G_m H_1 \ldots H_n.$$

Exercise 2.2.1

(i) *Prove that if $C_1 = C_2$ then equation (2.12) holds for any state $|\psi\rangle \in \mathcal{H}_V$ and for any $V \supseteq qvar(C_1) \cup qvar(C_2)$.*

(ii) *Prove that if $C_1 = C_2$ then $CC_1 = CC_2$ and $C_1 C = C_2 C$.*

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

2.2.2 ONE-QUBIT GATES

After introducing the general definitions of quantum gates and quantum circuits in the last subsection, let us look at some examples in this subsection.

The simplest quantum gates are one-qubit gates. They are represented by 2×2 unitary matrices. One example is the Hadamard gate presented in Example 2.1.3. The following are some other one-qubit gates that are frequently used in quantum computation.

Example 2.2.1

(i) *Global phase shift:*

$$M(\alpha) = e^{i\alpha} I,$$

where α is a real number, and

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

is the 2×2 unit matrix.

(ii) *(Relative) phase shift:*

$$P(\alpha) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix},$$

where α is a real number. In particular, we have:

(a) *Phase gate:*

$$S = P(\pi/2) = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}.$$

(b) *$\pi/8$ gate:*

$$T = P(\pi/4) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}.$$

Example 2.2.2. *The Pauli matrices:*

$$\sigma_x = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \ \sigma_y = Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \ \sigma_z = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Obviously, we have $X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$. So, Pauli matrix X is actually the NOT gate.

Example 2.2.3. *Rotations about the $\hat{x}, \hat{y}, \hat{z}$ axes of the Bloch sphere:*

$$R_x(\theta) = \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2} X = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix},$$

$$R_y(\theta) = \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2} Y = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix},$$

$$R_z(\theta) = \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2} Z = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix},$$

where θ is a real number.

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

The gates in Example 2.2.3 have a nice geometric interpretation: a single qubit state can be represented by a vector in the so-called Bloch sphere. The effect of $R_x(\theta), R_y(\theta), R_z(\theta)$ on this state is to rotate it by angle θ about the x, y, z-axis, respectively, of the Bloch sphere; for details, we refer to [174], Sections 1.3.1 and 4.2. It can be shown that any one-qubit gate can be expressed as a circuit consisting of only rotations and global phase shift.

Exercise 2.2.2. *Prove that all the matrices in the previous three examples are unitary.*

2.2.3 CONTROLLED GATES

One-qubit gates are not enough for any useful quantum computation. In this subsection, we introduce an important class of multiple-qubit gates, namely the controlled gates.

The most frequently used among them is the CNOT operator C defined in Example 2.1.7. Here, we look at it in a different way. Let q_1, q_2 be qubit variables. Then $C[q_1, q_2]$ is a two-qubit gate with q_1 as the control qubit and q_2 as the target qubit. It acts as follows:

$$C[q_1, q_2]|i_1, i_2\rangle = |i_1, i_1 \oplus i_2\rangle$$

for $i_1, i_2 \in \{0, 1\}$, where $\oplus$ is addition modulo 2; that is, if q_1 is set to $|1\rangle$, then q_2 is flipped, otherwise q_2 is left unchanged. As a simple generalization of the CNOT gate, we have:

Example 2.2.4. *Let U be a 2×2 unitary matrix. Then the controlled-U is a two-qubit gate defined by*

$$C(U)[q_1, q_2]|i_1, i_2\rangle = |i_1\rangle U^{i_1} |i_2\rangle$$

for $i_1, i_2 \in \{0, 1\}$. Its matrix representation is

$$C(U) = \begin{pmatrix} I & 0 \\ 0 & U \end{pmatrix}$$

where I is the 2×2 unit matrix. Obviously, $C = C(X)$; that is, CNOT is the controlled-X with X being the Pauli matrix.

Exercise 2.2.3. *SWAP is a two-qubit gate defined by*

$$SWAP[q_1, q_2]]|i_1, i_2\rangle = |i_2, i_1\rangle$$

for $i_1, i_2 \in \{0, 1\}$. Intuitively, it swaps the states of two qubits. Show that SWAP can be implemented by three CNOT gates:

$$SWAP[q_1, q_2] = C[q_1, q_2]C[q_2, q_1]C[q_1, q_2].$$

Exercise 2.2.4. *Prove the following properties of controlled gates:*

(i) $C[p, q] = H[q]C(Z)[p, q]H[q]$.
(ii) $C(Z)[p, q] = C(Z)[q, p]$.
(iii) $H[p]H[q]C[p, q]H[p]H[q] = C[q, p]$.

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

(iv) $C(M(\alpha))[p,q] = P(\alpha)[p]$.
(v) $C[p,q]X[p]C[p,q] = X[p]X[q]$.
(vi) $C[p,q]Y[p]C[p,q] = Y[p]X[q]$.
(vii) $C[p,q]Z[p]C[p,q] = Z[p]$.
(viii) $C[p,q]X[q]C[p,q] = X[q]$.
(ix) $C[p,q]Y[q]C[p,q] = Z[p]Y[q]$.
(x) $C[p,q]Z[q]C[p,q] = Z[p]Z[q]$.
(xi) $C[p,q]T[p] = T[p]C[p,q]$.

All the controlled gates considered previously are two-qubit gates. Actually, we can define a much more general notion of controlled gate.

Definition 2.2.5. *Let $\overline{p} = p_1, \ldots, p_m$ and $\overline{q}$ be registers with $\overline{p} \cap \overline{q} = \emptyset$. If $G = U[\overline{q}]$ is a quantum gate, then the controlled circuit $C^{(\overline{p})}(U)$ with control qubits $\overline{p}$ and target qubits $\overline{q}$ is the unitary transformation in the state Hilbert space $\mathcal{H}_{\overline{p}\cup\overline{q}}$ defined by*

$$C^{(\overline{p})}(U)|\overline{t}\rangle|\psi\rangle = \begin{cases} |\overline{t}\rangle U|\psi\rangle & \text{if } t_1 = \ldots = t_m = 1, \\ |\overline{t}\rangle|\psi\rangle & \text{otherwise} \end{cases}$$

for any $\overline{t} = t_1 \ldots t_m \in \{0,1\}^m$ and $|\psi\rangle \in \mathcal{H}_{\overline{q}}$.

The following example presents a class of three-qubit controlled gates.

Example 2.2.5. *Let p_1, p_2, q be qubit variables and U a 2×2 unitary matrix. The controlled-controlled-U gate:*

$$C^2(U) = C^{(p_1,p_2)}(U)$$

is the unitary transformation in $\mathcal{H}_{p_1} \otimes \mathcal{H}_{p_2} \otimes \mathcal{H}_q$:

$$C^{(2)}(U)|t_1,t_2,\psi\rangle = \begin{cases} |t_1,t_2,\psi\rangle & \text{if } t_1 = 0 \text{ or } t_2 = 0, \\ |t_1,t_2\rangle U|\psi\rangle & \text{if } t_1 = t_2 = 1 \end{cases}$$

for $t_1, t_2 \in \{0,1\}$ and for any $|\psi\rangle \in \mathcal{H}_q$. In particular, the controlled-controlled-NOT is called the Toffoli gate.

The Toffoli gate is universal for classical reversible computation, and it is universal for quantum computation with a little extra help (in the sense defined in Subsection 2.2.5 following). It is also very useful in quantum error-correction.

Exercise 2.2.5. *Prove the following equalities that allow us to combine several controlled gates into a single one:*

(i) $C^{(\overline{p})}(C^{(\overline{q})}(U)) = C^{(\overline{p},\overline{q})}(U)$.
(ii) $C^{(\overline{p})}(U_1)C^{(\overline{p})}(U_2) = C^{(\overline{p})}(U_1U_2)$.

2.2.4 QUANTUM MULTIPLEXOR

Controlled gates can be further generalized to multiplexors. In this subsection, we introduce the notion of a quantum multiplexor and its matrix representation.

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

For classical computation, the simplest multiplexor is a *conditional* described by the "**if** ... **then** ... **else** ..." construction: perform the action specified in the "**then**" clause when the condition after "**if**" is true, and perform the action specified in the "**else**" clause when it is false. The implementation of conditionals may be done by first processing the "**then**" and "**else**" clauses in parallel and then multiplexing the outputs.

Quantum conditional is a quantum analog of classical conditional. It is formed by replacing the condition (Boolean expression) after "**if**" by a qubit; that is, replacing truth values *true* and *false* by the basis states $|1\rangle$ and $|0\rangle$, respectively, of a qubit.

Example 2.2.6. *Let p be a qubit variable and $\overline{q} = q_1, \ldots, q_n$ a quantum register, and let $C_0 = U_0[\overline{q}]$ and $C_1 = U_1[\overline{q}]$ be quantum gates. Then quantum conditional $C_0 \oplus C_1$ is a gate on $1+n$ qubits $p, \overline{q}$ with the first qubit p as the select qubit and the remaining n qubits $\overline{q}$ as the data qubits, defined by:*

$$(C_0 \oplus C_1)|i\rangle|\psi\rangle = |i\rangle U_i|\psi\rangle$$

for $i \in \{0,1\}$ and for any $|\psi\rangle \in \mathcal{H}_{\overline{q}}$. Equivalently, it is defined by the matrix:

$$C_0 \oplus C_1 = \begin{pmatrix} U_0 & 0 \\ 0 & U_1 \end{pmatrix}.$$

The controlled-gate defined in Example 2.2.4 is a special case of quantum conditional: $C(U) = I \oplus U$, where I is the unit matrix.

The essential difference between classical and quantum conditionals is that the select qubit can be not only in the basis states $|0\rangle$ and $|1\rangle$ but also in their superpositions:

$$(C_0 \oplus C_1)(\alpha_0|0\rangle|\psi_0\rangle + \alpha_1|1\rangle|\psi_1\rangle) = \alpha_0|0\rangle U_0|\psi_0\rangle + \alpha_1|1\rangle U_1|\psi_1\rangle$$

for any states $|\psi_0\rangle, |\psi_1\rangle \in \mathcal{H}_{\overline{q}}$ and for any complex numbers α_0, α_1 with $|\alpha_0|^2 + |\alpha_1|^2 = 1$.

A multiplexor is a multi-way generalization of conditional. Roughly speaking, a multiplexor is a switch that passes one of its data inputs through to the output, as a function of a set of select inputs. Similarly, a quantum multiplexor (QMUX for short) is a multi-way generalization of quantum conditional.

Definition 2.2.6. *Let $\overline{p} = p_1, \ldots, p_m$ and $\overline{q} = q_1, \ldots, q_n$ be quantum registers, and for each $x \in \{0,1\}^m$, let $C_x = U_x[\overline{q}]$ be a quantum gate. Then QMUX*

$$\bigoplus_x C_x$$

is a gate on $m+n$ qubits $\overline{p}, \overline{q}$, having the first m qubits $\overline{p}$ as the select qubits and the remaining n qubits $\overline{q}$ as the data qubits. It preserves any state of the select qubits, and performs a unitary transformation on the data qubits, which is chosen according to the state of the select qubits:

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

$$\left(\bigoplus_x C_x\right)|t\rangle|\psi\rangle = |t\rangle U_t|\psi\rangle$$

for any $t \in \{0,1\}^m$ *and* $|\psi\rangle \in \mathcal{H}_{\overline{q}}$.

The matrix representation of the QMUX is a diagonal:

$$\bigoplus_x C_x = \bigoplus_{x=0}^{2^m-1} U_x = \begin{pmatrix} U_0 & & & & \\ & U_1 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & U_{2^m-1} \end{pmatrix}.$$

Here, we identify an integer $0 \leq x < 2^m$ with its binary representation $x \in \{0,1\}^m$. The difference between classical multiplexor and QMUX also comes from the fact that the select qubits $\overline{p}$ can be in a superposition of basis states $|x\rangle$:

$$\left(\bigoplus_x C_x\right)\left(\sum_{x=0}^{2^m-1} \alpha_x|x\rangle|\psi_x\rangle\right) = \sum_{x=0}^{2^m-1} \alpha_x|x\rangle U_x|\psi_x\rangle$$

for any states $|\psi_x\rangle \in \mathcal{H}_{\overline{q}}$ $(0 \leq x < 2^m)$ and any complex numbers α_x with $\sum_x |\alpha_x|^2 = 1$. Obviously, the controlled gate introduced in Definition 2.2.5 is a special QMUX:

$$C^{(\overline{p})}(U) = I \oplus \ldots \oplus I \oplus U,$$

where the first $2^m - 1$ summands are the unit matrix of the same dimension as U.

Exercise 2.2.6. *Prove the multiplexor extension property:*

$$\left(\bigoplus_x C_x\right)\left(\bigoplus_x D_x\right) = \bigoplus_x (C_x D_x).$$

In the next section, we will see a simple application of QMUX in quantum walks. A close connection between QMUX and a quantum program construct – quantum case statement – will be revealed in Chapter 6. QMUXs have been successfully used for synthesis of quantum circuits (see [201]) and thus will be useful for compilation of quantum programs.

2.2.5 UNIVERSALITY OF GATES

We have already introduced several important classes of quantum gates in the last three subsections. A question naturally arises: are they sufficient for quantum computation? This section is devoted to answering this question.

To better understand this question, let us first consider the corresponding question in classical computation. For each $n \geq 0$, there are 2^{2^n} n-ary Boolean functions. Totally, we have infinitely many Boolean functions. However, there are some small

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

sets of logic gates that are universal: they can generate all Boolean functions; for example, {NOT, AND}, {NOT, OR}. The notion of universality can be easily generalized to the quantum case:

Definition 2.2.7. *A set Ω of unitary matrices is universal if all unitary matrices can be generated by it; that is, for any positive integer n, and for any $2^n \times 2^n$ unitary matrix U, there exists a circuit C with $qvar(C) = \{q_1, \ldots, q_n\}$ constructed from the gates defined by unitary matrices in Ω such that*

$$U[q_1, \ldots, q_n] = C$$

(equivalence of circuits introduced in Definition 2.2.3).

One of the simplest universal sets of quantum gates is presented in the following:

Theorem 2.2.1. *The CNOT gate together with all one-qubit gates is universal.*

The universal sets of classical gates mentioned previously are all finite. However, the universal set of quantum gates given in Theorem 2.2.1 is infinite. Indeed, the set of unitary operators form a continuum, which is uncountably infinite. So, it is impossible to exactly implement an arbitrary unitary operator by a finite set of quantum gates. This forces us to consider approximate universality rather than the exact universality introduced in Definition 2.2.7.

Definition 2.2.8. *A set Ω of unitary matrices is approximately universal if for any unitary operator U and for any $\epsilon > 0$, there is a circuit C with $qvar(C) = \{q_1, \ldots, q_n\}$ constructed from the gates defined by unitary matrices in Ω such that*

$$d(U[q_1, \ldots, q_n], C) < \epsilon,$$

where the distance d is defined by equation (2.2).

Two well-known approximately universal sets of gates are given in the following:

Theorem 2.2.2. *The following two sets of gates are approximately universal:*

(i) *Hadamard gate H, $\pi/8$ gate T and CNOT gate C;*
(ii) *Hadamard gate H, phase gate S, CNOT gate C and the Toffoli gate (see Example 2.2.5).*

The proofs of Theorems 2.2.1 and 2.2.2 are omitted here, but the reader can find them in book [174], Section 4.5.

2.2.6 MEASUREMENT IN CIRCUITS

The universality theorems presented in the last subsection indicate that any quantum computation can be carried out by a quantum circuit constructed from the basic quantum gates described in Subsections 2.2.2 and 2.2.3. But the output of a quantum circuit is usually a quantum state, which cannot be observed directly from the outside. In order to read out the outcome of computation, we have to perform a measurement at the end of the circuit. So, sometimes we need to consider a generalized notion of quantum circuit, namely circuit with quantum measurements.

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

As shown in Subsection 2.1.4, we only need to use projective measurements if it is allowed to introduce ancilla qubits. Furthermore, if the circuit contains n qubit variables, the measurement in the computational basis $\{|x\rangle : x \in \{0,1\}^n\}$ is sufficient because any orthonormal basis of these qubits can be obtained from the computational basis by a unitary transformation.

Actually, quantum measurements are not only used at the end of a computation. They are also often performed as an intermediate step of a computation and the measurement outcomes are used to conditionally control subsequent steps of the computation. But Nielsen and Chuang [174] explicitly pointed out:

- **Principle of deferred measurement:** Measurements can always be moved from an intermediate stage of a quantum circuit to the end of the circuit; if the measurement results are used at any stage of the circuit then the classically controlled operations can be replaced by conditional quantum operations.

Exercise 2.2.7. *Elaborate the principle of deferred measurement and prove it. This can be done in the following steps:*

(i) *We can formally define the notion of quantum circuit with measurements (mQC for short) by induction:*
- **(a)** *Each quantum gate is an mQC;*
- **(b)** *If $\overline{q}$ is a quantum register, $M = \{M_m\} = \{M_{m_1}, M_{m_2}, \ldots, M_{m_n}\}$ is a quantum measurement in $\mathcal{H}_{\overline{q}}$, and for each m, C_m is an mQC with $\overline{q} \cap qvar(C_m) = \emptyset$, then*

$$\begin{aligned} \textbf{if } (\square m \cdot M[\overline{q}] = m \to C_m) \textbf{ if } \equiv\ & \textbf{fi}\ M[\overline{q}] = m_1 \to C_{m_1} \\ & \square \quad\ m_2 \to C_{m_2} \\ & \quad\ \ldots\ldots \\ & \square \quad\ m_n \to C_{m_n} \\ & \textbf{fi} \end{aligned} \tag{2.13}$$

is a mQC too;
- **(c)** *If C_1 and C_2 are mQCs, so is $C_1 C_2$.*

Intuitively, equation (2.13) means that we perform measurement M on $\overline{q}$, and then the subsequent computation is selected based on the measurement outcome: if the outcome is m, then the corresponding circuit C_m follows.

(ii) *Generalize the notion of equivalence between quantum circuits (Definition 2.2.3) to the case of mQCs.*

(iii) *Show that for any mQC C, there is a quantum circuit C' (without measurements) and a quantum measurement $M[\overline{q}]$ such that $C = C'M[\overline{q}]$ (equivalence).*

If we remove the condition $\overline{q} \cap qvar(C_m) = \emptyset$ from clause (ii), then the post-measurement states of measured qubits can be used in the subsequent computation. Is the principle of deferred measurement still true for this case?

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

2.3 QUANTUM ALGORITHMS

Quantum circuits together with measurements described in the last section give us a complete (but low-level) model of quantum computation. Since the early 1990s, various quantum algorithms that can offer speed-up over their classical counterparts have been discovered. Partially due to historical reasons and partially due to lack of convenient quantum programming languages at that time, all of them were described in the model of quantum circuits.

In this section, we present several interesting quantum algorithms. Our aim is to provide examples of the quantum program constructs introduced in the subsequent chapters but not to provide a thorough discussion of quantum algorithms. If the reader would like to enter the core of this book as quickly as possible, she/he can skip this section for the first reading, and directly move to Chapter 3. Of course, she/he will need to come back to this point if she/he wishes to understand the examples in the subsequent chapters where the quantum algorithms presented in this section are programmed.

2.3.1 QUANTUM PARALLELISM AND INTERFERENCE

Let us start from two basic techniques for designing quantum algorithms – quantum parallelism and interference. They are two key ingredients that enable a quantum computer to outperform its classical counterpart.

Quantum Parallelism:

Quantum parallelism can be clearly illustrated through a simple example. Consider an n-ary Boolean function:

$$f : \{0,1\}^n \to \{0,1\}.$$

The task is to evaluate $f(x)$ for different values $x \in \{0,1\}^n$ simultaneously. Classical parallelism for this task can be roughly imagined as follows: *multiple* circuits each for computing the same function f are built, and they are executed simultaneously for different inputs x. In contrast, we only need to build a *single* quantum circuit that implements the unitary transformation:

$$U_f : |x, y\rangle \to |x, y \oplus f(x)\rangle \tag{2.14}$$

for any $x \in \{0,1\}^n$ and $y \in \{0,1\}$. Obviously, unitary operator U_f is generated from the Boolean function f. This circuit consists of $n+1$ qubits, the first n qubits form the "data" register, and the last is the " target" register. It can be proved that given a classical circuit for computing f we can construct a quantum circuit with comparable complexity that implements U_f.

Exercise 2.3.1. *Show that U_f is a multiplexor (see Definition 2.2.6):*

$$U_f = \bigoplus_x U_{f,x},$$

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

where the first n qubits are used as the select qubits, and for each $x \in \{0,1\}^n$, $U_{f,x}$ *is a unitary operator on the last qubit defined by*

$$U_{f,x}|y\rangle = |y \oplus f(x)\rangle$$

for $y \in \{0,1\}$*; that is,* $U_{f,x}$ *is* I *(the identity operator) if* $f(x) = 0$ *and it is* X *(the NOT gate) if* $f(x) = 1$.

The following procedure shows how quantum parallelism can accomplish the task of evaluating $f(x)$ simultaneously for all inputs $x \in \{0,1\}^n$:

- An equal superposition of 2^n basis states of the data register is produced very efficiently by only n Hadamard gates:

$$|0\rangle^{\otimes n} \overset{H^{\otimes n}}{\rightarrow} |\psi\rangle \overset{\triangle}{=} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle,$$

  where $|0\rangle^{\otimes n} = |0\rangle \otimes \ldots \otimes |0\rangle$ (the tensor product of n $|0\rangle$'s), and $H^{\otimes n} = H \otimes \ldots \otimes H$ (the tensor product of n H's).
- Applying unitary transformation U_f to the data register in state $|\psi\rangle$ and the target register in state $|0\rangle$ yields:

$$|\psi\rangle|0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x,0\rangle \overset{U_f}{\rightarrow} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x,f(x)\rangle. \tag{2.15}$$

It should be noticed that the unitary transformation U_f was executed *only once* in this equation, but the different terms in the right-hand side of the equation contain information about $f(x)$ for all $x \in \{0,1\}^n$. In a sense, $f(x)$ was evaluated for 2^n different values of x *simultaneously*.

However, quantum parallelism is not enough for a quantum computer to outperform its classical counterpart. Indeed, to extract information from the state in the right-hand side of equation (2.15), a measurement must be performed on it; for example, if we perform the measurement in the computational basis $\{|x\rangle : x \in \{0,1\}^n\}$ on the data register, then it would give $f(x)$ at the target register only for a single value of x (with probability $1/2^n$), and we cannot obtain $f(x)$ for all $x \in \{0,1\}^n$ at the same time. Thus, a quantum computer has no advantage over a classical computer at all if such a naïve way of extracting information is used.

Quantum Interference:

In order to be really useful, quantum parallelism has to be combined with another feature of quantum systems – quantum interference. For example, let us consider a superposition

$$\sum_x \alpha_x |x, f(x)\rangle$$

of which the right-hand side of equation (2.15) is a special case. As said before, if we directly measure the data register in the computational basis, we can only get

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

local information about $f(x)$ for a single value of x. But if we first perform a unitary operator U on the data register, then the original superposition is transformed to

$$U\left(\sum_x \alpha_x |x, f(x)\rangle\right) = \sum_x \alpha_x \left(\sum_{x'} U_{x'x} |x', f(x)\rangle\right)$$
$$= \sum_{x'} \left[|x'\rangle \otimes \left(\sum_x \alpha_x U_{x'x} |f(x)\rangle\right)\right],$$

where $U_{x'x} = \langle x'|U|x\rangle$, and now the measurement in the computational basis will give certain *global* information about $f(x)$ for all $x \in \{0,1\}^n$. This global information resides in

$$\sum_x \alpha_x U_{x'x} |f(x)\rangle$$

for some single value x'. In a sense, the unitary transformation U was able to merge information about $f(x)$ for different values of x. It is worth noting that the measurement in a basis after a unitary transformation is essentially the measurement in a different basis. So, an appropriate choice of a basis in which a measurement is performed is crucial in order to extract the desired global information.

2.3.2 DEUTSCH-JOZSA ALGORITHM

It is still not convincing from the general discussion in the previous subsection that quantum parallelism and interference can actually help us to solve some interesting computational problems. However, the power of combining quantum parallelism and interference can be clearly seen in the Deutsch-Jozsa algorithm that solves the following:

- *Deutsch Problem:* Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$, known to be either constant, or balanced – $f(x)$ equals 0 for exactly half of all the possible x, and 1 for the other half. Determine whether it is constant or balanced.

The algorithm is described in Figure 2.1. It should be emphasized that in this algorithm, the unitary operator U_f determined by function f according to equation (2.14) is supplied as an oracle.

To understand this quantum algorithm, we need to carefully look at several key ideas in its design:

- In step 2, the target register (the last qubit) is cleverly initialized in state $|-\rangle = H|1\rangle$ rather than in state $|0\rangle$ as in equation (2.15). This special initialization is often referred to as the *phase kickback trick* since

$$U_f|x,-\rangle = |x\rangle \otimes (-1)^{f(x)}|-\rangle$$
$$= (-1)^{f(x)}|x,-\rangle.$$

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

- **Inputs:** A quantum oracle that implements the unitary operator U_f defined by equation (2.14).
- **Outputs:** 0 if and only if f is constant.
- **Runtime:** One application of U_f. Always succeeds.
- **Procedure:**

1. $|0\rangle^{\otimes n}|1\rangle$
2. $\overset{H^{\otimes(n+1)}}{\rightarrow} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle|-\rangle$
3. $\overset{U_f}{\rightarrow} \frac{1}{\sqrt{2^n}} \sum_x (-1)^{f(x)}|x\rangle|-\rangle$
4. $\overset{H^{\otimes n} \text{ on the first } n \text{ qubits}}{\rightarrow} \sum_z \frac{\sum_x (-1)^{x\cdot z+f(x)}}{2^n}|z\rangle|-\rangle$
5. $\overset{\text{measure on the first } n \text{ qubits in the computational basis}}{\rightarrow} z$

FIGURE 2.1

Deutsch-Jozsa algorithm.

Here, only the phase of the target register is changed from 1 to $(-1)^{f(x)}$, which can be moved to the front of the data register.

- Quantum parallelism happens in step 3 when applying the oracle U_f.
- Quantum interference is used in step 4: n Hadamard gates acting on the data register (the first n qubits) yields

$$\begin{aligned} & H^{\otimes n}\left(\frac{1}{\sqrt{2^n}}\sum_x |x\rangle \otimes (-1)^{f(x)}|-\rangle\right) \\ &= \frac{1}{\sqrt{2^n}}\sum_x \left(H^{\otimes n}|x\rangle \otimes (-1)^{f(x)}|-\rangle\right) \\ &= \frac{1}{2^n}\sum_x \left(\sum_z (-1)^{x\cdot z}|z\rangle \otimes (-1)^{f(x)}|-\rangle\right) \\ &= \frac{1}{2^n}\sum_z \left[\left(\sum_x (-1)^{x\cdot z+f(x)}\right)|z\rangle \otimes |-\rangle\right]. \end{aligned} \tag{2.16}$$

- In step 5, we measure the data register in the computational basis $\{|z\rangle : z \in \{0,1\}^n\}$. The probability that we get outcome $z = 0$ (i.e., $|z\rangle = |0\rangle^{\otimes n}$) is

$$\frac{1}{2^n}\left|\sum_x (-1)^{f(x)}\right|^2 = \begin{cases} 1 & \text{if } f \text{ is constant,} \\ 0 & \text{if } f \text{ is balanced.} \end{cases}$$

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

It is interesting to note that the positive and negative contributions to the amplitude for $|0\rangle^{\otimes n}$ cancel when f is balanced.

Exercise 2.3.2. *Prove the equality used in equation (2.16):*

$$H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{z\in\{0,1\}^n} (-1)^{x\cdot z}|z\rangle$$

for any $x \in \{0,1\}^n$*, where*

$$x \cdot z = \sum_{i=1}^{n} x_i z_i$$

if $x = x_1, \ldots, x_n$ *and* $z = z_1, \ldots, z_n$.

Finally, let us briefly compare the query complexities of the Deutsch problem in classical computing and the Deutsch-Jozsa algorithm. A deterministic classical algorithm should repeatedly select a value $x \in \{0,1\}^n$ and calculate $f(x)$ until it can determine with certainty whether f is constant or balanced. So, a classical algorithm requires $2^{n-1}+1$ evaluations of f. In contrast, U_f is executed only once in step 3 of the Deutsch-Jozsa algorithm.

2.3.3 GROVER SEARCH ALGORITHM

The Deutsch-Jozsa algorithm properly illustrates several key ideas for designing quantum algorithms, but the problem solved by it is somehow artificial. In this subsection, we introduce a quantum algorithm that is very useful for a wide range of practical applications, namely the Grover algorithm that solves the following:

- *Search Problem*: The task is to search through a database consisting of N elements, indexed by numbers $0, 1, \ldots, N-1$. For convenience, we assume that $N = 2^n$ so that the index can be stored in n bits. We also assume that the problem has exactly M solutions with $1 \leq M \leq N/2$.

As in the Deutsch-Jozsa algorithm, we are supplied with a quantum oracle – a black box with the ability to recognize a solution of the search problem. Formally, let function $f : \{0, 1, \ldots, N-1\} \to \{0,1\}$ be defined as follows:

$$f(x) = \begin{cases} 1 & \text{if } x \text{ is a solution,} \\ 0 & \text{otherwise.} \end{cases}$$

We write

$$\mathcal{H}_N = \mathcal{H}_2^{\otimes n} = span\{|0\rangle, |1\rangle, \ldots, |N-1\rangle\}$$

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

with $\mathcal{H}_2$ being the state Hilbert space of a qubit. Then the oracle can be thought of as the unitary operator $O = U_f$ in $\mathcal{H}_N \otimes \mathcal{H}_2$ defined by

$$O|x,q\rangle = U_f|x,q\rangle = |x\rangle|q \oplus f(x)\rangle \tag{2.17}$$

for $x \in \{0,1,\ldots,N-1\}$ and $q \in \{0,1\}$, where $|x\rangle$ is the index register, and $|q\rangle$ is the oracle qubit which is flipped if x is a solution, and is unchanged otherwise. In particular, the oracle has the phase kickback property:

$$|x,-\rangle \xrightarrow{O} (-1)^{f(x)}|x,-\rangle.$$

Thus, if the oracle qubit is initially in state $|-\rangle$, then it remains $|-\rangle$ throughout the search algorithm and can be omitted. So, we can simply write:

$$|x\rangle \xrightarrow{O} (-1)^{f(x)}|x\rangle. \tag{2.18}$$

Grover Rotation:

One key subroutine of the Grover algorithm is called the Grover rotation. It consists of four steps, as described in Figure 2.2.

- **Procedure:**

1. Apply the oracle O;
2. Apply the Hadamard transform $H^{\otimes n}$;
3. Perform a conditional phase shift :
   $$|0\rangle \to |0\rangle,$$
   $$|x\rangle \to -|x\rangle \text{ for all } x \neq 0;$$
4. Apply the Hadamard transform $H^{\otimes n}$.

FIGURE 2.2

Grover rotation.

Let us see what the Grover rotation actually does. We write G for the unitary transformation defined by the procedure in Figure 2.2; i.e., the composition of the operators in steps 1-4. It should be pointed out that the oracle O used in step 1 is thought of as a unitary operator in the space $\mathcal{H}_N$ (rather than $\mathcal{H}_N \otimes \mathcal{H}_2$) defined by equation (2.18). The conditional phase shift in step 3 is defined in the basis $\{|0\rangle, |1\rangle, \ldots, |N-1\rangle\}$ of the space $\mathcal{H}_N$. The following lemma presents the unitary operator of the quantum circuit that implements the Grover rotation.

Lemma 2.3.1. *$G = (2|\psi\rangle\langle\psi| - I)O$, where*

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$$

is the equal superposition in $\mathcal{H}_N$.

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

Exercise 2.3.3. *Prove Lemma 2.3.1.*

It is not easy to imagine from just the previous description that the operator G represents a rotation. A geometric visualization can help us to understand the Grover rotation better. Let us introduce two vectors in the space $\mathcal{H}_N$:

$$|\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_{x \text{ not solution}} |x\rangle,$$

$$|\beta\rangle = \frac{1}{\sqrt{M}} \sum_{x \text{ solution}} |x\rangle.$$

It is clear that the vectors $|\alpha\rangle$ and $|\beta\rangle$ are orthogonal. If we define angle θ by

$$\cos\frac{\theta}{2} = \sqrt{\frac{N-M}{N}} \ (0 \le \frac{\theta}{2} \le \frac{\pi}{2}),$$

then the equal superposition in Lemma 2.3.1 can be expressed as follows:

$$|\psi\rangle = \cos\frac{\theta}{2}|\alpha\rangle + \sin\frac{\theta}{2}|\beta\rangle.$$

Furthermore, we have:

Lemma 2.3.2. $G(\cos\delta|\alpha\rangle + \sin\delta|\beta\rangle) = \cos(\theta+\delta)|\alpha\rangle + \sin(\theta+\delta)|\beta\rangle$.

Intuitively, the Grover operator G is a rotation for angle θ in the two-dimensional space spanned by $|\alpha\rangle$ and $|\beta\rangle$. For any real number δ, the vector $\cos\delta|\alpha\rangle + \sin\delta|\beta\rangle$ can be represented by a point $(\cos\delta, \sin\delta)$. Thus, Lemma 2.3.2 indicates that the action of G is depicted by the mapping:

$$(\cos\delta, \sin\delta) \xrightarrow{G} (\cos(\theta+\delta), \sin(\theta+\delta)).$$

Exercise 2.3.4. *Prove Lemma 2.3.2.*

Grover Algorithm:

Using the Grover rotation as a subroutine, the quantum search algorithm can be described as shown in Figure 2.3.

It should be noted that k in Figure 2.3 is a constant integer; the value of k will be suitably fixed in the next paragraph.

Performance Analysis:

It can be shown that the search problem requires approximately N/M operations by a classical computer. Let us see how many iterations of G are needed in step 3 of the Grover algorithm. Note that in step 2 the index register (i.e., the first n qubits) is prepared in the state

$$|\psi\rangle = \sqrt{\frac{N-M}{N}}|\alpha\rangle + \sqrt{\frac{M}{N}}|\beta\rangle.$$

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

- **Inputs:** A quantum oracle O defined by equation (2.17).
- **Outputs:** A solution x.
- **Runtime:** $O(\sqrt{N})$ operations. Succeeds with probability $\Theta(1)$.
- **Procedure:**

1. $|0\rangle^{\otimes n}|1\rangle$
2. $\overset{H^{\otimes(n+1)}}{\rightarrow} \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle|-\rangle = \left(\cos\frac{\theta}{2}|\alpha\rangle + \sin\frac{\theta}{2}|\beta\rangle\right)|-\rangle$
3. $\overset{G^k \text{ on the first } n \text{ qubits}}{\rightarrow} \left[\cos\left(\frac{2k+1}{2}\theta\right)|\alpha\rangle + \sin\left(\frac{2k+1}{2}\theta\right)|\beta\rangle\right]|-\rangle$
4. $\overset{\text{measure the first } n \text{ qubits in the computational basis}}{\rightarrow} |x\rangle$

FIGURE 2.3

Grover search algorithm.

So, rotating through $\arccos\sqrt{\frac{M}{N}}$ radians takes the index register from $|\psi\rangle$ to $|\beta\rangle$. It is asserted by Lemma 2.3.2 that the Grover operator G is a rotation for angle θ. Let k be the integer closest to the real number

$$\frac{\arccos\sqrt{\frac{M}{N}}}{\theta}.$$

Then we have:

$$k \leq \left\lceil \frac{\arccos\sqrt{\frac{M}{N}}}{\theta} \right\rceil \leq \left\lceil \frac{\pi}{2\theta} \right\rceil$$

because $\arccos\sqrt{\frac{M}{N}} \leq \frac{\pi}{2}$. Consequently, k is a positive integer in the interval $\left[\frac{\pi}{2\theta} - 1, \frac{\pi}{2\theta}\right]$. By the assumption $M \leq \frac{N}{2}$, we have

$$\frac{\theta}{2} \geq \sin\frac{\theta}{2} = \sqrt{\frac{M}{N}}$$

and $k \leq \left\lceil \frac{\pi}{4}\sqrt{\frac{N}{M}} \right\rceil$, i.e., $k = O(\sqrt{N})$. On the other hand, by the definition of k we obtain:

$$\left| k - \frac{\arccos\sqrt{\frac{M}{N}}}{\theta} \right| \leq \frac{1}{2}.$$

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

It follows that

$$\arccos\sqrt{\frac{M}{N}} \leq \frac{2k+1}{2}\theta \leq \theta + \arccos\sqrt{\frac{M}{N}}.$$

Since $\cos\frac{\theta}{2} = \sqrt{\frac{N-M}{N}}$, we have $\arccos\sqrt{\frac{M}{N}} = \frac{\pi}{2} - \frac{\theta}{2}$ and

$$\frac{\pi}{2} - \frac{\theta}{2} \leq \frac{2k+1}{2}\theta \leq \frac{\pi}{2} + \frac{\theta}{2}.$$

Thus, since $M \leq \frac{N}{2}$, it holds that the success probability

$$\Pr(\text{ success}) = \sin^2\left(\frac{2k+1}{2}\theta\right) \geq \cos^2\frac{\theta}{2} = \frac{N-M}{N} \geq \frac{1}{2},$$

i.e., $\Pr(\text{ success}) = \Theta(1)$. In particular, if $M \ll N$, then the success probability is very high.

The previous derivation can be summarized as follows: The Grover algorithm can find a solution x with success probability $O(1)$ within $k = O(\sqrt{N})$ steps.

2.3.4 QUANTUM WALKS

In the previous subsections, we saw how the power of quantum parallelism and interference can be exploited to design the Deutsch-Jozsa algorithm and the Grover search algorithm. We now turn to consider a class of quantum algorithms for which the design idea looks very different from that used in the Deutsch-Jozsa algorithm and the Grover algorithm. This class of algorithms was developed based on the notion of quantum walk, which is the quantum counterpart of random walk.

One-Dimensional Quantum Walk:

The simplest random walk is the one-dimensional walk where a particle moves on a discrete line whose nodes are denoted by integers $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$. At each step, the particle moves one position left or right, depending on the flip of a "coin." A quantum variant of the one-dimensional random walk is the Hadamard walk defined in the following:

Example 2.3.1. *The state Hilbert space of the Hadamard walk is $\mathcal{H}_d \otimes \mathcal{H}_p$, where:*

- *$\mathcal{H}_d = span\{|L\rangle, |R\rangle\}$ is a two-dimensional Hilbert space, called the direction space, and $|L\rangle, |R\rangle$ are used to indicate the directions Left and Right, respectively;*
- *$\mathcal{H}_p = span\{|n\rangle : n \in \mathbb{Z}\}$ is an infinite-dimensional Hilbert space, and $|n\rangle$ indicates the position marked by integer n,*

and spanX for a nonempty set X is defined according to equation (2.1). One step of the Hadamard walk is represented by the unitary operator

$$W = T(H \otimes I_{\mathcal{H}_p}),$$

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

where the translation T is a unitary operator in $\mathcal{H}_d \otimes \mathcal{H}_p$ defined by

$$T|L,n\rangle = |L,n-1\rangle, \quad T|R,n\rangle = |R,n+1\rangle$$

for every $n \in \mathbb{Z}$, H is the Hadamard transformation in the direction space $\mathcal{H}_d$, and $I_{\mathcal{H}_p}$ is the identity operator in the position space $\mathcal{H}_p$. The Hadamard walk is then described by repeated applications of operator W.

Exercise 2.3.5. *We define the left and right translation operators T_L and T_R in the position space $\mathcal{H}_p$ by*

$$T_L|n\rangle = |n-1\rangle, \quad T_R|n\rangle = |n+1\rangle$$

for each $n \in \mathbb{Z}$. Then the translation operator T is the quantum conditional $T_L \oplus T_R$ with the direction variable d as the select qubit (see Example 2.2.6).

Although the Hadamard walk was defined by mimicking the one-dimensional random walk, some of their behaviors are very different:

- The translation operator T can be explained as follows: if the direction system is in state $|L\rangle$, then the walker moves from position n to $n-1$, and if the direction is in $|R\rangle$, then the walker moves from position n to $n+1$. This looks very similar to a random walk, but in a quantum walk, the direction can be in a superposition of $|L\rangle$ and $|R\rangle$, and intuitively the walker can move to the left and to the right simultaneously.
- In a random walk, we only need to specify the statistical behavior of the "coin"; for example, flipping a fair " coin" gives heads and tails with equal probability $\frac{1}{2}$. In a quantum walk, however, we have to explicitly define the dynamics of the " coin" underlying its statistical behavior; for example, the Hadamard transformation H can be seen as a quantum realization of the fair "coin"; but so does the following 2×2 unitary matrix (and many others):

$$C = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix}.$$

- Quantum interference may happen in a quantum walk; for example, let the Hadamard walk start in state $|L\rangle|0\rangle$. Then we have:

$$\begin{aligned} |L\rangle|0\rangle &\xrightarrow{H} \frac{1}{\sqrt{2}}\big(|L\rangle + |R\rangle\big)|0\rangle \\ &\xrightarrow{T} \frac{1}{\sqrt{2}}\big(|L\rangle|-1\rangle + |R\rangle|1\rangle\big) \\ &\xrightarrow{H} \frac{1}{2}\big[\big(|L\rangle + |R\rangle\big)|-1\rangle + \big(|L\rangle - |R\rangle\big)|1\rangle\big] \\ &\xrightarrow{T} \frac{1}{2}\big(|L\rangle|-2\rangle + |R\rangle|0\rangle + |L\rangle|0\rangle - |R\rangle|2\rangle\big) \\ &\xrightarrow{H} \frac{1}{2\sqrt{2}}\big[\big(|L\rangle + |R\rangle\big)|-2\rangle + \big(|L\rangle - |R\rangle\big)|0\rangle \\ &\qquad + \big(|L\rangle + |R\rangle\big)|0\rangle - \big(|L\rangle - |R\rangle\big)|2\rangle\big] \end{aligned} \tag{2.19}$$

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

Here, $-|R\rangle|0\rangle$ and $|R\rangle|0\rangle$ are out of phase and thus cancel one another.

Quantum Walk on a Graph:

Random walks on graphs are a class of random walks widely used in the design and analysis of algorithms. Let $G = (V, E)$ be an n-regular directed graph; that is, a graph where each vertex has n neighbors. Then we can label each edge with a number between 1 and n such that for each $1 \leq i \leq n$, the directed edges labeled i form a permutation. In this way, for each vertex v, the ith neighbor v_i of v is defined to be the vertex linked from v by an edge labeled i. A random walk on G is defined as follows: the vertices v's of G are used to represent the states of the walk, and for each state v the walk goes from v to its every neighbor with a certain probability. Such a random walk also has a quantum counterpart, which is carefully described in the following:

Example 2.3.2. *The state Hilbert space of a quantum walk on an n-regular graph $G = (V, E)$ is $\mathcal{H}_d \otimes \mathcal{H}_p$, where:*

- *$\mathcal{H}_d = span\{|i\rangle\}_{i=1}^{n}$ is an n-dimensional Hilbert space. We introduce an auxiliary quantum system, called the direction "coin," with the state space $\mathcal{H}_d$. For each $1 \leq i \leq n$, the state $|i\rangle$ is used to denote the ith direction. The space $\mathcal{H}_d$ is referred to as the "coin space";*
- *$\mathcal{H}_p = span\{|v\rangle\}_{v \in V}$ is the position Hilbert space. For each vertex v of the graph, there is a basis state $|v\rangle$ in $\mathcal{H}_p$.*

The shift S is an operator in $\mathcal{H}_d \otimes \mathcal{H}_p$ defined as follows:

$$S|i, v\rangle = |i\rangle|v_i\rangle$$

for any $1 \leq i \leq n$ and $v \in V$, where v_i is the ith neighbor of v. Intuitively, for each i, if the "coin" is in state $|i\rangle$, then the walker moves in the ith direction. Of course, the "coin" can be in a superposition of states $|i\rangle$ ($1 \leq i \leq n$) and the walker moves to all the directions simultaneously.

If we further choose a unitary operator C in the "coin" space $\mathcal{H}_d$, called the "coin-tossing operator," then a single step of a coined quantum walk on graph G can be modelled by the unitary operator:

$$W = S(C \otimes I_{\mathcal{H}_p}) \tag{2.20}$$

where $I_{\mathcal{H}_p}$ is the identity operator in the position space $\mathcal{H}_p$. For example, a fair "coin" can be implemented by choosing the discrete Fourier transform:

$$FT = \frac{1}{\sqrt{d}} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{d-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(d-1)} \\ & \dots & \dots & \dots & \\ 1 & \omega^{d-1} & \omega^{(d-1)2} & \dots & \omega^{(d-1)(d-1)} \end{pmatrix} \tag{2.21}$$

as the "coin-tossing operator," where $\omega = \exp(2\pi i/d)$. The operator FT maps each direction into a superposition of directions such that after measurement each of them

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

is obtained with the equal probability $\frac{1}{d}$. *The quantum walk is then an iteration of the single-step walk operator* W.

Exercise 2.3.6. *For each* $1 \leq i \leq n$, *we can define a shift operator* S_i *in the position space* $\mathcal{H}_V$:

$$S_i|v\rangle = |v_i\rangle$$

for any $v \in V$, *where* v_i *stands for the ith neighbor of* v. *If we slightly generalize the notion of quantum multiplexor (QMUX) by allowing the select variable being any quantum variable but not only qubits, then the shift operator* S *in Example 2.3.2 is the QMUX* $\bigoplus_i S_i$ *with the direction* d *as the select variable.*

It has been observed that sometimes quantum effect (e.g., interference) in a quantum walk can offer a significant speed-up; for example, it helps a quantum walk to hit a vertex from another much faster than a random walk.

2.3.5 QUANTUM-WALK SEARCH ALGORITHM

Is it possible to harness the quantum speed-up pointed out at the end of the last subsection to design quantum algorithms that outperform their classical counterparts? In this subsection, we present such an algorithm for solving the search problem considered in Subsection 2.3.3.

Assume that the database consists of $N = 2^n$ items, each of which is encoded as an n-bit string $x = x_1 \ldots x_n \in \{0,1\}^n$. It was assumed in Subsection 2.3.3 that there are M solutions. Here, we only consider the special case of $M = 1$. So, the task is to find the single target item (solution) x^*. The search algorithm in this subsection is based upon a quantum walk over the n-cube – the hypercube of dimension n. The n-cube is a graph with $N = 2^n$ nodes, each of which corresponds to an item x. Two nodes x and y are connected by an edge if they have only a one-bit difference:

$$x_d \neq y_d \text{ for some } d, \text{ and } x_i = y_i \text{ for all } i \neq d;$$

that is, x and y differ by only a single-bit flip. Thus, each of the 2^n nodes of the n-cube has degree n – it is connected to n other nodes.

As a special case of Example 2.3.2, the quantum walk over the n-cube is described as follows:

- The state Hilbert space is $\mathcal{H}_d \otimes \mathcal{H}_p$, where $\mathcal{H}_d = span\{|1\rangle, \ldots, |n\rangle\}$,

$$\mathcal{H}_p = \mathcal{H}_2^{\otimes n} = span\{|x\rangle : x \in \{0,1\}^n\},$$

  and $\mathcal{H}_2$ is the state space of a qubit.
- The shift operator S maps $|d, x\rangle$ to $|d, x \oplus e_d\rangle$ (the dth bit of x is flipped), where $e_d = 0 \ldots 010 \ldots 0$ (the dth bit is 1 and all others are 0) is the dth basis vector of the n-cube. Formally,

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

$$S = \sum_{d=1}^{n} \sum_{x \in \{0,1\}^n} |d, x \oplus e_d\rangle\langle d, x|$$

where $\oplus$ is component-wise addition modulo 2.

- The "coin tossing" operator C is chosen to be the Grover rotation without the oracle (see Lemma 2.3.1):

$$C = 2|\psi_d\rangle\langle\psi_d| - I$$

where I is the identity operator in $\mathcal{H}_d$ and $|\psi_d\rangle$ is the equal superposition over all n directions:

$$|\psi_d\rangle = \frac{1}{\sqrt{n}} \sum_{d=1}^{n} |d\rangle.$$

As in the Grover algorithm, we are supplied with an oracle that can mark the target item x^*. Suppose that this oracle is implemented via a perturbation of C:

$$D = C \otimes \sum_{x \neq x^*} |x\rangle\langle x| + C' \otimes |x^*\rangle\langle x^*| \tag{2.22}$$

where C' is a unitary operator in $\mathcal{H}_d$. Intuitively, the oracle applies the original "coin tossing" operator C to the direction system whenever the position corresponds to a nontarget item, but marks the target item x^* by applying a special "coin" action C'.

Now the search algorithm works as follows:

- Initialize the quantum computer to the equal superposition over both all directions and all positions: $|\psi_0\rangle = |\psi_d\rangle \otimes |\psi_p\rangle$, where

$$|\psi_p\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle.$$

- Apply the perturbed single-step walk operator

$$W' = SD = W - S\left[(C - C') \otimes |x^*\rangle\langle x^*|\right]$$

$t = \left\lceil \frac{\pi}{2}\sqrt{N} \right\rceil$ times, where W is the single-step walk operator defined by equation (2.20).

- Measure the state of the quantum computer in the $|d, x\rangle$ basis.

There is a remarkable difference between the "coin tossing" operator D used in this algorithm and the original "coin tossing" operator C (more precisely, $C \otimes I$) in Example 2.3.2: the operator C acts only in the direction space and thus is position-independent. However, D is obtained by modifying $C \otimes I$ with C' marking the target item x^*, and it is obvious from equation (2.22) that D is position-dependent.

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

For the case of $C' = -I$, it was proved that the algorithm finds the target item with probability $\frac{1}{2} - O(\frac{1}{n})$, and thus the target item can be found with an arbitrarily small probability of error by repeating the algorithm a constant number of times. The performance analysis of this algorithm is involved and not included here, but the reader can find it in the original paper [203].

The reader is invited to carefully compare this search algorithm based on quantum walk with the Grover search algorithms introduced in Subsection 2.3.3.

2.3.6 QUANTUM FOURIER TRANSFORM

Another important class of quantum algorithms is based on the quantum Fourier transform. Recall that the discrete Fourier transform takes as input a vector of complex numbers $x_0, \ldots, x_{N-1}$, and it outputs a vector of complex numbers $y_0, \ldots, y_{N-1}$:

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi ijk/N} x_j \tag{2.23}$$

for each $0 \leq j < N$. The quantum Fourier transform is a quantum counterpart of the discrete Fourier transform.

Definition 2.3.1. *The quantum Fourier transform on an orthonormal basis* $|0\rangle, \ldots, |N-1\rangle$ *is defined by*

$$FT : |j\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi ijk/N} |k\rangle.$$

More generally, the quantum Fourier transform on a general state in the N-dimensional Hilbert space is given as follows:

$$FT : \sum_{j=0}^{N-1} x_j |j\rangle \to \sum_{k=0}^{N-1} y_k |k\rangle,$$

where the amplitudes $y_0, \ldots, y_{N-1}$ are obtained by the discrete Fourier (2.23) transform on amplitudes $x_0, \ldots, x_{N-1}$. The matrix representation of the quantum Fourier transform was given in equation (2.21).

Proposition 2.3.1. *The quantum Fourier transform FT is unitary.*

Exercise 2.3.7. *Prove Proposition 2.3.1.*

The Circuit of Quantum Fourier Transform:

An implementation of the quantum Fourier transform FT by one-qubit and two-qubit gates is presented in the following proposition and its proof.

Proposition 2.3.2. *Let $N = 2^n$. Then the quantum Fourier transform can be implemented by a quantum circuit consisting of n Hadamard gates and*

$$\frac{n(n-1)}{2} + 3\lfloor \frac{n}{2} \rfloor$$

controlled gates.

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

Proof. We prove this proposition by explicitly constructing a quantum circuit that fulfils the stated conditions. We use the binary representation:

- $j_1 j_2 \ldots j_n$ denotes

$$j = j_1 2^{n-1} + j_2 2^{n-2} + \ldots + j_n 2^0;$$

- $0.j_k j_{k+1} \ldots j_n$ denotes

$$j_k/2 + j_{k+1}/2^2 + \ldots + j_n/2^{n-k+1}$$

for any $k \geq 1$.

Then the proposition can be proved in three steps:

(i) Using the notation introduced in Section 2.2, we design the circuit:

$$\begin{aligned} D \equiv\ & H[q_1]C(R_2)[q_2,q_1]\ldots C(R_n)[q_n,q_1]H[q_2]C(R_2)[q_3,q_2] \\ & \ldots C(R_{n-1})[q_n,q_2]\ldots H[q_{n-1}]C(R_2)[q_n,q_{n-1}]H[q_n] \end{aligned} \tag{2.24}$$

where R_k is the phase shift (see Example 2.2.1):

$$R_k = P\left(2\pi/2^k\right) = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix}$$

for $k = 2, \ldots, n$. If we input $|j\rangle = |j_1 \ldots j_n\rangle$ into the circuit (2.24), then the output is :

$$\frac{1}{\sqrt{2^n}}(|0\rangle + e^{2\pi i 0.j_1\ldots j_n}|1\rangle)\ldots(|0\rangle + e^{2\pi i 0.j_n}|1\rangle) \tag{2.25}$$

by a routine calculation.

(ii) We observe that whenever $N = 2^n$, the quantum Fourier transform can be rewritten as follows:

$$\begin{aligned} |j\rangle &\to \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi ijk/2^n}|k\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} e^{2\pi ij(k_1\cdot 2^{n-1}+\ldots+k_n\cdot 2^0)/2^n}|k_1\ldots k_n\rangle \\ &= \frac{1}{\sqrt{2^n}} \left(\sum_{k_1=0}^{1} e^{2\pi ijk_1/2^1}|k_1\rangle\right)\ldots\left(\sum_{k_n=0}^{1} e^{2\pi ijk_n/2^n}|k_n\rangle\right) \\ &= \frac{1}{\sqrt{2^n}} \left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right)\ldots\left(|0\rangle + e^{2\pi i 0.j_1\ldots j_n}|1\rangle\right). \end{aligned} \tag{2.26}$$

(iii) Finally, by comparing equations (2.26) and (2.25), we see that adding $\lfloor \frac{n}{2} \rfloor$ swap gates at the end of the circuit (2.24) will reverse the order of the qubits,

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

and thus yield the quantum Fourier transform. It is known that each swap gate can be accomplished by using 3 CNOT gates (see Exercise 2.2.3). □

2.3.7 PHASE ESTIMATION

Now we show how the quantum Fourier transform defined in the last subsection can be used in an algorithm for phase estimation. This quantum algorithm solves the following problem:

- *Phase Estimation*: A unitary operator U has an eigenvector $|u\rangle$ with eigenvalue $e^{2\pi i\varphi}$, where the value of φ is unknown. The goal is to estimate the phase φ.

The phase estimation algorithm is described in Figure 2.4. It uses two registers:

- The first consists of t qubits $q_1, \ldots, q_t$, all of which are initialized in state $|0\rangle$;
- The second is the system p which U applies to, initialized in state $|u\rangle$.

Using the notation introduced in Section 2.2, the circuit for this algorithm can be written as follows:

$$D \equiv E \cdot FT^{\dagger}[q_1, \ldots, q_t] \tag{2.27}$$

where:

$$\begin{aligned} E \equiv\ & H[q_1] \ldots H[q_{t-2}]H[q_{t-1}]H[q_t] \\ & C(U^{2^0})[q_t, p]C(U^{2^1})[q_{t-1}, p]C(U^{2^2})[q_{t-2}, p] \ldots C(U^{2^{t-1}})[q_1, p] \end{aligned}$$

$C(\cdot)$ is the controlled gate (see Definition 2.2.5), and $FT^{\dagger}$ is the inverse quantum Fourier transform FT and can be obtained by reversing the circuit of FT given in the proof of Proposition 2.3.2.

Obviously, circuit (2.27) consists of $O(t^2)$ Hadamard and controlled gates together with one call to oracle U^{2^j} for $j = 0, 1, \ldots, t-1$. We further observe that

$$\begin{aligned} E|0\rangle_{q_1} \ldots |0\rangle_{q_{t-2}}|0\rangle_{q_{t-1}}|0\rangle_{q_t}|u\rangle_p &= \frac{1}{\sqrt{2^t}}\left(|0\rangle + e^{2\pi i\varphi\cdot 2^{t-1}}|1\rangle\right) \\ &\ldots \left(|0\rangle + e^{2\pi i\varphi\cdot 2^2}|1\rangle\right)\left(|0\rangle + e^{2\pi i\varphi\cdot 2^1}|1\rangle\right)\left(|0\rangle + e^{2\pi i\varphi\cdot 2^0}|1\rangle\right)|u\rangle \\ &= \frac{1}{\sqrt{2^t}}\left(\sum_{k=0}^{2^t-1} e^{2\pi i\varphi k}|k\rangle\right)|u\rangle. \end{aligned} \tag{2.28}$$

A Special Case:

To understand why the algorithm works, let us first consider a special case where φ can be exactly expressed in t bits:

$$\varphi = 0.\varphi_1\varphi_2\varphi_3 \ldots \varphi_t.$$

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

- **Inputs:**

  (i) An oracle which performs controlled-U^{2^j} operators for $j = 0, 1, \ldots, t-1$;
  (ii) t qubits initialized to $|0\rangle$;
  (iii) An eigenvector $|u\rangle$ of U with eigenvalue $e^{2\pi i\varphi}$,

  where

$$t = n + \left\lceil \log(2 + \frac{1}{2\epsilon}) \right\rceil.$$

- **Outputs:** An n-bit approximation $\tilde{\varphi} = m$ to φ.
- **Runtime:** $O(t^2)$ operations and one call to each oracle. Success with probability at least $1 - \epsilon$.
- **Procedure:**

$$1.\ |0\rangle^{\otimes t}|u\rangle \overset{H^{\otimes t} \text{ on the first } t \text{ qubits}}{\rightarrow} \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle|u\rangle$$

$$2.\ \overset{\text{oracles}}{\rightarrow} \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle U^j|u\rangle = \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} e^{2\pi ij\varphi}|j\rangle|u\rangle$$

$$3.\ \overset{FT^\dagger}{\rightarrow} \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} e^{2\pi ij\varphi} \left(\frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} e^{-2\pi ijk/2^t}|k\rangle \right) |u\rangle$$

$$= \sum_{k=0}^{2^t-1} \alpha_k |k\rangle|u\rangle$$

$$4.\ \overset{\text{measure the first } t \text{ qubits}}{\rightarrow} |m\rangle|u\rangle,$$

  where

$$\alpha_k = \frac{1}{2^t} \sum_{j=0}^{2^t-1} e^{2\pi ij(\varphi - k/2^t)} = \frac{1}{2^t} \left[\frac{1 - e^{2\pi i(2^t\varphi - k)}}{1 - e^{2\pi i(\varphi - k/2^t)}} \right].$$

FIGURE 2.4

Phase estimation.

Then equation (2.28) can be rewritten as:

$$E|0\rangle \ldots |0\rangle|0\rangle|0\rangle|u\rangle = \frac{1}{\sqrt{2^t}} \left(|0\rangle + e^{2\pi i 0.\varphi_t}|1\rangle\right) \ldots \left(|0\rangle + e^{2\pi i 0.\varphi_3 \ldots \varphi_t}|1\rangle\right) \left(|0\rangle + e^{2\pi i 0.\varphi_2\varphi_3 \ldots \varphi_t}|1\rangle\right) \left(|0\rangle + e^{2\pi i \varphi_1\varphi_2\varphi_3 \ldots \varphi_t}|1\rangle\right) |u\rangle. \tag{2.29}$$

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

Furthermore, by equations (2.27) and (2.26) we obtain:

$$\begin{aligned} C|0\rangle\ldots|0\rangle|0\rangle|0\rangle|u\rangle &= FT^{\dagger}\left(E|0\rangle\ldots|0\rangle|0\rangle|0\rangle\right)|u\rangle \\ &= |\varphi_1\varphi_2\varphi_3\ldots\varphi_t\rangle|u\rangle. \end{aligned}$$

Performance Analysis:

The previous discussion about a special case should give the reader a hint why the algorithm is correct. Now we are ready to consider the general case. Let $0 \leq b < 2^t$ be such that $b/2^t = 0.b_1\ldots b_t$ is the best t bit approximation to φ which is less than φ; i.e.,

$$b/2^t \leq \varphi < b/2^t + 1/2^t.$$

We write $\delta = \varphi - b/2^t$ for the difference. It is clear that $0 \leq \delta < 1/2^t$. Note that

$$|\alpha_k| \leq \frac{1}{2^{t-1}|1 - e^{2\pi i(\varphi-k)/2^t}|}$$

because $|1 - e^{i\theta}| \leq 2$ for all θ. Put $\beta_l = \alpha_{(b+l \mod 2^t)}$ for any $-2^{t-1} < l \leq 2^{t-1}$. Then

$$|\beta_l| \leq \frac{1}{2^{t-1}|1 - e^{2\pi i(\delta-l/2^t)}|} \leq \frac{1}{2|l - 2^t\delta|}$$

because

(i) $|1 - e^{i\theta}| \geq \frac{2|\theta|}{\pi}$ if $-\pi \leq \theta \leq \pi$; and
(ii) $-\frac{1}{2} \leq \delta - l/2^t \leq \frac{1}{2}$.

Suppose the outcome of the final measurement is m. Then for a positive integer d, we have:

$$\begin{aligned} P(|m-b| > d) &= \sum_{m:|m-b|>d} |\alpha_m|^2 \\ &= \sum_{-2^{t-1}<l\leq-(d+1)} |\beta_l|^2 + \sum_{d+1\leq l\leq 2^{t-1}} |\beta_l|^2 \\ &\leq \frac{1}{4}\left[\sum_{l=-2^{t-1}+1}^{-(d+1)} \frac{1}{(l-2^t\delta)^2} + \sum_{l=d+1}^{2^{t-1}} \frac{1}{(l-2^t\delta)^2}\right] \\ &\leq \frac{1}{4}\left[\sum_{l=-2^{t-1}+1}^{-(d+1)} \frac{1}{l^2} + \sum_{l=d+1}^{2^{t-1}} \frac{1}{(l-1)^2}\right] \quad (\text{note that } 0 \leq 2^t\delta < 1) \\ &\leq \frac{1}{2}\sum_{l=d}^{2^{t-1}} \frac{1}{l^2} \\ &\leq \frac{1}{2}\int_{d-1}^{2^{t-1}} \frac{dl}{l^2} \leq \frac{1}{2(d-1)}. \end{aligned}$$

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

If we wish to approximate φ to a 2^{-n} accuracy and the success probability is at least $1-\epsilon$, then we only need to choose $d = 2^{t-n} - 1$ and require $\frac{1}{2(d-1)} \leq \epsilon$. This leads to

$$t \geq T \triangleq n + \left\lceil \log(\frac{1}{2\epsilon} + 2) \right\rceil$$

and we can make use of $t = T$ qubits in the phase estimation algorithm.

Combining the preceding derivation with equation (2.27) and Proposition 2.3.2 gives us the conclusion: the algorithm presented in Figure 2.4 can compute the n-bit approximation of phase φ with at least success probability $1 - \epsilon$ within $O(t^2)$ steps, using

$$n + \left\lceil \log(\frac{1}{2\epsilon} + 2) \right\rceil$$

qubits.

The phase estimation algorithm is a key procedure in a class of important quantum algorithms, including the famous Shor algorithm for factoring [204] and the Harrow-Hassidim-Lloyd algorithm for systems of linear equations [112]. A detailed presentation of these two algorithms is out of the scope of this book.

2.4 BIBLIOGRAPHIC REMARKS

- *Quantum Mechanics*: The material on quantum mechanics presented in Section 2.1 is standard and can be found in any (advanced) textbook of quantum mechanics.
- *Quantum Circuits*: Part of Section 2.2 is based on [34] and Chapter 4 of book [174]. The quantum multiplexor in Subsection 2.2.4 was introduced by Shende et al. [201]. The notations for quantum gates and circuits as well as the notion of a quantum circuit with measurements in Exercise 2.2.7 come from [226]. Section 2.2 is merely an introduction to the basics of quantum circuits. Quantum circuits have been developed into a large research area since [34]. In particular, in recent years, research on quantum circuits, including synthesis (decomposition of large unitary matrices) and optimization of quantum circuits, became very active with applications to compilation of quantum programming languages; see Section 8.2 for further discussion. It is worth mentioning that synthesis and optimization of quantum circuits are much harder than the corresponding problems for classical circuits.
- *Quantum Algorithms*: The presentation of Subsections 2.3.1 to 2.3.3, 2.3.6 and 2.3.7 largely follows Sections 1.4, 5.1, 5.2 and 6.1 of [174]. The one-dimensional quantum walk and the quantum walk on a graph in Subsection 2.3.4 were defined in [9] and [19], respectively. The algorithm given in Subsection 2.3.5 was proposed by Shenvi et al. [203].

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.

Quantum algorithms have been one of the most active research areas in quantum computing since Shor's factoring algorithm and Grover search were discovered. For the three major quantum algorithms in earlier times, namely the Shor algorithm, Grover algorithm and quantum simulation [154], and their variants, [174] is still one of the best expositions. Shor [205] proposed two explanations for why so few classes of quantum algorithms have been found and pointed out several lines of research that might lead to the discovery of new quantum algorithms. A large number of papers on quantum walks and algorithms based on them have been published in the last decade; see [18,192,214] for a thorough survey. A recent breakthrough in quantum algorithms is the Harrow-Hassidim-Lloyd algorithm for systems of linear equations [112]. It further led to active research on quantum machine learning algorithms [156,157,184] in the last few years; see [2] for some interesting discussions about this line of research.

Copyright © 2016. Elsevier Science & Technology. All rights reserved.

Ying, Mingsheng. Foundations of Quantum Programming, Elsevier Science & Technology, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/hud/detail.action?docID=4460281.
Created from hud on 2021-02-20 07:22:40.