# Mathematical Foundation of Quantum Computing

## 量子計算之數學基礎

鄭經斅

# Contents


**1 Logic Circuits** **1**

1.1 Classical Logic Gates . . . . . 1

1.1.1 The NOT gate . . . . . 2

1.1.2 The AND gate and the OR gate . . . . . 2

1.1.3 The NAND gate and the NOR gate . . . . . 3

1.1.4 The XOR gate and the XNOR gate . . . . . 4

1.1.5 The TOFFOLI gate . . . . . 5

1.2 Universal Gates . . . . . 5

1.3 How A Classical Computer Adds Numbers . . . . . 11

1.3.1 Binary numbers . . . . . 11

1.3.2 Adder using logic circuits . . . . . 13

1.4 Classical Circuits . . . . . 15

**2 Quantum Computing** **16**

2.1 Quantum Mechanics . . . . . 17

2.1.1 Schrödinger equation . . . . . 17

2.1.2 Superposition . . . . . 18

2.1.3 Measurement . . . . . 19

2.1.4 Unitary evolution . . . . . 20

2.2 Qubits and Quantum Gates . . . . . 21

2.2.1 Quantum bits . . . . . 22

2.2.2 Quantum gates . . . . . 22

2.3 Quantum Registers . . . . . 27

2.3.1 Tensor product of quantum registers - preview . . . . . 29

2.3.2 Entanglement . . . . . 31

2.4 Quantum Circuits . . . 32
2.4.1 Quantum Teleportation . . . 35
2.5 Universality of Various Sets of Elementary Gates . . . 37
2.6 Quantum Parallelism . . . 38
2.7 The Early Algorithms . . . 39
2.7.1 Deutsch-Jozsa . . . 40
2.7.2 Bernstein-Vazirani . . . 42

**3 Mathematical Backgrounds 43**
3.1 Vector Spaces and Linear Maps . . . 43
3.1.1 Vector Spaces . . . 43
3.1.2 Linear maps and their matrix representation . . . 45
3.1.3 Algebraic dual spaces . . . 47
3.2 Direct Sum of Vector Spaces and Multi-Linear Maps . . . 49
3.2.1 Direct sum of vector spaces . . . 49
3.2.2 Multi-linear maps . . . 50
3.3 Inner Product Spaces and Hilbert Spaces . . . 51
3.4 Dual Spaces and Adjoint Operators . . . 56
3.5 Unitary Operators and Unitary Matrices . . . 61
3.5.1 Unitary operators . . . 61
3.5.2 Unitary matrices . . . 63
3.6 Quantum Mechanics . . . 65
3.7 Tensor Product of Vector Spaces . . . 69
3.7.1 Tensor product . . . 69
3.7.2 Correspondence between tensor product and quantum circuits . . . 79
3.7.3 More examples . . . 81
3.8 Unitary Decomposition . . . 100
3.8.1 1-qubit gate decomposition . . . 100
3.8.2 Singular value decomposition . . . 102
3.8.3 The CS decomposition . . . 105
3.8.4 Decomposition of arbitrary quantum gates . . . 108
3.9 Implementation of Multi-Controlled Rotation Gates . . . 113

**4 Simon's Algorithm** **123**
4.1 Simon's Problem . . . 123
4.2 The Quantum Algorithm . . . 124
4.3 Classical Algorithms for Simon's Problem . . . 126
4.3.1 Upper bound . . . 126
4.3.2 Lower bound . . . 126

**5 The Fourier Transform** **129**
5.1 The Classical Discrete Fourier Transform . . . 129
5.2 The Fast Fourier Transform . . . 130
5.3 Application: Multiplying Two Polynomials . . . 131
5.4 The Quantum Fourier Transform . . . 133
5.5 Application: phase estimation . . . 136

**6 Shor's Factoring Algorithm** **138**
6.1 RSA Encryption . . . 138
6.1.1 Mathematical foundation . . . 138
6.1.2 Encryption based on factoring large numbers . . . 144
6.2 Reduction from Factoring to Period-finding . . . 147
6.3 Shor's Period-finding Algorithm . . . 149
6.4 Continued fractions . . . 152
6.5 Efficiency of Shor's Algorithm . . . 154
6.5.1 Shor's period-finding algorithm . . . 154
6.5.2 The period of $f(a) = x^a \bmod N$ is most likely even . . . 159

**7 Grover's Search Algorithm** **173**
7.1 The Problem . . . 173
7.2 Grover's Algorithm . . . 173
7.3 Amplitude Amplification . . . 178

**8 The HHL Algorithm** **179**
8.1 The Linear System Problem . . . 179
8.2 The Basic HHL Algorithm for Linear Systems . . . 180
8.2.1 Detailed quantum algorithm . . . 182

# Chapter 1

# Logic Circuits

## 1.1 Classical Logic Gates

In a classical computer the processor essentially performs nothing more than a sequence of transformations of a **classical state** into another one. In mathematical terminology, a classical processor performs a sequence of evaluation of maps of the form

$$\begin{array}{rccc} f: & \{0,1\}^n & \to & \{0,1\}^m \\ & x & \mapsto & f(x) \end{array} \tag{1.1}$$

This is what we will refer to as the classical computational process, which is realized with a concatenation of classical gates and circuits.

**Definition 1.1.** A classical logical gate, also called a Boolean function, is a map

$$\begin{array}{rccc} g: & \{0,1\}^n & \to & \{0,1\} \\ & (x_1, \cdots, x_n) & \mapsto & g(x_1, \cdots, x_n) \end{array} .$$

We define an extended classical logical gate $g$ as a map

$$\begin{array}{rccc} g: & \{0,1\}^n & \to & \{0,1\}^m \\ & (x_1, \cdots, x_n) & \mapsto & \big(g_1(x_1, \cdots, x_n), \cdots, g_m(x_1, \cdots, x_n)\big) \end{array} ,$$

where each $g_j$ is a classical logic gate. A classical gate $g$ is called reversible if it is a bijection and thus invertible.

**Example 1.2.** The addition $\oplus$ on $\mathbb{Z}_2$ defined by

$$a \oplus b \equiv a + b - 2ab \qquad \forall\, a, b \in \{0,1\} \tag{1.2}$$

can be treated as a classical logic gate from $\{0,1\}^2$ to $\{0,1\}$ given by

$$(a, b) \mapsto a \oplus b \qquad \forall\, a, b \in \{0,1\}\,.$$

In the following sub-sections, we introduce some important classical logic gates.

### 1.1.1 The NOT gate

The **NOT** gate, also called an inverter, is a logic gate (from $\{0,1\}$ to $\{0,1\}$) which implements logical negation. It behaves according to the truth table below:

| **INPUT** | **OUTPUT** |
|---|---|
| 0 | 1 |
| 1 | 0 |

The analytical form of the **NOT** gate is given by $\mathbf{NOT}(a) = 1 - a$ for $a \in \{0,1\}$. We note that the **NOT** gate is reversible, and the inverse of the **NOT** gate is itself.

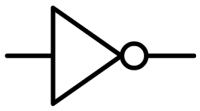


Figure 1.1: Traditional **NOT** Gate (Inverter) symbol

### 1.1.2 The AND gate and the OR gate

The **AND** gate (及閘) is a basic digital logic gate (from $\{0,1\}^2$ to $\{0,1\}$) that implements logical conjunction, and the **OR** gate (或閘) is a digital logic gate that implements logical disjunction. They behave according to the truth tables below:

| **INPUT** | | **OUTPUT** |
|---|---|---|
| A | B | A **AND** B |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| **INPUT** | | **OUTPUT** |
|---|---|---|
| A | B | A **OR** B |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Analytically, the function of **AND** finds the product of binary digits, while the function of **OR** finds the maximum between two binary digits; that is,

$$\mathbf{AND}(a,b) = a \cdot b \quad \text{and} \quad \mathbf{OR}(a,b) = \max\{a,b\} \qquad \forall\, a,b \in \{0,1\}\,.$$

The logic gate symbols for the **AND** and **OR** gates are

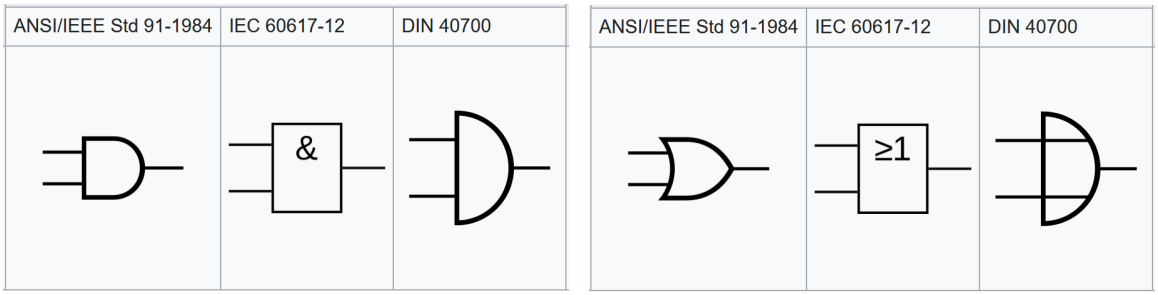



Figure 1.2: Logic gate symbols for **AND** (left) and **OR** (right) gates

We note that the **AND** gate and the **OR** gate are not reversible.

### 1.1.3 The NAND gate and the NOR gate

The **NAND** gate (NOT-AND，反及閘) is a logic gate whose output is complement to that of an **AND** gate. In other words, the **NAND** gate produces an output which is false only if all its inputs are true. On the other hand, the **NOR** gate (NOT-OR，反或閘) is a logic gate whose output is complement to that of an **OR** gate. They behave according to the truth tables below:

| **INPUT** | | **OUTPUT** |
|---|---|---|
| A | B | A **NAND** B |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

| **INPUT** | | **OUTPUT** |
|---|---|---|
| A | B | A **NOR** B |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

The logic gate symbols for the **NAND** and **NOR** gates are

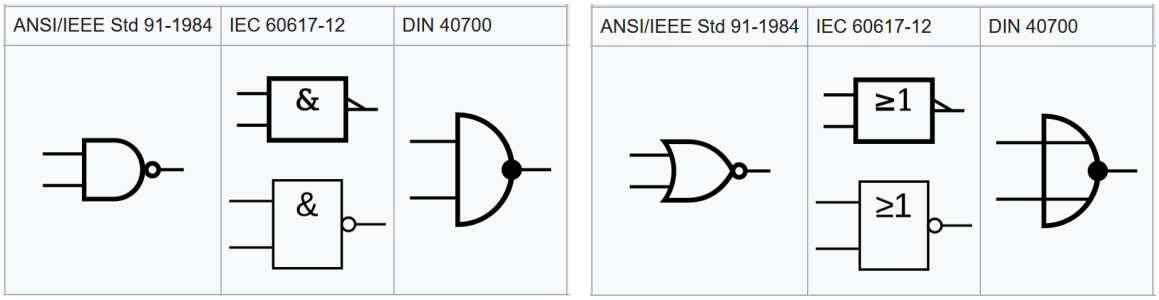



Figure 1.3: Logic gate symbols for **NAND** (left) and **NOR** (right) gates

We also note that the **NAND** gate and the **NOR** gates are not reversible.

### 1.1.4 The XOR gate and the XNOR gate

The **XOR** gate (sometimes **EOR**, or **EXOR** and pronounced as Exclusive OR，互斥或閘) is a digital logic gate (from $\{0,1\}^2$ to $\{0,1\}$) that gives a true (1 or HIGH) output when the number of true inputs is odd. If both inputs are false (0 or LOW) or both are true, a false output results. **XOR** represents the inequality function; that is, the output is true if the inputs are not alike otherwise the output is false. **XOR** can also be viewed as addition modulo 2. As a result, **XOR** gates are used to implement binary addition in computers.

The **XNOR** gate (sometimes **ENOR**, **EXNOR** or **NXOR** and pronounced as Exclusive NOR，反互斥或閘) is a digital logic gate whose function is the logical complement of the Exclusive OR (**XOR**) gate. The **XOR** and **XNOR** gates behave according to the truth table below:

| **INPUT** | | **OUTPUT** |
|---|---|---|
| A | B | A **XOR** B |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

| **INPUT** | | **OUTPUT** |
|---|---|---|
| A | B | A **XNOR** B |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

The analytic form of the **XOR** and the **XNOR** gate, respectively, are

$$\mathbf{XOR}(a,b) = a \oplus b = a + b - 2ab \qquad \forall\, a,b \in \{0,1\}\,,$$
$$\mathbf{XNOR}(a,b) = 1 \oplus a \oplus b = 1 + 2ab - a - b \qquad \forall\, a,b \in \{0,1\}\,.$$

The logic gate symbols for the **XOR** and **XNOR** gates are

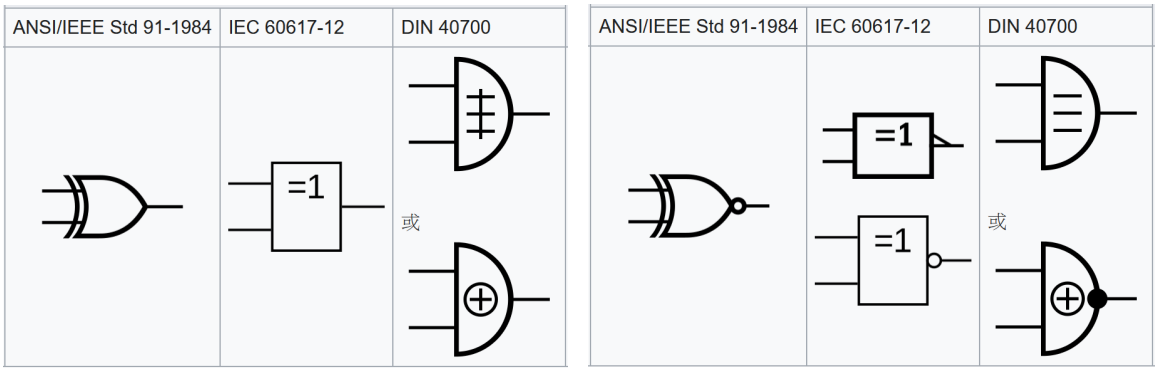



Figure 1.4: Logic gate symbols for **XOR** and **XNOR** gates

Similar to the **AND**, **OR**, **NAND** and **NOR** gates, the **XOR** and **XNOR** gates are not reversible.

### 1.1.5 The Toffoli gate

The Toffoli gate, also called **CCNOT** (pronounced controlled-controlled-not) gate, is a digital logic gate which behaves according to the truth table below:

| INPUT | | | OUTPUT | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 |

The analytic form of the Toffoli gate is

$$\textbf{TOF}(a,b,c) = (a,b,ab \oplus c) \qquad \forall\, a,b,c \in \{0,1\}$$

and the symbol for the Toffoli gate is

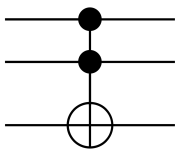


Figure 1.5: Circuit representation of Toffoli gate

The $n$-bit Toffoli gate is a generalization of Toffoli gate. It takes $n$ bits $x_1$, $x_2$, $\cdots$, $x_n$ as inputs and outputs $n$ bits: the first $n-1$ output bits are just $x_1, \cdots, x_{n-1}$, and the last output bit is $x_1 x_2 \cdots x_{n-1} \oplus x_n$.

## 1.2 Universal Gates

Universal gates can be "combined" to perform "all" Boolean functions. Before talking about the precise definition of the universality of classical logic gates, we need to introduce two basic operations that can be easily performed by classical computers (via storing/swapping data in the memory, maybe?).

**Definition 1.3.** 1. Let $n, \ell \in \mathbb{N}$ and $\ell \leqslant n$. For a pairwise distinct set $\{j_1, \cdots, j_\ell\} \subseteq \{1, \cdots, n\}$, the **restriction and/or re-ordering** operation $r^{(n)}_{j_1 j_2 \cdots j_\ell}$ is a classical gate

from $\{0,1\}^n$ to $\{0,1\}^\ell$ given by

$$r^{(n)}_{j_1 j_2 \cdots j_\ell}(x_1, \cdots, x_n) = (x_{j_1}, x_{j_2}, \cdots, x_{j_\ell})\,.$$

2. Let $n, \ell \in \mathbb{N}$. For a given point $(y_1, y_2, \cdots, y_\ell) \in \{0,1\}^\ell$ and a pairwise distinct set $\{j_1, \cdots, j_\ell\} \subseteq \{1, 2, \cdots, n+\ell\}$, the **padding** operation $p^{(n)}_{y_1,\cdots,y_\ell\,;j_1,\cdots,j_\ell}$ is a classical gate from $\{0,1\}^n$ to $\{0,1\}^{n+\ell}$ given by

$$p^{(n)}_{y_1,\cdots,y_\ell\,;j_1,\cdots,j_\ell}(x_1, \cdots, x_n) = (z_1, \cdots, z_{n+\ell})\,,$$

where

$$z_k = \begin{cases} x_{k-\#\{r\in\{j_1,\cdots,j_\ell\}\,|\,r<k\}} & \text{if } k \notin \{j_1, j_2, \cdots, j_\ell\}\,, \\ y_{j_r} & \text{if } k \in \{j_1, j_2, \cdots, j_\ell\} \text{ and } k = j_r\,. \end{cases}$$

In other words, the padding operation $p^{(n)}_{y_1,\cdots,y_\ell;j_1,\cdots,j_\ell}$ inserts pre-determined bit values $y_1, \cdots, y_\ell \in \{0,1\}$ at pre-determined slots $j_1, \cdots, j_\ell \in \{1, \cdots, n+\ell\}$.

**Definition 1.4.** Let $\{g_1, g_2, \cdots, g_k\}$ be a collection of classical gates. The collection of all gates that can be constructed from $g_1$, $g_2$, $\cdots$, $g_k$, denoted by $\mathscr{F}[g_1, \cdots, g_k]$, is the set satisfying the following construction rules:

1. For any $1 \leqslant j \leqslant k$, $g_j \in \mathscr{F}[g_1, \cdots, g_k]$.

2. For any $y_1, \cdots, y_\ell \in \{0,1\}$ and pairwise distinct $j_1, \cdots, j_\ell \in \{1, \cdots, n+\ell\}$, where $\ell, n \in \mathbb{N}$,

$$p^{(n)}_{y_1,\cdots,y_\ell\,;j_1,\cdots,j_\ell} \in \mathscr{F}[g_1, \cdots, g_k]\,.$$

3. For any pairwise distinct $j_1, \cdots, j_\ell \in \{1, \cdots, \ell\}$, where $\ell, n \in \mathbb{N}$ and $\ell \leqslant n$,

$$r^{(n)}_{j_1 j_2 \cdots j_\ell} \in \mathscr{F}[g_1, \cdots, g_k]\,.$$

4. Compositions of elements of $\mathscr{F}[g_1, \cdots, g_k]$ belong to $\mathscr{F}[g_1, \cdots, g_k]$; that is, for any $h_1 : \{0,1\}^n \to \{0,1\}^m$ and $h_2 : \{0,1\}^\ell \to \{0,1\}^n$, we have

$$h_1, h_2 \in \mathscr{F}[g_1, ..., g_k] \quad \Rightarrow \quad h_1 \circ h_2 \in \mathscr{F}[g_1, ..., g_k]\,.$$

5. Cartesian products of elements of $\mathscr{F}[g_1, \cdots, g_k]$ belong to $\mathscr{F}[g_1, \cdots, g_k]$; that is, for any $h = (h_1, \cdots, h_m) : \{0,1\}^n \to \{0,1\}^m$ and $k = (k_1, \cdots, k_q) : \{0,1\}^p \to \{0,1\}^q$, we have

$$h, k \in \mathscr{F}[g_1, \cdots, g_k] \quad \Rightarrow \quad h \times k \in \mathscr{F}[g_1, \cdots, g_k]\,,$$

where $h \times k : \{0,1\}^{n+p} \to \{0,1\}^{m+q}$ is the Cartesian product of $h$ and $k$ defined by

$$\begin{aligned}&(h \times k)(x_1, \cdots, x_{n+p}) \\ &\quad = \big(h_1(x_1, \cdots, x_n), \cdots, h_m(x_1, \cdots, x_n), k_1(x_{n+1}, \cdots, x_{n+p}), \cdots, k_q(x_{n+1}, \cdots, x_{n+p})\big)\,.\end{aligned}$$

**Example 1.5.** Let **ID** be classical gates given by

$$\mathbf{ID}(a) = a \qquad \forall\, a \in \{0,1\}\,.$$

Then $\mathbf{ID}(a) = \mathbf{AND}(a,1) = (\mathbf{AND} \circ p^{(1)}_{1;2})(a)$ which implies that

$$\mathbf{ID} = \mathbf{AND} \circ p^{(1)}_{1;2}\,.$$

Therefore, $\mathbf{ID} \in \mathscr{F}[\mathbf{AND}]$.

**Example 1.6.** For $n \in \mathbb{N}$, let $\mathbf{COPY}^{(n)}$ be the classical gate given by

$$\mathbf{COPY}^{(n)}(\boldsymbol{a}) = (\boldsymbol{a}, \boldsymbol{a}) \qquad \forall\, \boldsymbol{a} \in \{0,1\}^n, .$$

Using the identity

$$(\mathbf{COPY}^{(1)} \times \cdots \times \mathbf{COPY}^{(1)})(a_1, \cdots, a_n) = (a_1, a_1, a_2, a_2, \cdots, a_n, a_n) \quad \forall\, a_1, \cdots, a_n \in \{0,1\}\,,$$

we find that

$$\mathbf{COPY}^{(n)} = r^{(2n)}_{1,3,\cdots,2n-1,2,4,\cdots,2n} \circ \underbrace{\mathbf{COPY}^{(1)} \times \cdots \times \mathbf{COPY}^{(1)}}_{n \text{ copies of } \mathbf{COPY}^{(1)}}\,.$$

Therefore, $\mathbf{COPY}^{(n)} \in \mathscr{F}[\mathbf{COPY}^{(1)}]$ for all $n \in \mathbb{N}$.

**Example 1.7.** Let $\mathbf{COPY}^{(1)}$ be classical gates given in the previous example. Then for $a, b, c \in \{0,1\}$,

$$\begin{aligned}&\big((\mathbf{ID} \times \mathbf{ID} \times \mathbf{XOR}) \circ (\mathbf{ID} \times \mathbf{ID} \times \mathbf{AND} \times \mathbf{ID}) \circ r^{(5)}_{1,3,2,4,5} \circ (\mathbf{COPY}^{(1)} \times \mathbf{COPY}^{(1)} \times \mathbf{ID})\big)(a,b,c) \\ &\quad = \big((\mathbf{ID} \times \mathbf{ID} \times \mathbf{XOR}) \circ (\mathbf{ID} \times \mathbf{ID} \times \mathbf{AND} \times \mathbf{ID}) \circ r^{(5)}_{1,3,2,4,5}\big)(a,a,b,b,c) \\ &\quad = \big((\mathbf{ID} \times \mathbf{ID} \times \mathbf{XOR}) \circ (\mathbf{ID} \times \mathbf{ID} \times \mathbf{AND} \times \mathbf{ID})\big)(a,b,a,b,c) \\ &\quad = (\mathbf{ID} \times \mathbf{ID} \times \mathbf{XOR})(a,b,\mathbf{AND}(a,b),c) = (a,b,ab \oplus c) = \mathbf{TOF}(a,b,c)\,.\end{aligned}$$

Therefore, $\mathbf{TOF} \in \mathscr{F}\big[\mathbf{ID}, \mathbf{XOR}, \mathbf{AND}, \mathbf{COPY}^{(1)}\big]$. Moreover, Example 1.5 further shows that $\mathbf{TOF} \in \mathscr{F}\big[\mathbf{XOR}, \mathbf{AND}, \mathbf{COPY}^{(1)}\big]$

**Example 1.8.** In classical complexity theory, a Boolean circuit is a finite directed acyclic graph with **AND**, **OR**, and **NOT** gates. It has $n$ input nodes, which contain the $n$ input bits ($n \geqslant 0$). The internal nodes are **AND**, **OR**, and **NOT** gates. In other words, a Boolean circuit is an element of $\mathscr{F}[\mathbf{AND}, \mathbf{OR}, \mathbf{NOT}]$. We note that $\mathbf{NAND}, \mathbf{NOR} \in \mathscr{F}[\mathbf{AND}, \mathbf{OR}, \mathbf{NOT}]$ since

$$\mathbf{NAND} = \mathbf{NOT} \circ \mathbf{AND} \qquad \text{and} \qquad \mathbf{NOR} = \mathbf{NOT} \circ \mathbf{OR}\,.$$

In the following, we "show" that

$$\mathscr{F}[\mathbf{AND}, \mathbf{OR}, \mathbf{NOT}] = \mathscr{F}[\mathbf{NAND}] = \mathscr{F}[\mathbf{NOR}]\,. \tag{1.3}$$

To see this, it suffices to show that **AND**, **OR** and **NOT** can be constructed solely by **NAND** or **NOR**. The **AND** and **OR** gates can be implemented using **NAND** or **NOR** by the following logic circuit:

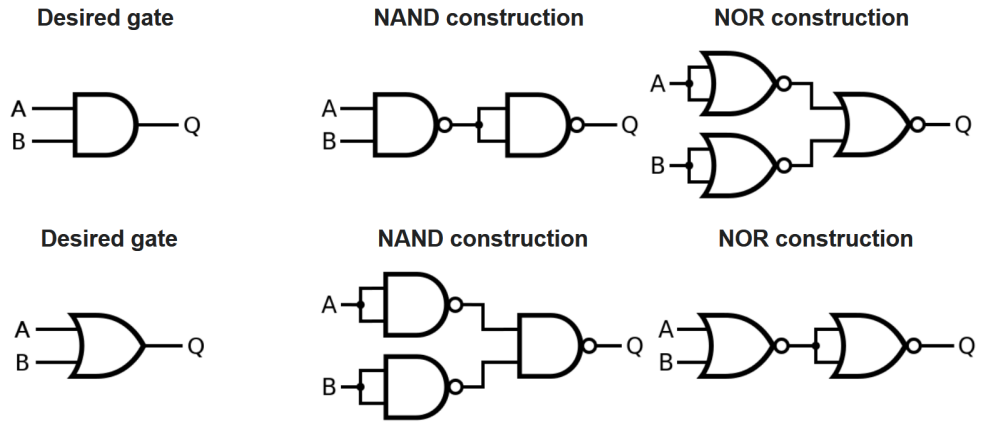



Figure 1.6: The construction of the **AND** and **OR** gates from the **NAND** or **NOR** gates

and the **NOT** gate can be constructed by **NAND** or **NOR** by the following logic circuit:

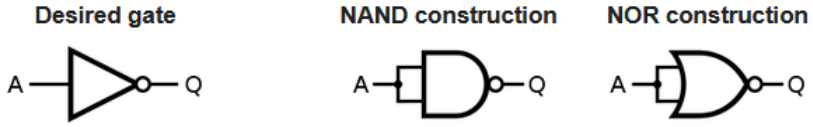



Figure 1.7: The construction of the **NOT** gate from the **NAND** or **NOR** gates

We also note that the **XOR** and **XNOR** gates can be constructed from **NAND** or **NOR** gates to construct these logic gates.

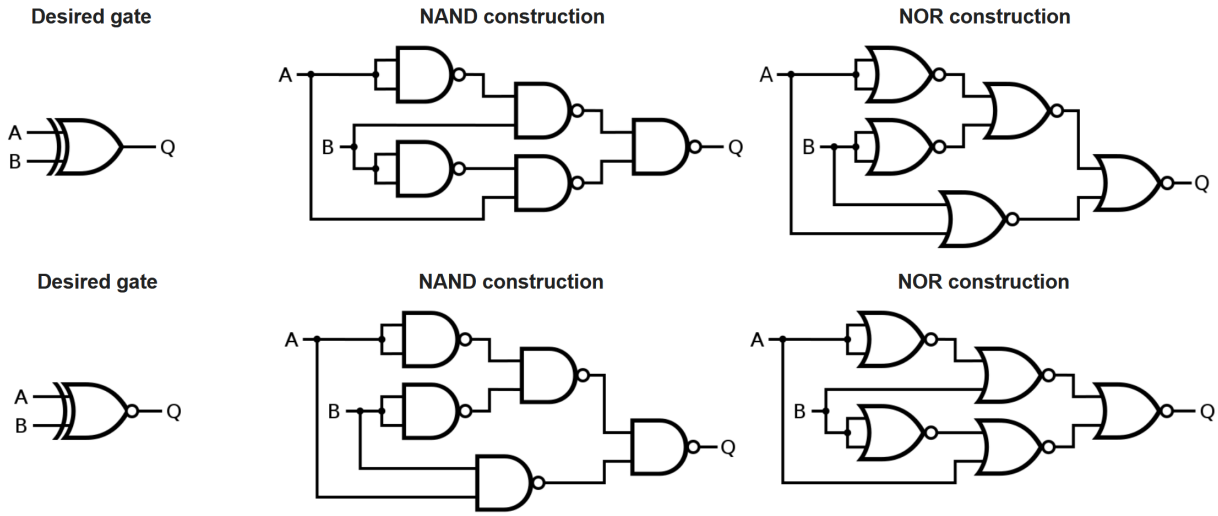



Figure 1.8: The construction of the **XOR** and **XNOR** gates from the **NAND** or **NOR** gates

**Remark 1.9.** In the construction of basic logic gate using **NAND** or **NOR**, the gate $\mathbf{COPY}^{(1)}$ is used implicitly. In other words, to be more precise (1.3) should be written as

$$\mathscr{F}[\mathbf{NAND}] \subseteq \mathscr{F}[\mathbf{AND}, \mathbf{OR}, \mathbf{NOT}] \subseteq \mathscr{F}[\mathbf{NAND}, \mathbf{COPY}^{(1)}]\,,$$
$$\mathscr{F}[\mathbf{NOR}] \subseteq \mathscr{F}[\mathbf{AND}, \mathbf{OR}, \mathbf{NOT}] \subseteq \mathscr{F}[\mathbf{NOR}, \mathbf{COPY}^{(1)}]\,.$$

The following proposition should be clear.

**Proposition 1.10.** *Let $\{g_1, \cdots, g_k\}$ be a collection of classical gates, and $h_1, \cdots, h_\ell \in \mathscr{F}[g_1, \cdots, g_k]$. Then*

$$\mathscr{F}[h_1, \cdots, h_\ell] \subseteq \mathscr{F}[g_1, \cdots, g_k]\,.$$

**Definition 1.11.** A collection $G = \{g_1, \cdots, g_k\}$ of classical gates is said to be **universal** if any gate $g$ can be constructed with gates from $G$; that is, $\{g_1, \cdots, g_k\}$ is universal if $g \in \mathscr{F}[g_1, \cdots, g_k]$ for every classical gate $g$.

**Theorem 1.12.** *The classical TOFFOLI-gate is universal and reversible.*

*Proof.* Since every gate $g : \{0,1\}^n \to \{0,1\}^m$ is a Cartesian product of $m$ gates $g_1, g_2, \cdots, g_m : \{0,1\}^n \to \{0,1\}$, it suffices to show the universality only for a gate of the form $f : \{0,1\}^n \to \{0,1\}$, which we shall do by induction in $n$.

Before initiating the induction argument, let us first construct the **AND**, **XOR** and $\mathbf{COPY}^{(n)}$ gates using the Toffoli gate. Since

$$\mathbf{TOF}(a,b,0) = (a,b,ab) \quad \text{and} \quad \mathbf{TOF}(1,a,b) = (1,a,a\oplus b) \qquad \forall\, a,b\in\{0,1\}\,,$$

we find that

$$\begin{aligned}
\mathbf{AND}(a,b) = ab &= (r_3^{(3)}\circ\mathbf{TOF})(a,b,0) = (r_3^{(3)}\circ\mathbf{TOF}\circ p_{0;3}^{(2)})(a,b)\,,\\
\mathbf{XOR}(a,b) = a\oplus b &= (r_3^{(3)}\circ\mathbf{TOF})(1,a,b) = (r_3^{(3)}\circ\mathbf{TOF}\circ p_{1;1}^{(2)})(a,b)\,,\\
\mathbf{COPY}^{(1)}(a) &= (r_{1,3}^{(3)}\circ\mathbf{TOF})(a,1,0) = (r_{1,3}^{(3)}\circ\mathbf{TOF}\circ p_{1,0;2,3}^{(1)})(a)\,.
\end{aligned}$$

Therefore, $\mathbf{AND},\mathbf{XOR},\mathbf{COPY}^{(1)}\in\mathscr{F}[\mathbf{TOF}]$. Together with Example 1.6, we also conclude that $\mathbf{COPY}^{(n)}\in\mathscr{F}[\mathbf{TOF}]$ for all $n\in\mathbb{N}$.

Now we initiate the induction process. First we need to show that **TOF** is universal for gates of the form $f:\{0,1\}\to\{0,1\}$. There are four gates in this case: the identity gate **ID**, the **NOT** gate, the **TRUE** gate whose output is always 1, and the **FALSE** gate whose output is always 0. Note that

$$\mathbf{TOF}(1,0,a) = (1,0,a) \quad \text{and} \qquad \mathbf{TOF}(1,1,a) = (1,1,1\oplus a) \qquad \forall\, a\in\{0,1\}\,.$$

Using the identity $p_{1,0;1,2}^{(1)}(a) = (1,0,a)$, we obtain that

$$\begin{aligned}
\mathbf{ID}(a) &= (r_3^{(3)}\circ\mathbf{TOF})(1,0,a) = (r_3^{(3)}\circ\mathbf{TOF}\circ p_{1,0;1,2}^{(1)})(a)\,,\\
\mathbf{TRUE}(a) &= (r_1^{(3)}\circ\mathbf{TOF})(1,0,a) = (r_1^{(3)}\circ\mathbf{TOF}\circ p_{1,0;1,2}^{(1)})(a)\,,\\
\mathbf{FALSE}(a) &= (r_2^{(3)}\circ\mathbf{TOF})(1,0,a) = (r_2^{(3)}\circ\mathbf{TOF}\circ p_{1,0;1,2}^{(1)})(a)\,,\\
\mathbf{NOT}(a) &= (r_3^{(3)}\circ\mathbf{TOF})(1,1,a) = (r_3^{(3)}\circ\mathbf{TOF}\circ p_{1,1;1,2}^{(1)})(a)\,.
\end{aligned}$$

Therefore, **TOF** is universal for gates of the form $f:\{0,1\}\to\{0,1\}$, and as a summary we have

$$\mathbf{ID},\mathbf{FALSE},\mathbf{TRUE},\mathbf{NOT},\mathbf{AND},\mathbf{XOR},\mathbf{COPY}^{(n)}\in\mathscr{F}[\mathbf{TOF}]\,. \tag{1.4}$$

Suppose that **TOF** is universal for gates of the form $f:\{0,1\}^{n-1}\to\{0,1\}$. Let $f:\{0,1\}^n\to\{0,1\}$ be a classical gate. Let $g_0,g_1:\{0,1\}^{n-1}\to\{0,1\}$ be classical gates given by

$$g_0(x_1,\cdots,x_{n-1}) = f(x_1,\cdots,x_{n-1},0) \quad \text{and} \quad g_1(x_1,\cdots,x_{n-1}) = f(x_1,\cdots,x_{n-1},1)\,,$$

and define $h:\{0,1\}^n\to\{0,1\}$ by

$$h(x_1,\cdots,x_n) = \mathbf{XOR}\big(\mathbf{AND}(g_0(x_1,\cdots,x_{n-1}),\mathbf{NOT}(x_n)),\mathbf{AND}(g_1(x_1,\cdots,x_{n-1}),x_n)\big)\,.$$

For a fixed $\widehat{x}_n\equiv(x_1,\cdots,x_{n-1})\in\{0,1\}^{n-1}$, there are four cases:

1. $g_0(\widehat{x}_n) = g_1(\widehat{x}_n) = 0$: in this case

$$h(x_1, \cdots, x_n) = \mathbf{XOR}\big(\mathbf{AND}(0, \mathbf{NOT}(x_n)), \mathbf{AND}(0, x_n)\big) = 0 = f(x_1, \cdots, x_n)\,.$$

2. $g_0(\widehat{x}_n) = g_1(\widehat{x}_n) = 1$: in this case

$$h(x_1, \cdots, x_n) = \mathbf{XOR}\big(\mathbf{AND}(1, \mathbf{NOT}(x_n)), \mathbf{AND}(1, x_n)\big) = 1 = f(x_1, \cdots, x_n)\,.$$

3. $g_0(\widehat{x}_n) = 0$ and $g_1(\widehat{x}_n) = 1$: in this case,

$$h(x_1, \cdots, x_n) = \mathbf{XOR}\big(\mathbf{AND}(0, \mathbf{NOT}(x_n)), \mathbf{AND}(1, x_n)\big) = \mathbf{ID}(x_n) = f(x_1, \cdots, x_n)\,.$$

4. $g_0(\widehat{x}_n) = 1$ and $g_1(\widehat{x}_n) = 0$: in this case,

$$h(x_1, \cdots, x_n) = \mathbf{XOR}\big(\mathbf{AND}(1, \mathbf{NOT}(x_n)), \mathbf{AND}(0, x_n)\big) = \mathbf{NOT}(x_n) = f(x_1, \cdots, x_n)\,.$$

Therefore, $h = f$. By the induction assumption, $g_0, g_1 \in \mathscr{F}[\mathbf{TOF}]$ and the fact that

$$f = h = \mathbf{XOR} \circ (\mathbf{AND} \times \mathbf{AND}) \circ \big((g_0 \times \mathbf{NOT}) \times (g_1 \times \mathbf{ID})\big) \circ \mathbf{COPY}^{(n)},$$

we conclude from (1.4) that $f \in \mathscr{F}[\mathbf{TOF}]$. □

**Remark 1.13.** Since **XOR** can be constructed using **NAND**, by Example 1.7 we find that $\mathbf{TOF} \in \mathscr{F}\big[\mathbf{NAND}, \mathbf{COPY}^{(1)}\big]$. Therefore, Theorem 1.12 and Remark 1.9 imply that

$$\mathscr{F}[\mathbf{TOF}] = \mathscr{F}\big[\mathbf{NAND}, \mathbf{COPY}^{(1)}\big] = \mathscr{F}[\mathbf{AND}, \mathbf{OR}, \mathbf{NOT}]\,.$$

## 1.3 How A Classical Computer Adds Numbers

### 1.3.1 Binary numbers

In a (classical) computer, each number is stored as a ***binary number*** which is a number expressed in the base-2 numeral system. In an $N$-bit system, the first bit is always used to store the information of non-negativity/negativity of the number, and the rest $(N-1)$ bits are used to express the number (we will not go further into the fixed point or floating point system).

Every **non-negative** binary number takes the form $0i_ni_{n-1}i_{n-2}\cdots i_1$ (or more precisely, $(0i_ni_{n-1}\cdots i_1)_2$), where $i_k \in \{0,1\}$ for each $k$, and is the same as the number

$$2^{n-1}i_n + 2^{n-2}i_{n-2} + \cdots + 2i_2 + i_1 = \sum_{k=1}^{n} 2^{k-1}i_k \tag{1.5}$$

in the usual base-10 numeral system. For example, the number 13 in the base-10 numeral system is expressed as $0\cdots01101$ in the base-2 numeral system.

Every **negative** binary number takes the form $1i_ni_{n-1}i_{n-2}\cdots i_1$ (or more precisely, $(1i_ni_{n-1}\cdots i_1)_2$), where $i_k \in \{0,1\}$ for each $k$, and is the same as the number

$$-2^{n-1}(1-i_n) - 2^{n-2}(1-i_{n-1}) - \cdots - 2^1(1-i_2) - (1-i_1) - 1 = -1 - \sum_{k=1}^{n} 2^{k-1}(1-i_k)$$

in the usual base-10 numeral system (here we use the two's-complement number system - 二補數系統). For example, the number $-13$ is $1\cdots10011$ (which is obtained by exchanging 0 and 1 in the binary expression of 13 and the outcome plus 1 is the binary expression of $-13$). We also note that the number $1i_ni_{n-1}i_{n-2}\cdots i_1$ is the same as $-2^n + 0i_ni_{n-1}\cdots i_1$, where $0i_ni_{n-1}\cdots i_1$ denotes the number given in (1.5).

**Example 1.14.** Since 7 in the base-10 numeral system is the same as $0\cdots0111$ is the base-2 numeral system, the classical computers compute $7+13$ and $7-13$ (which is the same as $7+(-13)$) as follows:

$$7+13 = (0\cdots00111)_2 + (0\cdots01101)_2 = (0\cdots010100)_2 = 2^4 + 2^2 = 20\,,$$
$$7+(-13) = (0\cdots00111)_2 + (1\cdots10011)_2 = (1\cdots111010)_2 = -2^2 - 2^0 - 1 = -6\,.$$

**Remark 1.15.** For a non-negative integer $k = (k_{n-1}k_{n-2}\cdots k_0)_2$, in matlab® $k_j$ is the $(j+1)$-th component of the vector $\boldsymbol{x}$ given by

$$\boldsymbol{x} = \textbf{de2bi}(k,n)\,.$$

In other words, $\boldsymbol{x}$ given above lists the lowest bit to the highest bit of $k$ from left to right. To obtain the bit expression in exactly the same order, we use the **flip** function so that

$$(k_{n-1}, k_{n-2}, \cdots, k_0) = \textbf{flip}(\textbf{de2bi}(k,n))\,.$$

We also remark that in matlab® the input of **de2bi** has to be non-negative integers (so it will not output the bit expression of negative integer in the two's complement number system).

### 1.3.2 Adder using logic circuits

An adder (加法器) is a digital circuit that performs addition of numbers. In many computers and other kinds of processors adders are used in the arithmetic logic units or ALU. The most common adders operate on binary numbers.

- **Half adder (半加法器)**

The half adder adds two single binary digits A and B. It has two outputs, sum (S) and carry (C, 進位). The carry signal represents an overflow into the next digit of a multi-digit addition. The sum of A and B is $2C + S$. The truth table for the half adder is:

| **INPUT** | | **OUTPUT** | |
|---|---|---|---|
| A | B | C | S |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

The simplest half-adder design, pictured below,

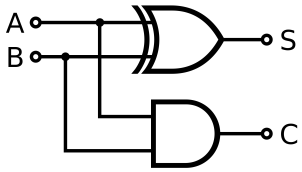



Figure 1.9: The logic diagram of the half adder

incorporates an **XOR** gate (that gives a true output when the number of true inputs is odd) for S and an **AND** gate for C. The Boolean logic for the sum (in this case S) will be $A'B + AB'$ (which is $(1 - A)B + A(1 - B)$) whereas for the carry (C) will be AB. The half adder adds two input bits and generates a carry and sum, which are the two outputs of a half adder.

- **Full adder (全加法器)**

A full adder adds binary numbers and accounts for values carried in as well as out. A one-bit full-adder adds **three** one-bit numbers, often written as A, B, and $C_{in}$; A and B are the operands, and $C_{in}$ is a bit carried in from the previous less-significant stage. The circuit

produces a two-bit output. Output carry and sum typically represented by the signals $C_{out}$ and S, where the sum of A and B equals $2C_{out} + S$. The truth table for the full adder is:

| INPUT | | | OUTPUT | |
|---|---|---|---|---|
| A | B | $C_{in}$ | $C_{out}$ | S |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

A circuit design for the full adder is given below:

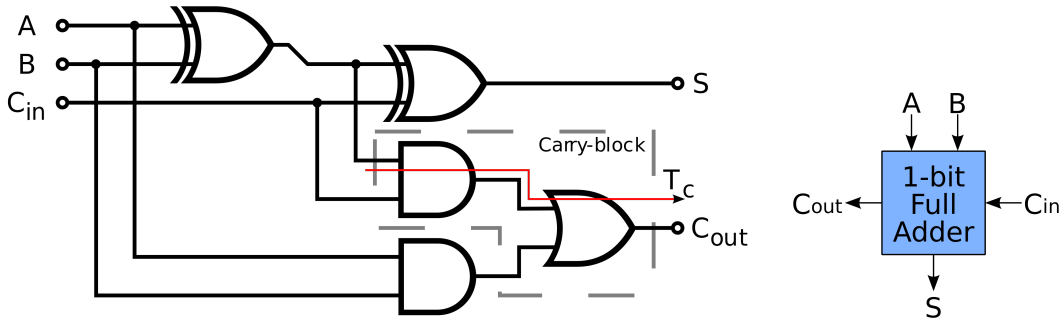
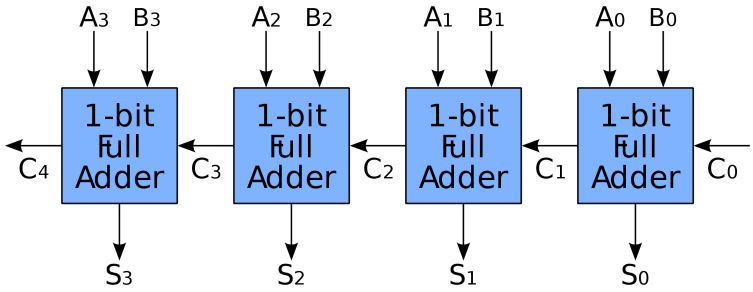



Figure 1.10: The logic diagram of the full adder (left) and a schematic symbol for a 1-bit full adder (right), here $C_{in}$ and $C_{out}$ drawn on sides of block to emphasize their use in a multi-bit adder

We can create a logical circuit using multiple full adders to add N-bit numbers. Each full adder inputs a $C_{in}$, which is the $C_{out}$ of the previous adder. This kind of adder is called a **ripple-carry adder** (RCA), since each carry bit "ripples" to the next full adder. Note that the first (and only the first) full adder may be replaced by a half adder (under the assumption that $C_{in} = 0$). The following figure provides a symbol for a 4-bit full adder:



here two input 4-bit numbers is $A = (A_3A_2A_1A_0)_2$, $B = (B_3B_2B_1B_0)_2$ and the sum of A and B is a 5-bit number $S = (C_4S_3S_2S_1S_0)_2$.

## 1.4 Classical Circuits

In classical complexity theory, a Boolean circuit is a finite directed acyclic graph with **AND**, **OR**, and **NOT** gates. It has $n$ input nodes, which contain the $n$ input bits ($n \geqslant 0$). The internal nodes are **AND**, **OR**, and **NOT** gates, and there are one or more designated output nodes. The initial input bits are fed into **AND**, **OR**, and **NOT** gates according to the circuit, and eventually the output nodes assume some value. We say that a circuit computes some Boolean function $f : \{0,1\}^n \to \{0,1\}^m$ if the output nodes get the right value $f(x)$ for every input $x \in \{0,1\}^n$.

A circuit family is a collection $C = \{C_n\}$ of circuits, one for each input size $n$. Each circuit has one output bit. Such a family recognizes or decides a language $L \subseteq \{0,1\}^* \equiv \bigcup_{n \geqslant 0} \{0,1\}^n$ if, for every $n$ and every input $x \in \{0,1\}^n$, the circuit $C_n$ outputs 1 if $x \in L$ and outputs 0 otherwise. Such a circuit family is ***uniformly polynomial*** if there is a deterministic Turing machine that outputs $C_n$ given $n$ as input, using space logarithmic in $n$. Note that the size (number of gates) of the circuits $C_n$ can then grow at most polynomially with $n$. It is known that uniformly polynomial circuit families are equal in power to polynomial-time deterministic Turing machines: a language $L$ can be decided by a uniformly polynomial circuit family $L \in \mathbf{P}$, where **P** is the class of languages decidable by polynomial-time Turing machines. Similarly we can consider randomized circuits. These receive, in addition to the n input bits, also some random bits ("coin flips") as input. A randomized circuit computes a function $f$ if it successfully outputs the right answer $f(x)$ with probability at least $2/3$ for every $x$ (probability taken over the values of the random bits). Randomized circuits are equal in power to randomized Turing machines: a language $L$ can be decided by a uniformly polynomial randomized circuit family $L \in \mathbf{BPP}$, where **BPP** ("Bounded-error Probabilistic Polynomial time") is the class of languages that can efficiently be recognized by randomized Turing machines with success probability at least $2/3$. Because we can efficiently reduce the error probability of randomzied algorithms (see Appendix B.2), the particular value $2/3$ does not really matter here and may be replaced by any fixed constant in $(1/2, 1)$.

# Chapter 2

# Quantum Computing

Classical computers carry out logical operations using the "**definite position** of a physical state" (also called classical state). These are usually binary, meaning its operations are based on one of two positions. A single state - such as on or off, up or down, 1 or 0 - is called a bit.

In quantum computing, operations instead use the quantum state of an object. These states have indefinite/undetermined positions before they are measured, such as the spin of an electron or the polarisation of a photon. Rather than having a clear position, unmeasured quantum states occur in a mixed "superposition", not unlike a coin spinning through the air before it lands in your hand. These superpositions can be entangled with those of other objects, meaning their final outcomes will be mathematically related even if we do not know yet what they are.

In a classical computer, each number is in classical state. Call these states $|1\rangle, |2\rangle, \cdots, |N\rangle$ (here we treat $|1\rangle, \cdots, |N\rangle$ as $N$ distinct outcomes but not necessarily natural numbers from 1 to $N$). A superposition of these states is a quantum state

$$|\psi\rangle = \alpha_1|1\rangle + \alpha_2|2\rangle + \cdots + \alpha_N|N\rangle,$$

where $\alpha_1, \cdots, \alpha_N$ are complex numbers satisfying $|\alpha_1|^2 + \cdots + |\alpha_N|^2 = 1$ and this particular quantum state, upon measurement, gives $|j\rangle$ with probability $|\alpha_j|^2$. Quantum computers perform calculations based on the probability of an object's quantum state before it is measured - instead of just 1s or 0s - which means they have the potential to process exponentially more data compared to classical computers. Quantum computation is the field that investigates the computational power and other properties of computers based on quantum-mechanical principles. An important objective is to find quantum algorithms that

are significantly faster than any classical algorithm solving the same problem.

## 2.1 Quantum Mechanics

Here we give a brief and abstract introduction to quantum mechanics. In short: a quantum state is a superposition of classical states, to which we can apply either a measurement or a unitary operation.

### 2.1.1 Schrödinger equation

In "continuous" quantum mechanics, the Schrödinger equation for a single non-relativistic particle with mass $m$ is given by

$$i\hbar\frac{\partial}{\partial t}\psi = \Big(-\frac{\hbar}{2m}\Delta + V\Big)\psi \qquad \text{in} \quad \mathbb{R}^n \times \{t > 0\}\,, \tag{2.1}$$

where $\hbar \approx 1.05457181765 \times 10^{-34} J \cdot s$ is the reduced Planck constant, $\psi = \psi(x,t)$ is the wave function, a function that assigns a complex number to each point $x$ at each time $t$, and $V = V(x,t)$ is a real-valued function, called the potential, that represents the environment in which the particle exists. The square of the absolute value of the wave function at each point is taken to define a probability density function: given a wave function in position space $\psi(x,t)$ as above, the function $\big|\psi(x,t)\big|^2$ denotes the probability density of the presence of the particle at position $x$ at time $t$.

Taking the complex conjugate of the Schrödinger equation (2.1), we obtain that

$$-i\hbar\frac{\partial}{\partial t}\bar{\psi} = \Big(-\frac{\hbar}{2m}\Delta + V\Big)\bar{\psi}$$

thus

$$i\hbar\bar{\psi}\frac{\partial}{\partial t}\psi = \bar{\psi}\Big(-\frac{\hbar}{2m}\Delta + V\Big)\psi\,, \qquad i\hbar\psi\frac{\partial}{\partial t}\bar{\psi} = -\psi\Big(-\frac{\hbar}{2m}\Delta + V\Big)\bar{\psi}\,.$$

Therefore,

$$i\hbar\frac{\partial}{\partial t}|\psi|^2 = i\hbar\frac{\partial}{\partial t}(\bar{\psi}\psi) = \frac{\hbar}{2m}\big(\psi\Delta\bar{\psi} - \bar{\psi}\Delta\psi\big)$$

so that the divergence theorem implies that

$$i\hbar\frac{d}{dt}\int_{\mathbb{R}^3}\big|\psi(x,t)\big|^2\,dx = \frac{\hbar}{2m}\int_{\mathbb{R}^3}\big[\psi(x,t)\Delta\bar{\psi}(x,t) - \bar{\psi}(x,t)\Delta\psi(x,t)\big]\,dx = 0\,.$$

Therefore, $\displaystyle\int_{\mathbb{R}^3}\big|\psi(x,t)\big|^2\,dx$ is a constant (which is assumed to be 1 if at a certain time this

integral is 1). This shows that the probability of the presence of a particle (whose dynamics is described by (2.1)) at a certain point in $\mathbb{R}^3$ is 1. The physical interpretation of this identity is "the position at which the particle locates is in a superposition of all the points in $\mathbb{R}^3$".

On the other hand, when you try to figure out the location of the particle by implementing some kind of measurements, you always obtain an unambiguous result. The outcome of the measurement follows the probability distribution that the probability density function $\left|\psi(\cdot,t)\right|^2$ provides: the probability of that the particle locations in the region $R \subseteq \mathbb{R}^3$ at time $t$ is given by $\int_R \left|\psi(x,t)\right|^2 dx$.

**Definition 2.1.** A **quantum state** is a mathematical entity that provides a probability distribution for the outcomes of each possible measurement on a system.

### 2.1.2 Superposition

In quantum computing, each number is a superposition of "classical numbers".Consider some physical system that can be in $N$ different, mutually exclusive classical states. Call these states $|1\rangle$, $|2\rangle$, $\cdots$, $|N\rangle$. A superposition of these states is described by the wave function

$$\phi(x,t) = \begin{cases} \alpha_1 & \text{if } x = |1\rangle, \\ \alpha_2 & \text{if } x = |2\rangle, \\ \vdots & \\ \alpha_N & \text{if } x = |N\rangle, \end{cases}$$

where $\alpha_j$ is a complex number called the **amplitude** of $|j\rangle$ in $|\phi\rangle$, and $\alpha_1, \cdots, \alpha_N$ satisfy $|\alpha_1|^2 + |\alpha_2|^2 + \cdots + |\alpha_N|^2 = 1$. The wave function above is a pure quantum state (usually just called state) and is usually written as

$$|\phi\rangle = \sum_{j=1}^{N} \alpha_j |j\rangle = \alpha_1|1\rangle + \alpha_2|2\rangle + \cdots + \alpha_N|N\rangle.$$

Intuitively, a system in quantum state $|\phi\rangle$ is in all classical states at the same time! It is in state $|1\rangle$ with amplitude $\alpha_1$ (and probability $|\alpha_1|^2$), in state $|2\rangle$ with amplitude $\alpha_2$ (and probability $|\alpha_2|^2$), and so on. Mathematically, the states $|1\rangle$, $\cdots$, $|N\rangle$ form an orthonormal basis of an $N$-dimensional Hilbert space (that is, an $N$-dimensional vector space equipped with an inner product), and a quantum state $|\phi\rangle$ is a vector in this space.

**Notation**: Let $(\mathbb{H}, \langle\cdot,\cdot\rangle)$ be a Hilbert space over field $\mathbb{F}$. Any vectors $\boldsymbol{v}$ in $\mathbb{H}$ is expressed as $|\boldsymbol{v}\rangle$. For example, in "continuous" quantum mechanics every quantum state $|\psi\rangle$ lives in the Hilbert space $L^2(\mathbb{R}^3)$. For a vector $\boldsymbol{v} \in \mathbb{H}$, the notation $\langle \boldsymbol{v}|$ is an element in the dual space of $\mathbb{H}$ (see Definition 3.46) satisfying $\langle \boldsymbol{v}|\boldsymbol{w}\rangle \equiv \langle \boldsymbol{v}, \boldsymbol{w}\rangle$. In other word, for each $\boldsymbol{w} \in \mathbb{H}$, $\boldsymbol{w} = \alpha\boldsymbol{v} + \beta\boldsymbol{v}^{\perp}$ for some $\alpha \in \mathbb{F}$ and $\langle \boldsymbol{v}| : \boldsymbol{w} \mapsto \alpha\|\boldsymbol{v}\|^2$.

### 2.1.3 Measurement

There are two things we can do with a quantum state: measure it or let it evolve unitarily without measuring it. We will deal with measurement first.

- **Measurement in the computational basis**

Suppose we measure state $|\phi\rangle$. We cannot "see" a superposition itself, but only classical states. Accordingly, if we measure state $|\phi\rangle$ we will see one and only one classical state $|j\rangle$. Which specific $|j\rangle$ will we see? This is not determined in advance; the only thing we can say is that we will see state $|j\rangle$ with probability $|\alpha_j|^2$, which is the squared norm of the corresponding amplitude $\alpha_j$ ($|a + ib| = \sqrt{a^2 + b^2}$ for $a, b \in \mathbb{R}$). Thus observing a quantum state induces a probability distribution on the classical states, given by the squared norms of the amplitudes. This implies $\sum_{j=1}^{N} |\alpha_j|^2 = 1$, so the vector of amplitudes has (Euclidean) norm 1. If we measure $|\phi\rangle$ and see classical state $|j\rangle$ as a result, then $|\phi\rangle$ itself has "disappeared", and all that is left is $|j\rangle$. In other words, observing $|\phi\rangle$ "collapses" the quantum superposition $|\phi\rangle$ to the classical state $|j\rangle$ that we saw, and all "information" that might have been contained in the amplitudes $\alpha_i$ is gone.

- **Projective measurement**

A somewhat more general kind of measurement than the above "measurement in the computational (or standard) basis" is possible. This will be used only sparsely in the course, so it may be skipped on a first reading. Such a projective measurement is described by projectors $\mathrm{P}_1$, $\mathrm{P}_2$, $\cdots$, $\mathrm{P}_m$ ($m \leqslant N$) which sum to identity. These projectors are then pairwise orthogonal, meaning that $\mathrm{P}_i\mathrm{P}_j = 0$ if $i \neq j$. The projector $\mathrm{P}_j$ projects on some subspace $\mathbb{H}_j$ of the total Hilbert space $\mathbb{H}$, and every state $|\phi\rangle \in \mathbb{H}$ can be decomposed in a unique way as $|\phi\rangle = \sum_{j=1}^{N} |\phi_j\rangle$, with $|\phi_j\rangle = \mathrm{P}_j|\phi\rangle \in \mathbb{H}_j$. Because the projectors are orthogonal, the subspaces $\mathbb{H}_j$ are orthogonal as well, as are the states $|\phi_j\rangle$. When we apply this measurement to the

pure state $|\phi\rangle$, then we will get outcome in $\mathbb{H}_j$ with probability $\||\phi_j\rangle\|^2 = \mathrm{tr}(\mathrm{P}_j|\phi\rangle\langle\phi|)$ and the state will then "collapse" to the new state $|\phi_j\rangle/\||\phi_j\rangle\| = \mathrm{P}_j|\phi\rangle/\|\mathrm{P}_j|\phi\rangle\|$.

**Example 2.2.** A measurement in the standard basis is the specific projective measurement where $m = N$ and $\mathrm{P}_j = |j\rangle\langle j|$; that is, $\mathrm{P}_j$ projects onto the standard basis state $|j\rangle$ and the corresponding subspace $\mathbb{H}_j$ is the space spanned by $|j\rangle$. Consider the state $|\phi\rangle = \sum_{j=1}^{N} \alpha_j|j\rangle$. Note that $\mathrm{P}_j|\phi\rangle = \alpha_j|j\rangle$, so applying our measurement to $|\phi\rangle$ will give outcome in $\mathbb{H}_j$ with probability $\|\alpha_j|j\rangle\|^2 = |\alpha_j|^2$, and in that case the state collapses to $\dfrac{\alpha_j|j\rangle}{\|\alpha_j|j\rangle\|} = \dfrac{\alpha_j}{|\alpha_j|}|j\rangle$. The norm-1 factor $\dfrac{\alpha_j}{|\alpha_j|}$ may be disregarded because it has no physical significance, so we end up with the state $|j\rangle$ as we saw before.

**Example 2.3.** A measurement that distinguishes between $|j\rangle$ with $j \leqslant \dfrac{N}{2}$ and $|j\rangle$ with $j > \dfrac{N}{2}$ corresponds to the two projectors $\mathrm{P}_1 = \sum_{j\leqslant N/2} |j\rangle\langle j|$ and $\mathrm{P}_2 = \sum_{j>N/2} |j\rangle\langle j|$. Applying this measurement to the state

$$|\phi\rangle = \frac{1}{2}|1\rangle + \frac{\sqrt{3}}{\sqrt{8}}|2\rangle + \frac{1}{2}|N-1\rangle + \frac{1}{\sqrt{8}}|N\rangle\,,$$

where $N \geqslant 4$, will give outcome 1 with probability $\|\mathrm{P}_1|\phi\rangle\|^2 = \dfrac{5}{8}$, in which case the state collapses to $\dfrac{\sqrt{2}}{\sqrt{5}}|1\rangle + \dfrac{\sqrt{3}}{\sqrt{5}}|2\rangle$, and will give outcome 2 with probability $\|\mathrm{P}_2|\phi\rangle\|^2 = \dfrac{3}{8}$, in which case the state collapses to $\dfrac{\sqrt{2}}{\sqrt{3}}|N-1\rangle + \dfrac{1}{\sqrt{3}}|N\rangle$.

### 2.1.4 Unitary evolution

Instead of measuring $|\phi\rangle$, we can also apply some operation to it; that is, change the state $|\phi\rangle$ to some other state

$$|\psi\rangle = \sum_{j=1}^{N} \beta_j|j\rangle = \beta_1|1\rangle + \beta_2|2\rangle + \cdots + \beta_N|N\rangle\,.$$

Quantum mechanics only allows linear operations to be applied to quantum states. What this means is: if we view a state like $|\phi\rangle$ as an $N$-dimensional vector $[\alpha_1, \alpha_2, \cdots, \alpha_N]^{\mathrm{T}}$ (sometimes called the "**qubit state vector**"), then applying an operation that changes $|\phi\rangle$

to $|\psi\rangle$ corresponds to multiplying $|\phi\rangle$ with an $N \times N$ complex-valued matrix U:

$$\mathrm{U}\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_N \end{bmatrix}.$$

Note that by linearity we have

$$|\psi\rangle = \mathrm{U}|\phi\rangle = \mathrm{U}\Big(\sum_{j=1}^{N} \alpha_j |j\rangle\Big) = \sum_{j=1}^{N} \alpha_j \mathrm{U}|j\rangle\,.$$

Because measuring $|\psi\rangle$ should also give a probability distribution, P we have the constraint $\sum_{j=1}^{N} |\beta_j|^2 = 1$. This implies that the operation U must preserve the norm of vectors, and U always maps a vector of norm 1 to a vector of norm 1. Such a linear map is said to be unitary and always has an inverse (since $\mathrm{U}\boldsymbol{x} = \mathbf{0}$ if and only if $\boldsymbol{x} = \mathbf{0}$), and it follows that any (non-measuring) operation on quantum states must be reversible: by applying $\mathrm{U}^{-1}$ we can always "undo" the action of U, and nothing is lost in the process. On the other hand, a measurement is clearly non-reversible, because we cannot reconstruct $|\phi\rangle$ from the observed classical state $|j\rangle$.

## 2.2 Qubits and Quantum Gates

In the previous sections, we talked about the superposition

$$|\phi\rangle = \sum_{j=1}^{N} \alpha_j |j\rangle = \alpha_1|1\rangle + \alpha_2|2\rangle + \cdots + \alpha_N|N\rangle$$

of $N$ classical states. In a quantum computer, $|\phi\rangle$ is used to expressed a random numbers. Each such number is created using random bits, called qubits, and every qubit can be created with different amplitude (or probability) of the 0 and 1 state. A 1-qubit state is represented in braket notation as $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$, and an $n$-qubit state is represented as

$$|\phi\rangle = \sum_{j=0}^{2^n-1} \alpha_j |j\rangle \qquad \text{or} \qquad |\phi\rangle = \sum_{j=0}^{2^n-1} \alpha_j |j_{n-1}\cdots j_0\rangle\,,$$

where $(0j_{n-1}\cdots j_1 j_0)_2$ is the binary representation of $j$; that is,

$$j = 2^{n-1}j_{n-1} + 2^{n-2}j_{n-2} + \cdots + 2^1 j_1 + j_0\,.$$

### 2.2.1 Quantum bits

**Definition 2.4** (Qubits)**.** A qubit is a quantum state with two possible outcomes of measurement. A qubit is usually represented by

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

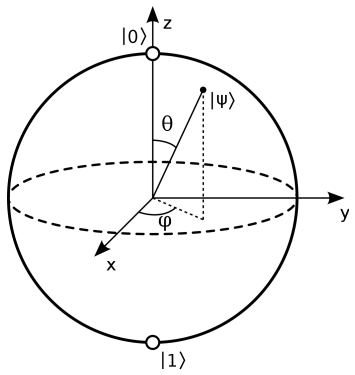
where $\alpha, \beta \in \mathbb{C}$ satisfying $|\alpha|^2 + |\beta|^2 = 1$. Two qubits $|\psi_1\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$ and $|\psi_2\rangle = \alpha_2|0\rangle + \beta_2|1\rangle$ are said to be equivalent if there exists $\theta \in \mathbb{R}$ such that $(\alpha_2, \beta_2) = e^{i\theta}(\alpha_1, \beta_1)$.

**Remark 2.5.** A qubit is more than a two-valued random variable.

**Definition 2.6.** A Bloch sphere $B$ is a subset of $\mathbb{C}^2$ defined by $(\alpha, \beta) \in B$ if and only if $|\alpha|^2 + |\beta|^2 = 1$. Each point $(\alpha, \beta) \in B$ is represented by

$$|\psi\rangle = e^{i\delta}\Big(\cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle\Big),$$

where $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi)$.




### 2.2.2 Quantum gates

A unitary transformation that acts on a small numer of qubits (say, at most 3) is often called a gate, in analogy to classical logic gates. Two simple but important 1-qubit gates are the bitflip-gate X (which negates the bit; that is, swaps $|0\rangle$ and $|1\rangle$) and the phaseflip gate Z (which puts a minus sign "$-$" in front of $|1\rangle$). Represented as $2 \times 2$ unitary matrices, these are

$$\mathrm{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad \text{and} \qquad \mathrm{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{2.2}$$

**Remark 2.7.** Let $|\psi\rangle = e^{i\delta}\Big(\cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle\Big)$ be a 1-qubit quantum state. Then on the Bloch sphere,

1. $\mathrm{X}|\psi\rangle$ is the reflection of $|\psi\rangle$ (or the rotation by angel $\pi$) about the $x$-axis); that is,
$$\begin{aligned}\mathrm{X}|\psi\rangle &= e^{i\delta}\Big(\cos\frac{\pi-\theta}{2}|0\rangle + e^{-i\phi}\sin\frac{\pi-\theta}{2}|1\rangle\Big) = e^{i(\delta-\phi)}\Big(e^{i\phi}\sin\frac{\theta}{2}|0\rangle + \cos\frac{\theta}{2}|1\rangle\Big)\\ &= e^{i(\delta-\phi)}\Big(\cos\frac{\theta}{2}|1\rangle + e^{i\phi}\sin\frac{\theta}{2}|0\rangle\Big).\end{aligned}$$

2. $\mathrm{Z}|\psi\rangle$ is the reflection of $|\psi\rangle$ (or the rotation by angel $\pi$) about the $z$-axis; that is, then
$$\mathrm{Z}|\psi\rangle = e^{i\delta}\Big(\cos\frac{\theta}{2}|0\rangle + e^{i(\pi+\phi)}\sin\frac{\theta}{2}|1\rangle\Big) = e^{i\delta}\Big(\cos\frac{\theta}{2}|0\rangle - e^{i\phi}\sin\frac{\theta}{2}|1\rangle\Big).$$

Possibly the most important 1-qubit gate is the Hadamard transform, specified by:

$$H|0\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \qquad \text{and} \qquad H|1\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle.$$

The Hadamard transform is represented as

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

If we apply H to initial state $|0\rangle$ and then measure, we have equal probability of observing $|0\rangle$ or $|1\rangle$. Similarly, applying H to $|1\rangle$ and observing gives equal probability of $|0\rangle$ or $|1\rangle$. However, if we apply H to the superposition $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ then we obtain $|0\rangle$: the positive and negative amplitudes for $|1\rangle$ cancel out! (note that this also means that H is its own inverse) This effect is called interference, and is analogous to interference patterns between light or sound waves.

Let us also consider the reflection (or the rotation by angle $\pi$) about the $y$-axis. This rotation is denoted by Y and is given by

$$\cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle \overset{\mathrm{Y}}{\mapsto} \cos\frac{\pi-\theta}{2}|0\rangle + e^{i(\pi-\phi)}\sin\frac{\pi-\theta}{2}|1\rangle$$

so that the matrix representation of Y is

$$\mathrm{Y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}.$$

These three gates X, Y, Z are called the Pauli gates. We note that if $A$ and $B$ are two different Pauli gates, then $AB + BA = 0$.

**Remark 2.8.** In principle, the matrix representation of a quantum gate can differ by a multiple of a constant whose modulus is 1 because these representations give equivalent quantum states. We choose X, Y and Z in such a way that $\mathrm{X}^2 = \mathrm{Y}^2 = \mathrm{Z}^2 = \mathrm{I}$.

In general, we can consider the rotation by angle $\tau$ about the $x$-axis, $y$-axis and $z$-axis. These rotations are denoted by $\mathrm{R}_x(\tau)$, $\mathrm{R}_y(\tau)$ and $\mathrm{R}_z(\tau)$, respectively.

**Theorem 2.9.** *For $\tau \in \mathbb{R}$, the matrix representations of $\mathrm{R}_x(\tau)$, $\mathrm{R}_y(\tau)$ and $\mathrm{R}_z(\tau)$ are respectively given by*

$$\mathrm{R}_x(\tau) = \begin{bmatrix} \cos\frac{\tau}{2} & -i\sin\frac{\tau}{2} \\ -i\sin\frac{\tau}{2} & \cos\frac{\tau}{2} \end{bmatrix}, \quad \mathrm{R}_y(\tau) = \begin{bmatrix} \cos\frac{\tau}{2} & -\sin\frac{\tau}{2} \\ \sin\frac{\tau}{2} & \cos\frac{\tau}{2} \end{bmatrix}, \quad \mathrm{R}_z(\tau) = \begin{bmatrix} e^{-i\frac{\tau}{2}} & 0 \\ 0 & e^{i\frac{\tau}{2}} \end{bmatrix}. \quad (2.3)$$

*Proof.* Let $|\psi\rangle$ be a 1-qubit quantum state

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$$

whose coordinate on the Bloch sphere is $\cos\phi\sin\theta\boldsymbol{i} + \sin\phi\sin\theta\boldsymbol{j} + \cos\theta\boldsymbol{k}$.

1. On the unit sphere, the rotation of the vector $\cos\phi\sin\theta\boldsymbol{i} + \sin\phi\sin\theta\boldsymbol{j} + \cos\theta\boldsymbol{k}$ with angle $\tau$ about the $x$-axis is

   $$\cos\phi\sin\theta\boldsymbol{i} + (\cos\tau\sin\phi\sin\theta - \sin\tau\cos\theta)\boldsymbol{j} + (\sin\tau\sin\phi\sin\theta + \cos\tau\cos\theta)\boldsymbol{k},$$

   where the coefficients for $\boldsymbol{j}$ and $\boldsymbol{k}$ are obtained by

   $$\begin{bmatrix} \cos\tau & -\sin\tau \\ \sin\tau & \cos\tau \end{bmatrix} \begin{bmatrix} \sin\phi\sin\theta \\ \cos\theta \end{bmatrix} = \begin{bmatrix} \cos\tau\sin\phi\sin\theta - \sin\tau\cos\theta \\ \sin\tau\sin\phi\sin\theta + \cos\tau\cos\theta \end{bmatrix}.$$

   Suppose that

   $$\begin{aligned} &\cos\phi\sin\theta\boldsymbol{i} + (\cos\tau\sin\phi\sin\theta - \sin\tau\cos\theta)\boldsymbol{j} + (\sin\tau\sin\phi\sin\theta + \cos\tau\cos\theta)\boldsymbol{k} \\ &\quad = \cos\varphi\sin\vartheta\boldsymbol{i} + \sin\varphi\sin\vartheta\boldsymbol{j} + \cos\vartheta\boldsymbol{k} \end{aligned}$$

   for some $\varphi$ and $\vartheta$. Then

   $$\cos^2\frac{\vartheta}{2} = \frac{1 + \sin\tau\sin\phi\sin\theta + \cos\tau\cos\theta}{2}, \ \tan\varphi = \frac{\cos\tau\sin\phi\sin\theta - \sin\tau\cos\theta}{\cos\phi\sin\theta}. \quad (2.4)$$

   Next we show that $\mathrm{R}_x(\tau)$ with matrix representation given by (2.3) indeed has the property that

   $$\mathrm{R}_x(\tau)|\psi\rangle = e^{i\delta}\Big(\cos\frac{\vartheta}{2}|0\rangle + e^{i\varphi}\sin\frac{\vartheta}{2}|1\rangle\Big)$$

for some $\delta \in \mathbb{R}$. Expanding $\begin{bmatrix} \cos\frac{\tau}{2} & -i\sin\frac{\tau}{2} \\ -i\sin\frac{\tau}{2} & \cos\frac{\tau}{2} \end{bmatrix} \begin{bmatrix} \cos\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2} \end{bmatrix}$, it is to show that there exists $\delta \in \mathbb{R}$ such that

$$\cos\frac{\tau}{2}\cos\frac{\theta}{2} + \sin\phi\sin\frac{\tau}{2}\sin\frac{\theta}{2} - i\cos\phi\sin\frac{\tau}{2}\sin\frac{\theta}{2} = e^{i\delta}\cos\frac{\vartheta}{2}\,, \tag{2.5a}$$

$$\cos\phi\cos\frac{\tau}{2}\sin\frac{\theta}{2} + i\big(\sin\phi\cos\frac{\tau}{2}\sin\frac{\theta}{2} - \cos\frac{\theta}{2}\sin\frac{\tau}{2}\big) = e^{i(\delta+\varphi)}\sin\frac{\vartheta}{2}\,. \tag{2.5b}$$

Since

$$\begin{aligned}
&\big(\cos\frac{\tau}{2}\cos\frac{\theta}{2} + \sin\phi\sin\frac{\tau}{2}\sin\frac{\theta}{2}\big)^2 + \cos^2\phi\sin^2\frac{\tau}{2}\sin^2\frac{\theta}{2} \\
&\quad= \cos^2\frac{\tau}{2}\cos^2\frac{\theta}{2} + \sin^2\frac{\tau}{2}\sin^2\frac{\theta}{2} + 2\cos\frac{\tau}{2}\cos\frac{\theta}{2}\sin\phi\sin\frac{\tau}{2}\sin\frac{\theta}{2} \\
&\quad= \frac{(1+\cos\tau)(1+\cos\theta) + (1-\cos\tau)(1-\cos\theta)}{4} + \frac{\sin\phi\sin\tau\sin\theta}{2} \\
&\quad= \frac{1+\cos\tau\cos\theta+\sin\phi\sin\tau\sin\theta}{2} = \cos^2\frac{\vartheta}{2}\,,
\end{aligned}$$

there exists $\delta \in \mathbb{R}$ such that

$$\cos\frac{\tau}{2}\cos\frac{\theta}{2} + \sin\phi\sin\frac{\tau}{2}\sin\frac{\theta}{2} - i\cos\phi\sin\frac{\tau}{2}\sin\frac{\theta}{2} = e^{i\delta}\cos\frac{\vartheta}{2}\,;$$

thus (2.5a) holds. Moreover, by the fact that $\mathrm{R}_x(\tau)$ given by (2.3) is unitary, (2.5a) implies that

$$\Big|\cos\phi\cos\frac{\tau}{2}\sin\frac{\theta}{2} + i\big(\sin\phi\cos\frac{\tau}{2}\sin\frac{\theta}{2} - \cos\frac{\theta}{2}\sin\frac{\tau}{2}\big)\Big|^2 = \sin^2\frac{\vartheta}{2}\,, \tag{2.6}$$

Therefore, to show (2.5b) it suffices to extract the phase information. Computing the product of the left-hand side of (2.5b) and the complex conjugate of (2.5a), we obtain that

$$\begin{aligned}
&\big(\cos\frac{\tau}{2}\cos\frac{\theta}{2} + \sin\phi\sin\frac{\tau}{2}\sin\frac{\theta}{2} + i\cos\phi\sin\frac{\tau}{2}\sin\frac{\theta}{2}\big)\times \\
&\quad\times\Big[\cos\phi\cos\frac{\tau}{2}\sin\frac{\theta}{2} + i\big(\sin\phi\cos\frac{\tau}{2}\sin\frac{\theta}{2} - \cos\frac{\theta}{2}\sin\frac{\tau}{2}\big)\Big] \\
&\qquad= \cos\phi\cos^2\frac{\tau}{2}\sin\frac{\theta}{2}\cos\frac{\theta}{2} + \cos\phi\sin^2\frac{\tau}{2}\sin\frac{\theta}{2}\cos\frac{\theta}{2} \\
&\qquad\quad+ i\Big[\cos^2\phi\sin^2\frac{\theta}{2}\sin\frac{\tau}{2}\cos\frac{\tau}{2} + \sin^2\phi\sin^2\frac{\theta}{2}\sin\frac{\tau}{2}\cos\frac{\tau}{2} - \cos^2\frac{\theta}{2}\sin\frac{\tau}{2}\cos\frac{\tau}{2} \\
&\qquad\qquad+ \sin\phi\cos^2\frac{\tau}{2}\sin\frac{\theta}{2}\cos\frac{\theta}{2} - \sin\phi\sin^2\frac{\tau}{2}\sin\frac{\theta}{2}\cos\frac{\theta}{2}\Big] \\
&\qquad= \frac{1}{2}\big[\cos\phi\sin\theta + i(-\cos\theta\sin\tau + \sin\phi\cos\tau\sin\theta)\big]\,.
\end{aligned}$$

Identity (2.4) then shows that (2.5b) holds.

2. The proof of this part is similar to the one in the first part, and the proof is left to the readers.

3. It is clear that $\mathrm{R}_z(\tau)$ maps $|\psi\rangle$ to the quantum state $\cos\frac{\theta}{2}|0\rangle + e^{i(\phi+\tau)}\sin\frac{\theta}{2}|1\rangle$. Therefore, the matrix representations of $\mathrm{R}_z(\tau)$ is given by

$$\mathrm{R}_z(\tau) = \begin{bmatrix} e^{-i\frac{\tau}{2}} & 0 \\ 0 & e^{i\frac{\tau}{2}} \end{bmatrix}. \qquad \square$$

For a $2\times 2$ matrix $A$ (with complex entries) satisfying $A^2 = \mathrm{I}$, one has

$$\begin{aligned} e^{iAx} &= \sum_{k=0}^{\infty} \frac{(iAx)^k}{k!} = \sum_{k=0}^{\infty} \frac{i^{2k}A^{2k}x^{2k}}{(2k)!} + \sum_{k=0}^{\infty} \frac{i^{2k+1}A^{2k+1}x^{2k+1}}{(2k+1)!} \\ &= \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k}}{(2k)!}\mathrm{I} + i\sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{(2k+1)!}A = \cos x\mathrm{I} + i\sin x A\,. \end{aligned}$$

Using the notation of exponential, we find the matrix representation of $\mathrm{R}_x(\tau)$, $\mathrm{R}_y(\tau)$ and $\mathrm{R}_z(\tau)$ given in (2.3) in fact can be expressed as

$$\mathrm{R}_x(\tau) = \exp\Big(\frac{-i\tau\mathrm{X}}{2}\Big), \qquad \mathrm{R}_y(\tau) = \exp\Big(\frac{-i\tau\mathrm{Y}}{2}\Big), \qquad \mathrm{R}_z(\tau) = \exp\Big(\frac{-i\tau\mathrm{Z}}{2}\Big). \tag{2.7}$$

Before proceeding, we note that for a unit vector $\boldsymbol{a} = (a_x, a_y, a_z)$ in $\mathbb{R}^3$,

$$\begin{aligned} &(a_x\mathrm{X} + a_y\mathrm{Y} + a_z\mathrm{Z})^2 \\ &\quad = a_x^2\mathrm{X}^2 + a_y^2\mathrm{Y}^2 + a_z^2\mathrm{Z}^2 + a_xa_y(\mathrm{XY}+\mathrm{YX}) + a_xa_z(\mathrm{XZ}+\mathrm{ZX}) + a_ya_x(\mathrm{YZ}+\mathrm{ZY}) \\ &\quad = (a_x^2 + a_y^2 + a_z^2)\mathrm{I} = \mathrm{I}\,. \end{aligned}$$

**Definition 2.10.** For a general unit vector $\boldsymbol{a} = (a_x, a_y, a_z)$ in $\mathbb{R}^3$, the rotation of an 1-qubit state with angle $\phi$ about an axis in direction $\boldsymbol{a}$, denoted by $R_{\boldsymbol{a}}(\phi)$, is a 1-qubit quantum gate given by

$$R_{\boldsymbol{a}}(\phi) = \exp\Big(-\frac{i\phi}{2}\big(a_x\mathrm{X} + a_y\mathrm{Y} + a_z\mathrm{Z}\big)\Big) = \cos\frac{\phi}{2}\mathrm{I} - i\sin\frac{\phi}{2}(a_x\mathrm{X} + a_y\mathrm{Y} + a_z\mathrm{Z})\,.$$

The matrix representation of $R_{\boldsymbol{a}}(\phi)$ is given by

$$R_{\boldsymbol{a}}(\phi) = \begin{bmatrix} \cos\frac{\phi}{2} - ia_z\sin\frac{\phi}{2} & -(a_y + ia_x)\sin\frac{\phi}{2} \\ (a_y - ia_x)\sin\frac{\phi}{2} & \cos\frac{\phi}{2} + ia_z\sin\frac{\phi}{2} \end{bmatrix}. \tag{2.8}$$

We will see gates acting on more than one qubit later.

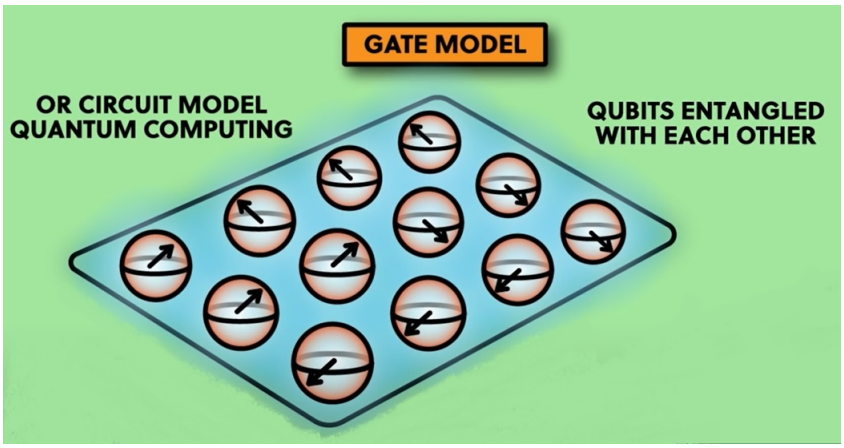



Figure 2.1: Gate model or circuit model of quantum computing - it consists of a lot of qubits, each qubit represents a digit of a number, and qubits are manipulated using quantum gates.

## 2.3 Quantum Registers

A quantum register is a system comprising multiple qubits. It is the quantum analog of the classical processor register. Quantum computers perform calculations by manipulating qubits within a quantum register.

**Remark 2.11.** There is a conceptual difference between the quantum and classical register. A classical register of $n$ bits refers to an array of $n$ flip flops (flip flops - 可儲存狀態 0 或 1 的電路), while a quantum register of $n$ qubits is merely a collection of $n$ qubits.

Classically, information is represented by finite chunks of bits - such as bytes - and multiples thereof. These are essentially words $(x_1, x_2, x_3, \cdots, x_n)$ built from the alphabet $\{0, 1\}$; that is, $x_\ell \in \{0, 1\}$ for all $1 \leqslant \ell \leqslant n$. Hence, we need $2^n$ classical storage configurations in order to represent all such words.

A classical two-bit word $(x_1, x_2)$ is an element of the set $\{0, 1\} \times \{0, 1\} = \{0, 1\}^2$, and classically we can represent the words 00, 01, 10, 11 by storing the first letter $x_1$ (the first bit or the highest bit) and the second letter $x_2$ (the second bit) accordingly. If we represent each of these bits quantum mechanically by qubits, we are dealing with a two-qubit quantum system composed of two quantum mechanical sub-systems. A two-qubit word in a two-quit quantum system is in superposition

$$\alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle\,, \quad \alpha_0, \alpha_1, \alpha_2, \alpha_3 \in \mathbb{C}, |\alpha_0|^2 + |\alpha_1|^2 + |\alpha_2|^2 + |\alpha_3|^2 = 1\,,$$

where $|x_1x_2\rangle$ denotes the state that the first qubit is in state $|x_1\rangle$ and the second qubit is in state $|x_2\rangle$.

More generally, a quantum register of $n$ qubits has $2^n$ basis states of the form $|b_1b_2\cdots b_n\rangle$. We will often abbreviate $0\cdots0$ to $0^n$ (so that $|0^n\rangle = |0\cdots0\rangle$). Since bitstrings of length $n$ can be viewed as numbers between 0 and $2^n-1$, we can also write the basis states as numbers $|0\rangle$, $|1\rangle$, $|2\rangle$, $\cdots$, $|2^n-1\rangle$. In other words, for $b = b_1b_2\cdots b_n \in \{0,1\}^n$ we often use $|b_12^{n-1} + b_22^{n-2} + \cdots + b_n\rangle$ to identify $|b_1b_2\cdots b_n\rangle$ ($b_1b_2\cdots b_n$ in binary equals $b_12^{n-1} + b_22^{n-2} + \cdots + b_n$ in decimal). A quantum register of $n$ qubits can be in any superposition

$$\alpha_0|0\rangle + \alpha_1|1\rangle + \cdots + \alpha_{2^n-1}|2^n-1\rangle = \sum_{j=0}^{2^n-1} \alpha_j|j\rangle\,, \qquad \sum_{j=0}^{2^n-1} |\alpha_j|^2 = 1\,.$$

The superposition above sometimes is also written as $\sum_{j\in\{0,1\}^n} \alpha_j|j\rangle$.

In an $n$-qubit quantum system, one can perform measurement on certain qubits. A measuement of $m$ qubits, where $m < n$, is a projective measurement, and the quantum register

$$\alpha_0|0\rangle + \alpha_1|1\rangle + \cdots + \alpha_{2^n-1}|2^n-1\rangle$$

under such a projective measurement collapses to another quantum register

$$\beta_0|0\rangle + \beta_1|1\rangle + \cdots + \beta_{2^n-1}|2^n-1\rangle\,,$$

where at most $2^{n-m}$ $\beta_j$'s are non-zero, and $\beta_0, \beta_1, \cdots, \beta_{2^n-1}$ are determined by the outcomes of the measurement, the exact position of the qubits on which measurement is performed, and $\alpha_0, \alpha_1, \cdots, \alpha_{2^n-1}$. For example, if we perform a measurement on the second qubit of the 3-qubit register

$$\alpha_0|000\rangle + \alpha_1|001\rangle + \alpha_2|010\rangle + \alpha_3|011\rangle + \alpha_4|100\rangle + \alpha_5|101\rangle + \alpha_6|110\rangle + \alpha_7|111\rangle$$

and obtain value 0, then the 3-qubit register above collapses to the quantum register

$$\frac{\alpha_0}{\|\boldsymbol{\alpha}\|}|000\rangle + \frac{\alpha_1}{\|\boldsymbol{\alpha}\|}|001\rangle + \frac{\alpha_4}{\|\boldsymbol{\alpha}\|}|100\rangle + \frac{\alpha_5}{\|\boldsymbol{\alpha}\|}|101\rangle$$

where $\|\boldsymbol{\alpha}\| = \sqrt{|\alpha_0|^2 + |\alpha_1|^2 + |\alpha_4|^2 + |\alpha_5|^2}$.

### 2.3.1 Tensor product of quantum registers - preview

Suppose that two single qubit states $|\psi_1\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ and $|\psi_2\rangle = \beta_0|0\rangle + \beta_1|1\rangle$ are given, and a quantum register of two qubits is formed from these two single qubits: the output of the first and the second qubit of the quantum register upon measurement follows the distribution given by states $|\psi_1\rangle$ and $|\psi_2\rangle$, respectively. This means that when measuring this particular quantum register, the first qubit outputs $|0\rangle$ or $|1\rangle$ with probability $|\alpha_1|^2$ or $|\beta_1|^2$, while the second qubit outputs respectively $|0\rangle$ or $|1\rangle$ with probability $|\alpha_2|^2$ or $|\beta_2|^2$, respectively. Therefore, measuring this quantum register of two qubits gives $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$ with probability $|\alpha_0\beta_0|^2$, $|\alpha_0\beta_1|^2$, $|\alpha_1\beta_0|^2$ and $|\alpha_1\beta_1|^2$, respectively. This motivates us to consider the quantum state of two qubits

$$|\psi\rangle = \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle\,.$$

We will write the quantum state $|\psi\rangle$ above as $|\psi_1\rangle\otimes|\psi_2\rangle$, called the tensor product of states $|\psi_1\rangle$ and $|\psi_2\rangle$. The detail explanation of the tensor product is given in Section 3.7.

In general, let $|\psi_1\rangle$ and $|\psi_2\rangle$ be two quantum states of $n$ qubits and $m$ qubits, respectively. The tensor product of $|\psi_1\rangle$ and $|\psi_2\rangle$ is a quantum state of $(n+m)$ qubits. Let us first consider the "continuous" case to illustrate the idea of the tensor product. Suppose that the states of two non-relativistic particles of the same mass $m$, labeled as particle 1 and particle 2, are described by Schrödinger equations

$$i\hbar\frac{\partial}{\partial t}\psi_1 = \Big(-\frac{\hbar}{2m}\Delta + V_1\Big)\psi_1 \qquad \text{in} \quad \mathbb{R}^n \times \{t > 0\}$$

and

$$i\hbar\frac{\partial}{\partial t}\psi_2 = \Big(-\frac{\hbar}{2m}\Delta + V_2\Big)\psi_2 \qquad \text{in} \quad \mathbb{R}^n \times \{t > 0\}\,,$$

respectively. Then at time $t$ the probability of the presence of particle 1 at location $x$ and particle 2 at location $y$ is given by $\big|\psi_1(x,t)\big|^2\big|\psi_2(y,t)\big|^2 = \big|\psi_1(x,t)\psi_2(y,t)\big|^2$. This motivates of considering the function $\psi(x,y,t) = \psi_1(x,t)\psi_2(y,t)$. This function $\psi$ satisfies

$$i\hbar\frac{\partial}{\partial t}\psi = \Big(-\frac{\hbar}{2m}\Delta + V\Big)\psi \qquad \text{in} \quad \mathbb{R}^n \times \mathbb{R}^n \times \{t > 0\}\,,$$

where $V(x,y,t) = V_1(x,t) + V_2(y,t)$ and

$$(\Delta\psi)(x,y,t) = (\Delta_x + \Delta_y)\psi(x,y,t) = \psi_2(y,t)\Delta_x\psi_1(x,t) + \psi_1(x,t)\Delta_y\psi_2(y,t)\,.$$

If there is no interference between the two particles (which is the case if $V_1$ and $V_2$ satisfy certain conditions), then the state of the "combined system" (meaning that we use $(x,y) \in$

$\mathbb{R}^n \times \mathbb{R}^n$ to write the position of these two particles) is described by the wave function $\psi$: the probability of the presence of the combined system at location $(x, y)$ at time $t$ is given by $\big|\psi(x,y,t)\big|^2 = \big|\psi_1(x,t)\big|^2\big|\psi_2(y,t)\big|^2$. In other words, the state of the combined system is simply the "product" (which is exactly the tensor product) of the individual states.

Now suppose the states of two qubits are given by $|\psi_1\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ and $|\psi_2\rangle = \beta_0|0\rangle + \beta_1|1\rangle$. Recall that this is a shorthand notation for the quantum states

$$\psi_1(x_1) = \begin{cases} \alpha_0 & \text{if } x_1 = 0\,, \\ \alpha_1 & \text{if } x_1 = 1\,, \end{cases} \qquad \text{and} \qquad \psi_2(x_2) = \begin{cases} \beta_0 & \text{if } x_2 = 0\,, \\ \beta_1 & \text{if } x_2 = 1\,, \end{cases}$$

Then the state of the combined system (which can be used to describe for random numbers $(0)_{10} = (00)_2$, $(1)_{10} = (01)_2$, $(2)_{10} = (10)_2$ and $(3)_{10} = (11)_2$, where $\psi_1$ is the state of the first bit and $\psi_2$ is the state of the second bit) is given by

$$\psi(x_1, x_2) \equiv \psi_1(x_1)\psi_2(x_2) = \begin{cases} \alpha_0\beta_0 & \text{if } (x_1, x_2) = (0,0)\,, \\ \alpha_0\beta_1 & \text{if } (x_1, x_2) = (0,1)\,, \\ \alpha_1\beta_0 & \text{if } (x_1, x_2) = (1,0)\,, \\ \alpha_1\beta_1 & \text{if } (x_1, x_2) = (1,1)\,, \end{cases}$$

which is abbreviated as

$$|\psi\rangle = \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle\,.$$

In general, if

$$|\psi_1\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle + \cdots + \alpha_{2^n-1}|2^n - 1\rangle$$

and

$$|\psi_2\rangle = \beta_0|0\rangle + \beta_1|1\rangle + \cdots + \beta_{2^m-1}|2^m - 1\rangle$$

are two quantum states, then

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle = \Big(\sum_{k=0}^{2^n-1} \alpha_k|k\rangle\Big) \otimes \Big(\sum_{\ell=0}^{2^m-1} \beta_\ell|\ell\rangle\Big) = \sum_{k=0}^{2^n-1}\sum_{\ell=0}^{2^m-1} \alpha_k\beta_\ell|k\rangle \otimes |\ell\rangle\,,$$

where by writing $k = (k_1k_2\cdots k_n)_2$ and $\ell = (\ell_1\ell_2\cdots\ell_m)_2$,

$$|k\rangle \otimes |\ell\rangle = |k_1k_2\cdots k_n\ell_1\ell_2\cdots\ell_m\rangle\,.$$

Sometimes $|\psi_1\rangle \otimes |\psi_2\rangle$ is written as $|\psi_1\rangle|\psi_2\rangle$.

### 2.3.2 Entanglement

An important property that deserves to be mentioned is entanglement, which refers to quantum correlations between different qubits. For instance, consider a 2-qubit register that is in the state

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

Such 2-qubit states are sometimes called EPR-pairs in honor of Einstein, Podolsky, and Rosen, who first examined such states and their seemingly paradoxical properties. Initially neither of the two qubits has a classical value $|0\rangle$ or $|1\rangle$. However, if we measure the first qubit and observe, say, a $|0\rangle$, then the whole state collapses to $|00\rangle$. Thus observing the first qubit immediately fixes also the second, unobserved qubit to a classical value. Since the two qubits that make up the register may be far apart, this example illustrates some of the non-local effects that quantum systems can exhibit. In general, a bipartite state $|\phi\rangle$ is called entangled if it cannot be written as a tensor product $|\phi_A\rangle \otimes |\phi_B\rangle$, where $|\phi_A\rangle$ lives in the first space and $|\phi_B\rangle$ lives in the second.

At this point, a comparison with classical probability distributions may be helpful. Suppose we have two probability spaces, $A$ and $B$, the first with $2^n$ possible outcomes, the second with $2^m$ possible outcomes. A distribution on the first space can be described by $2^n$ parameters (non-negative reals summing to 1; actually there are only $2^n-1$ degrees of freedom here) and a distribution on the second by $2^m$ parameters. Accordingly, a product distribution on the joint space can be described by $2^n + 2^m$ parameters. However, an arbitrary (non-product) distribution on the joint space takes $2^{n+m}$ numbers, since there are $2^{n+m}$ possible outcomes in total. Analogously, an $n$-qubit state $|\phi_A\rangle$ can be described by $2^n$ parameters (complex numbers whose squared moduli sum to 1), an $m$-qubit state $|\phi_B\rangle$ by $2^m$ parameters, and their tensor product $|\phi_A\rangle \otimes |\phi_B\rangle$ by $2^n + 2^m$ parameters. However, an arbitrary (possibly entangled) state in the joint space takes $2^{n+m}$ numbers, since it lives in a $2^{n+m}$-dimensional space. We see that the number of parameters required to describe quantum states is the same as the number of parameters needed to describe probability distributions. Also note the analogy between statistical independence of two random variables $A$ and $B$ and non-entanglement of the product state $|\phi_A\rangle \otimes |\phi_B\rangle$. However, despite the similarities between probabilities and amplitudes, quantum states are much more powerful than distributions, because amplitudes may have negative parts which can lead to interference effects. Amplitudes only become probabilities when we square them. The art of quantum

computing is to use these special properties for interesting computational purposes.

## 2.4 Quantum Circuits

A quantum circuit (also called quantum network or quantum gate array) generalizes the idea of classical circuit families, replacing the **AND**, **OR**, and **NOT** gates by elementary quantum gates. A quantum gate is a unitary transformation on a small (usually 1, 2, or 3) number of qubits. We saw a number of examples already in Section 2.2: the bitflip-gate X, the phaseflip gate Z, the Hadamard gate H. Mathematically, these gates can be composed by taking tensor products (if gates are applied in parallel to different parts of the register) and ordinary products (if gates are applied sequentially). Simple examples of such circuits of elementary gates are given in the next section.

For example, if we apply the Hadamard gate H to each bit in a register of $n$ zeroes, we obtain $\frac{1}{\sqrt{2^n}} \sum_{j\in\{0,1\}^n} |j\rangle$ which is a superposition of all $n$-bit strings. More generally, if we apply $\mathrm{H}^{\otimes n}$ to an initial state $|i\rangle$, with $i \in \{0,1\}^n$, we obtain

$$\mathrm{H}^{\otimes n}|i\rangle = \frac{1}{\sqrt{2^n}} \sum_{j\in\{0,1\}^n} (-1)^{i\bullet j}|j\rangle\,, \tag{2.9}$$

where $i \bullet j = \sum_{k=1}^{n} i_k j_k$ denotes the bitwise product of the $n$-bit strings $i, j \in \{0,1\}^n$. For instance,

$$\mathrm{H}^{\otimes 2}|01\rangle \equiv (\mathrm{H}|0\rangle) \otimes (\mathrm{H}|1\rangle) = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \frac{1}{2} \sum_{j\in\{0,1\}^2} (-1)^{01\bullet j}|j\rangle\,.$$

The $n$-fold Hadamard transform $\mathrm{H}^{\otimes n}$ will be very useful for all the quantum algorithms explained later.

Another important 1-qubit gate is the phase gate $\mathrm{R}_\phi$, which merely rotates the phase of the $|1\rangle$-state by an angle $\phi$:

$$\mathrm{R}_\phi|0\rangle = |0\rangle \qquad \text{and} \qquad \mathrm{R}_\phi|1\rangle = e^{i\phi}|1\rangle\,.$$

This corresponds to the unitary matrix

$$\mathrm{R}_\phi = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix}.$$

An example of a 2-qubit gate is the controlled-not gate **CNOT**. It negates the second bit of its input if the first bit is 1, and does nothing if first bit is 0:

$$\mathbf{CNOT}|ab\rangle = |a\rangle \otimes |a \oplus b\rangle \qquad \forall\, a, b \in \{0, 1\}\,.$$

In matrix form, it is

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

More generally, if $U$ is some 1-qubit gate, then the 2-qubit controlled-$U$ gate given by

$$|0b\rangle \mapsto |0b\rangle \quad \text{and} \quad |1b\rangle \mapsto |1\rangle \otimes U|b\rangle \qquad \forall\, b \in \{0, 1\}$$

corresponds to the following $4 \times 4$ unitary matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{11} & u_{12} \\ 0 & 0 & u_{21} & u_{22} \end{bmatrix}.$$

Adding another control register to **CNOT**, we get the 3-qubit Toffoli gate, also called controlled-controlled-not (CCNOT) gate. This negates the third bit of its input if both of the first two bits are 1 so that

$$\mathbf{CCNOT}|abc\rangle = |ab\rangle \otimes |ab \oplus c\rangle \qquad \forall\, a, b, c \in \{0, 1\}$$

or more precise,

$$\begin{aligned} &\mathbf{CCNOT}\big(\alpha_0|000\rangle + \alpha_1|001\rangle + \alpha_2|010\rangle + \alpha_3|011\rangle + \alpha_4|100\rangle + \alpha_5|101\rangle + \alpha_6|110\rangle + \alpha_7|111\rangle\big) \\ &\quad = \alpha_0|000\rangle + \alpha_1|001\rangle + \alpha_2|010\rangle + \alpha_3|011\rangle + \alpha_4|100\rangle + \alpha_5|101\rangle + \alpha_6|111\rangle + \alpha_7|110\rangle \end{aligned}$$

which shows that **CCNOT**relative to the basis

$$\big\{|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle\big\}$$

has the matrix form

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

The Toffoli gate is important because it is complete for classical reversible computation: any classical computation can be implemented by a circuit of Toffoli gates.

A quantum circuit is a finite directed acyclic graph of input nodes, gates, and output nodes. There are $n$ nodes that contain the input; in addition we may have some more input nodes that are initially $|0\rangle$ ("workspace"). The internal nodes of the quantum circuit are quantum gates that each operate on at most 2 qubits of the state. The gates in the circuit transform the initial state vector into a final state, which will generally be a superposition. We measure some dedicated output bits of this final state to (probabilistically) obtain an answer.

To draw such circuits, we typically let time progress from left to right: we start with the initial state on the left. Each qubit is pictured as a wire, and the circuit prescribes which gates are to be applied to which wires. Single-qubit gates like X and H just act on one wire, while multi-qubit gates such as the **CNOT** act on multiple wires simultaneously. When one qubit "controls" the application of a gate to another qubit, then the controlling wire is drawn with a dot linked vertically to the gate that is applied to the target qubit. This happens for instance with the **CNOT**, where the applied single-qubit gate is X (sometimes drawn as '$\oplus$'). Figure 2.1 gives a simple example on two qubits, initially in basis state $|00\rangle$: first apply the Hadamard gate H to the first qubit, then **CNOT** to both qubits (with the first qubit acting as the control), and then Z to the last qubit.

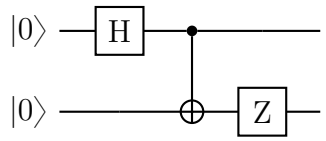



Figure 2.2: Simple circuit for turning $|00\rangle$ into an entangled state

Let $A \otimes B$ denote the map defined by $(A \otimes B)(|a\rangle \otimes |b\rangle) = (A|a\rangle) \otimes (B|b\rangle)$. Then

$$\begin{aligned} |00\rangle \overset{\mathrm{H}\otimes\mathrm{I}}{\mapsto} \mathrm{H}|0\rangle \otimes \mathrm{I}|0\rangle = \frac{1}{\sqrt{2}}\big(|00\rangle + |10\rangle\big) \overset{\textbf{CNOT}}{\mapsto} \frac{1}{\sqrt{2}}\big(|00\rangle + |11\rangle\big) \\ \overset{\mathrm{I}\otimes\mathrm{Z}}{\mapsto} \frac{1}{\sqrt{2}}\big(\mathrm{I}|0\rangle \otimes \mathrm{Z}|0\rangle + \mathrm{I}|1\rangle \otimes \mathrm{Z}|1\rangle\big) = \frac{1}{\sqrt{2}}\big(|00\rangle - |11\rangle\big)\,. \end{aligned} \tag{2.10}$$

Therefore, the resulting state of the circuit given in Figure 2.2 is $\frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$.

Note that if we have a circuit for unitary $U$, it is very easy to find a circuit for the inverse $U^{-1}$ with the same complexity: just reverse the order of the gates, and take the inverse of each gate. For example, if $U = U_1U_2U_3$, then $U^{-1} = U_3^{-1}U_2^{-1}U_1^{-1}$.

**Example 2.12.** One possible implementation of a 2-bit full adder (using **CNOT** gates and TOFFOLI gates):

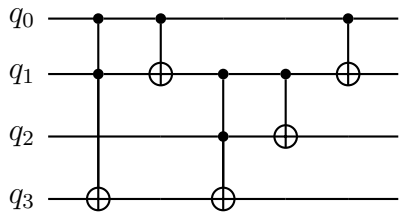



Figure 2.3: Circuit diagram of a quantum full adder

where the inputs are $q_0 = A$, $q_1 = B$, $q_2 = \mathrm{C_{in}}$, and the ouputs are $q_0 = A$, $q_1 = B$, $q_2 = \mathrm{Sum_{out}}$, $q_3 = \mathrm{C_{out}}$.

The validity of that the quantum circuit above is indeed a full adder can be verified by the following truth table:

| **INPUT** | | | | **OUTPUT** | | | |
|---|---|---|---|---|---|---|---|
| $q_3$ | $q_2$ $\mathrm{C_{in}}$ | $q_1$ B | $q_0$ A | $q_3$ $\mathrm{C_{out}}$ | $q_2$ S | $q_1$ B | $q_0$ A |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

### 2.4.1 Quantum Teleportation

As an example of the use of elementary gates, we will explain teleportation. Suppose there are two parties, Alice and Bob. Alice has a qubit $\alpha_0|0\rangle + \alpha_1|1\rangle$ that she wants to send to Bob via a classical channel. Without further resources this would be impossible, but Alice also shares an EPR-pair

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

with Bob (say Alice holds the first qubit and Bob the second). Initially, their joint state is

$$\big(\alpha_0|0\rangle + \alpha_1|1\rangle\big) \otimes \frac{1}{\sqrt{2}}\big(|00\rangle + |11\rangle\big) = \frac{\alpha_0}{\sqrt{2}}\big(|000\rangle + |011\rangle\big) + \frac{\alpha_1}{\sqrt{2}}\big(|100\rangle + |111\rangle\big)\,.$$

The first two qubits belong to Alice, the third to Bob. Alice performs a CNOT on her two qubits to obtain

$$\frac{\alpha_0}{\sqrt{2}}\big(|000\rangle + |011\rangle\big) + \frac{\alpha_1}{\sqrt{2}}\big(|110\rangle + |101\rangle\big)$$

and then a Hadamard transform on her first qubit so that their joint state now becomes

$$\begin{aligned}
&\frac{\alpha_0}{2}\Big[\big(|0\rangle + |1\rangle\big)\otimes\big(|00\rangle + |11\rangle\big)\Big] + \frac{\alpha_1}{2}\Big[\big(|0\rangle - |1\rangle\big)\otimes\big(|10\rangle + |01\rangle\big)\Big] \\
&\quad = \frac{\alpha_0}{2}\big(|000\rangle + |011\rangle + |100\rangle + |111\rangle\big) + \frac{\alpha_1}{2}\big(|010\rangle + |001\rangle - |110\rangle - |101\rangle\big) \\
&\quad = \frac{1}{2}|00\rangle\otimes\big(\alpha_0|0\rangle + \alpha_1|1\rangle\big) + \frac{1}{2}|01\rangle\otimes\big(\alpha_0|1\rangle + \alpha_1|0\rangle\big) + \frac{1}{2}|10\rangle\otimes\big(\alpha_0|0\rangle - \alpha_1|1\rangle\big) \\
&\qquad + \frac{1}{2}|11\rangle\otimes\big(\alpha_0|1\rangle - \alpha_1|0\rangle\big)\,.
\end{aligned}$$

Alice then measures her two qubits in the computational basis and sends the result $b_1b_2$, a 2 random classical bits, to Bob over a classical channel. In order to recover Alice's qubit, Bob applies the transformation $\mathrm{Z}^{b_1}\mathrm{X}^{b_2}$, where X is the bitflip-gate and Z is the phaseflip gate given by (2.2), to the qubit he has now (once Alice makes a measurement, the 3 qubit Bob has collapses to a qubit). For example, if Alice sent 11 to Bob over a classical channel, Bob then applies ZX to the qubit $\alpha_0|1\rangle - \alpha_1|0\rangle$ (which is the qubit Bob has now since Alice's two qubits has been measured) and obtain $\alpha_0|0\rangle + \alpha_1|1\rangle$ which is the qubit Alice has originally. In fact, if Alice's qubit had been entangled with other qubits, then teleportation preserves this entanglement: Bob then receives a qubit that is entangled in the same way as Alice's original qubit was.

Note that the qubit on Alice's side has been destroyed: teleporting moves a qubit from $A$ to $B$, rather than copying it. In fact, copying an unknown qubit is impossible. This can be seen as follows. Suppose $C$ were a 1-qubit copier; that is, $C|\phi\rangle|0\rangle = |\phi\rangle|\phi\rangle$ for every qubit $|\phi\rangle$. In particular, $C|0\rangle|0\rangle = |0\rangle|0\rangle$ and $C|1\rangle|0\rangle = |1\rangle|1\rangle$. But then $C$ would not copy $|\phi\rangle = \mathrm{H}|0\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big)$ correctly, since by linearity

$$C|\phi\rangle|0\rangle = \frac{1}{\sqrt{2}}\big(C|0\rangle|0\rangle + C|1\rangle|0\rangle\big) = \frac{1}{\sqrt{2}}\big(|0\rangle|0\rangle + |1\rangle|1\rangle\big) \neq |\phi\rangle|\phi\rangle\,.$$

**Remark 2.13.** The fact that copying an unknown qubit is impossible implies that not all the Boolean function can be implemented by current quantum computers. The lack of the ability of performing all Boolean functions will put a lot of constraints to the use of quantum computers.

## 2.5 Universality of Various Sets of Elementary Gates

Similar to Definition 1.4 and 1.11, we have the following

**Definition 2.14.** Let $\{U_1, \cdots, U_k\}$ be a collection of quantum gates, where each $U_j$ is an $n_j$-qubit quantum gate for some $n_j \in \mathbb{N}$. The collection of all quantum gates that can be constructed from $U_1$, $U_2$, $\cdots$, $U_k$, denoted by $\mathscr{F}[U_1, \cdots, U_k]$, is the set satisfying the following construction rules:

1. For any $1 \leqslant j \leqslant k$, $U_j \in \mathscr{F}[U_1, \cdots, U_k]$.
2. For any $n \in \mathbb{N}$, $\mathbf{1}^{\otimes n} \in \mathscr{F}[U_1, \cdots, U_k]$, where $\mathbf{1}$ denotes the identity gate.
3. For any $n$-qubit quantum gates $V_1, V_2$, we have
$$V_1, V_2 \in \mathscr{F}[U_1, \cdots, U_k] \quad \Rightarrow \quad V_1 V_2 \in \mathscr{F}[U_1, \cdots, U_k]\,.$$
4. For any two quantum gates $V_1, V_2$, we have
$$V_1, V_2 \in \mathscr{F}[U_1, \cdots, U_k] \quad \Rightarrow \quad V_1 \otimes V_2 \in \mathscr{F}[U_1, \cdots, U_k]\,.$$

A collection of quantum gates $\mathcal{U} = \{U_1, \cdots, U_k\}$ is called universal if any quantum gate $U$ can be constructed with gates from $\mathcal{U}$; that is, for every quantum gate $U$, $U \in \mathscr{F}[U_1, \cdots, U_k]$.

Similar to Proposition 1.10, we have the following

**Proposition 2.15.** *For quantum gates $V_1, \cdots, V_\ell$, $U_1, \cdots, U_k$, we have*
$$V_1, \cdots, V_\ell \in \mathscr{F}[U_1, \cdots, U_k] \quad \Rightarrow \quad \mathscr{F}[V_1, \cdots, V_\ell] \subseteq \mathscr{F}[U_1, \cdots, U_k]\,.$$
*In particular, $\mathscr{F}\big[\mathscr{F}[U_1, \cdots, U_k]\big] = \mathscr{F}[U_1, \cdots, U_k]$.*

Which set of elementary gates should we allow? There are several reasonable choices.

(1) The set of all 1-qubit operations together with the 2-qubit **CNOT** gate is universal, meaning that any other unitary transformation can be built from these gates.

Allowing all 1-qubit gates is not very realistic from an implementational point of view, as there are uncountably many of them. However, the model is usually restricted, only allowing a small finite set of 1-qubit gates from which all other 1-qubit gates can be efficiently approximated.

**Theorem 2.16** (Solovay-Kitaev)**.** *Let $\mathcal{G}$ be a finite set of elements in* SU(2) *containing its own inverses and such that the group $\langle\mathcal{G}\rangle$ they generate is dense in* SU(2)*. There exists $c > 0$ such that for any $\varepsilon > 0$ and $U \in \mathrm{SU}(2)$, there is a sequence $S$ of gates from $\mathcal{G}$ of length $\mathcal{O}(\log^c(1/\varepsilon))$ such that $\|S - U\| \leqslant \varepsilon$.*

(2) The set consisting of **CNOT**, Hadamard, and the phase-gate $\mathrm{R}_{\frac{\pi}{4}}$ is universal in the sense of approximation, meaning that any other unitary can be arbitrarily well approximated using circuits of only these gates. The Solovay-Kitaev Theorem says that this approximation is quite efficient: we can approximate any gate on 1 or 2 qubits up to error $\varepsilon$ using polylog$(1/\varepsilon)$ gates from our small set; in particular, simulating arbitrary gates up to exponentially small error costs only a polynomial overhead.

(3) The set of Hadamard H, **CNOT**, $\mathrm{R}_y(\tau)$, $\mathrm{R}_z(\tau)$ (for all $\tau \in \mathbb{R}$) and **SWAP** is universal.

It is often convenient to restrict to real numbers and use an even smaller set of gates:

(4) The set of Hadamard and Toffoli (**CCNOT**) is universal for all unitaries with real entries in the sense of approximation, meaning that any unitary with only real entries can be arbitrarily well approximated using circuits of only these gates (again the Solovay-Kitaev Theorem says that this simulation can be done efficiently).

## 2.6 Quantum Parallelism

One uniquely quantum-mechanical effect that we can use for building quantum algorithms is quantum parallelism. Suppose we can build a quantum circuit to represent a boolean function $f : \{0,1\}^n \to \{0,1\}^m$. Then we can build a quantum circuit U that maps $|x\rangle|0\rangle \to |x\rangle|f(x)\rangle$ for every $x \in \{0,1\}^n$. Now suppose we apply U to a superposition of all inputs $x$:

$$\mathrm{U}\left(\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle|0\rangle\right) = \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle|f(x)\rangle\,.$$

We applied U just once, but the final superposition contains $f(x)$ for all $2^n$ input values $x$! However, by itself this is not very useful and does not give more than classical randomization, since observing the final superposition will give just one random $|x\rangle|f(x)\rangle$ and all other information will be lost. As we will see below, quantum parallelism needs to be combined with the effects of interference and entanglement in order to get something that is better than classical.

## 2.7 The Early Algorithms

The two best-known successes of quantum algorithm so far are Shor's factoring algorithm from 1994 and Grover's search algorithm from 1996, which will be explained in later chapters. In this section we describe some of the earlier quantum algorithms that preceded Shor's and Grover's. Virtually all quantum algorithms work with queries in some form or other. We will explain this model here. It may look contrived at first, but eventually will lead smoothly to Shor's and Grover's algorithm.

To explain the query setting, consider an $N$-bit input $x = (x_0, x_1, \cdots, x_{N-1}) \in \{0,1\}^N$. Usually we will have $N = 2^n$, so that we can address bit $x_i$ using an $n$-bit index $i \in \{0,1\}^n$. One can think of the input as an $N$-bit memory which we can access at any point of our choice (a Random Access Memory). A memory access is via a so-called "black-box", which is equipped to output the bit $x_i$ on input $i$. As a quantum operation, this would be the following unitary mapping on $n+1$ qubits:

$$\mathrm{O}_x : |i\rangle|0\rangle \mapsto |i\rangle|x_i\rangle\,.$$

The first $n$ qubits of the state are called the address bits (or address register), while the $(n+1)$-th qubit is called the target bit. Since this operation must be unitary, we also have to specify what happens if the initial value of the target bit is 1. Therefore we actually let $\mathrm{O}_x$ be the following unitary transformation:

$$\mathrm{O}_x : |i\rangle|b\rangle \mapsto |i\rangle|b \oplus x_i\rangle\,,$$

here $i \in \{0,1\}^n$, $b \in \{0,1\}$, and $\oplus$ denotes exclusive-or (addition modulo 2). In matrix representation, $\mathrm{O}_x$ is now a permutation matrix and hence unitary. Note also that a quantum computer can apply $\mathrm{O}_x$ on a superposition of various $i$, something a classical computer cannot do. One application of this black-box is called a query, and counting the required number of queries to compute this or that function of $x$ is something we will do a lot in the first half of these notes.

Given the ability to make a query of the above type, we can also make a query of the form $|i\rangle \mapsto (-1)^{x_i}|i\rangle$ by setting the target bit to the state $|-\rangle \equiv \mathrm{H}|1\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big)$:

$$\mathrm{O}_x(|i\rangle|-\rangle) = |i\rangle\frac{1}{\sqrt{2}}\big(|x_i\rangle - |1-x_i\rangle\big) = (-1)^{x_i}|i\rangle|-\rangle\,.$$

This $\pm$-kind of query puts the output variable in the phase of the state: if $x_i$ is 1 then we get a $-1$ in the phase of basis state $|i\rangle$; if $x_i = 0$ then nothing happens to $|i\rangle$. This "phase-oracle"

is sometimes more convenient than the standard type of query. We sometimes denote the corresponding $n$-qubit unitary transformation (ignoring the last qubit $|-\rangle$) by $\mathrm{O}_{x,\pm}$.

### 2.7.1 Deutsch-Jozsa

**Deutsch-Jozsa problem**: For $N = 2^n$, we are given $x \in \{0,1\}^N$ such that either

1. all $x_i$ have the same value ("constant"), or
2. $N/2$ of the $x_i$ are 0 and $N/2$ are 1 ("balanced").

The goal is to find out whether $x$ is constant or balanced.

The algorithm of Deutsch and Jozsa is as follows. We start in the $n$-qubit zero state $|0^n\rangle$, apply a Hadamard transform to each qubit, apply a query (in its $\pm$-form), apply another Hadamard to each qubit, and then measure the final state. As a unitary transformation, the algorithm would be $\mathrm{H}^{\otimes n}\mathrm{O}_{\pm}\mathrm{H}^{\otimes n}$. We have drawn the corresponding quantum circuit in Figure 2.4 (where time progresses from left to right).

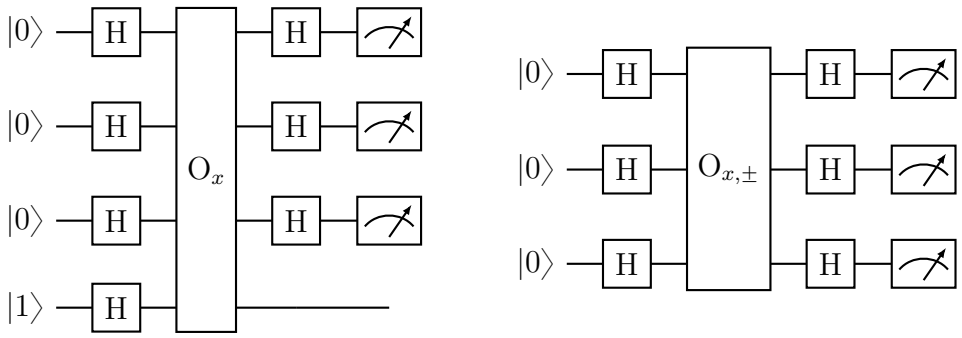



Figure 2.4: The Deutsch-Jozsa algorithm for $n = 3$. Left - the usual way of drawing the circuit (a circuit with the target qubit). Right - Only care about the first $n$ qubits.

Let us follow the state through these operations. Initially we have the state $|0^n\rangle$. Using (2.9), after the first Hadamard transforms we have obtained the uniform superposition of all $i$:

$$\frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} |i\rangle .$$

The $\mathrm{O}_{\pm}$-query turns this into

$$\frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} (-1)^{x_i} |i\rangle .$$

Applying the second batch of Hadamards gives (again by Equation (2.9)) the final superposition

$$\frac{1}{2^n}\sum_{i\in\{0,1\}^n}(-1)^{x_i}\sum_{j\in\{0,1\}^n}(-1)^{i\bullet j}|j\rangle\,,$$

where $i\bullet j=\sum_{k=1}^{n}i_kj_k$ is the bitwise dot product of $i$ and $j$ as before. Since $i\bullet 0^n=0$ for all $i\in\{0,1\}^n$, we see that the amplitude of the $|0^n\rangle$-state in the final superposition is

$$\frac{1}{2^n}\sum_{i\in\{0,1\}^n}(-1)^{x_i}=\begin{cases}1 & \text{if } x_i=0 \text{ for all } i\,,\\ -1 & \text{if } x_i=1 \text{ for all } i\,,\\ 0 & \text{if } x \text{ is balanced}\,.\end{cases}$$

Hence the final observation will yield $|0^n\rangle$ if $x$ is constant and will yield some other state if $x$ is balanced. Accordingly, the Deutsch-Jozsa problem can be solved with certainty using only 1 quantum query and $\mathcal{O}(n)$ other operations (the original solution of Deutsch and Jozsa used 2 queries, the 1-query solution is from [24]). In contrast, it is easy to see that any classical deterministic algorithm needs $N/2+1$ queries in the worst case scenario: if it has made only $N/2$ queries and seen only 0s, the correct output is still undetermined. However, a classical algorithm can solve this problem efficiently if we allow a small error probability: just query $x$ at two random positions, output "constant" if those bits are the same and "balanced" if they are different. This algorithm outputs the correct answer with probability 1 if $x$ is constant and outputs the correct answer with probability $1/2$ if $x$ is balanced. Thus the quantum-classical separation of this problem only holds if we consider algorithms without error probability.

**Remark 2.17.** In a lot of literatures, the Deutsch-Jozsa problem is formulated as: Let $f:\{0,1\}^n\to\{0,1\}$ satisfy either $f$ is a constant function or $\#f^{-1}(\{0\})=\#f^{-1}(\{1\})=2^{n-1}$ (such $f$ is said to be balanced). Determine if $f$ is constant or balanced. In such a case, the $\mathrm{O}_x$ operator is usually denoted by $U_f$, and the quantum circuit for the Deutsch-Jozsa algorithm is usually drawn as

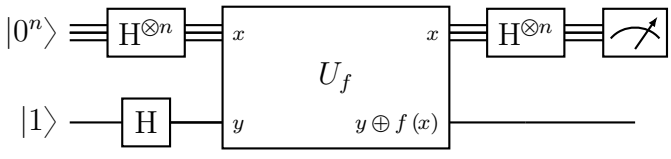



Figure 2.5: Another way of drawing the quantum circuit for the Deutsch-Jozsa algorithm

**Remark 2.18.** In general it is not easy to construct a quantum circuit for the oracle $U_f$; however, for some specific $f$ a quantum implementation of $U_f$ is possible. For example, let $f : \{0,1\}^n \to \{0,1\}$ be given by $f(x) = x_n$ if $x = (x_1, \cdots, x_n)$; that is, the value of $f$ is identical to the lowest digits of the input. Then $U_f = \mathbf{I}_{n-1} \otimes \mathbf{CNOT}$, where $\mathbf{I}_{n-1}$ is the identity map on $(n-1)$ qubit system, since

$$\begin{aligned}(\mathbf{I}_{n-1} \otimes \mathbf{CNOT})(|x\rangle|y\rangle) &= (\mathbf{I}_{n-1} \otimes \mathbf{CNOT})(|x_1 \cdots x_{n-1}x_n\rangle|y\rangle) \\ &= (\mathbf{I}_{n-1} \otimes \mathbf{CNOT})(|x_1 \cdots x_{n-1}\rangle|x_n y\rangle) = \big(\mathbf{I}_{n-1}|x_1 \cdots x_{n-1}\rangle\big) \otimes \big(\mathbf{CNOT}(|x_n\rangle|y\rangle)\big) \\ &= |x_1 \cdots x_{n-1}\rangle|x_n\rangle|y \oplus x_n\rangle = |x_1 \cdots x_n\rangle|y \oplus x_n\rangle = |x\rangle|y \oplus f(x)\rangle\,.\end{aligned}$$

Therefore, $U_f$ can be implemented by the following quantum circuit

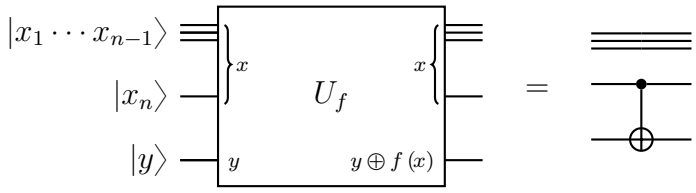



Figure 2.6: A quantum circuit for $U_f$ with $f(x_1, \cdots, x_n) = x_n$

### 2.7.2 Bernstein-Vazirani

**Bernstein-Vazirani problem**: For $N = 2^n$, we are given $x \in \{0,1\}^N$ with the property that there is some unknown $a \in \{0,1\}^n$ such that $x_i = (i \bullet a) \bmod 2$. The goal is to find $a$.

The Bernstein-Vazirani algorithm is exactly the same as the Deutsch-Jozsa algorithm, but now the final observation miraculously yields $a$. Since $(-1)^{x_i} = (-1)^{(i \bullet a) \bmod 2} = (-1)^{i \bullet a}$, we can write the state obtained after the query as:

$$\frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} (-1)^{x_i} |i\rangle = \frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} (-1)^{i \bullet a} |i\rangle\,.$$

Applying a Hadamard to each qubit will turn this into the classical state $|a\rangle$ and hence solves the problem with 1 query and $\mathcal{O}(n)$ other operations. In contrast, any classical algorithm (even a randomized one with small error probability) needs to ask $n$ queries for information-theoretic reasons: the final answer consists of $n$ bits and one classical query gives at most 1 bit of information. Bernstein and Vazirani also defined a recursive version of this problem, which can be solved exactly by a quantum algorithm in poly$(n)$ steps, but for which any classical randomized algorithm needs $n^{\Omega(\log n)}$ steps.

# Chapter 3

# Mathematical Backgrounds

## 3.1 Vector Spaces and Linear Maps

### 3.1.1 Vector Spaces

**Definition 3.1.** A ***vector space*** $\mathbb{V}$ over a scalar field $\mathbb{F}$ is a set of elements called vectors, with given operations of vector addition $+ : \mathbb{V} \times \mathbb{V} \to \mathbb{V}$ and scalar multiplication $\cdot : \mathbb{F} \times \mathbb{V} \to \mathbb{V}$ such that

1. $\boldsymbol{v} + \boldsymbol{w} = \boldsymbol{w} + \boldsymbol{v}$ for all $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{V}$.
2. $(\boldsymbol{v} + \boldsymbol{w}) + \boldsymbol{u} = \boldsymbol{v} + (\boldsymbol{u} + \boldsymbol{w})$ for all $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in \mathbb{V}$.
3. there exists $\mathbf{0}$, the zero vector, such that $\boldsymbol{v} + \mathbf{0} = \boldsymbol{v}$ for all $\boldsymbol{v} \in \mathbb{V}$.
4. for each $\boldsymbol{v} \in \mathbb{V}$ there exists $\boldsymbol{w} \in \mathbb{V}$ such that $\boldsymbol{v} + \boldsymbol{w} = \mathbf{0}$.
5. $\lambda \cdot (\boldsymbol{v} + \boldsymbol{w}) = \lambda \cdot \boldsymbol{v} + \lambda \cdot \boldsymbol{w}$ for all $\lambda \in \mathbb{F}$ and $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{V}$.
6. $(\lambda + \mu) \cdot \boldsymbol{v} = \lambda \cdot \boldsymbol{v} + \mu \cdot \boldsymbol{v}$ for all $\lambda, \mu \in \mathbb{F}$ and $\boldsymbol{v} \in \mathbb{V}$.
7. $(\lambda \cdot \mu) \cdot \boldsymbol{v} = \lambda \cdot (\mu \cdot \boldsymbol{v})$ for all $\lambda, \mu \in \mathbb{F}$ and $\boldsymbol{v} \in \mathbb{V}$.
8. $1 \cdot \boldsymbol{v} = \boldsymbol{v}$ for all $\boldsymbol{v} \in \mathbb{V}$.

**Remark 3.2.** In property 4 of the definition above, it is easy to see that for each $\boldsymbol{v}$, there is only one vector $\boldsymbol{w}$ such that $\boldsymbol{v} + \boldsymbol{w} = \mathbf{0}$. We often denote this $\boldsymbol{w}$ by $-\boldsymbol{v}$, and the vector substraction $- : \mathbb{V} \times \mathbb{V} \to \mathbb{V}$ is then defined (or understood) as $\boldsymbol{v} - \boldsymbol{w} = \boldsymbol{v} + (-\boldsymbol{w})$.

**Example 3.3.** Let $\mathbb{F}$ be a scalar field. The space $\mathbb{F}^n$ is the collection of $n$-tuple $\boldsymbol{v} = (\mathrm{v}_1, \mathrm{v}_2, \cdots, \mathrm{v}_n)$ with $\mathrm{v}_i \in \mathbb{F}$ with addition $+$ and scalar multiplication $\cdot$ defined by

$$(\mathrm{v}_1, \cdots, \mathrm{v}_n) + (\mathrm{w}_1, \cdots, \mathrm{w}_n) \equiv (\mathrm{v}_1 + \mathrm{w}_1, \cdots, \mathrm{v}_n + \mathrm{w}_n)\,,$$
$$\alpha(\mathrm{v}_1, \cdots, \mathrm{v}_n) \equiv (\alpha\mathrm{v}_1, \cdots, \alpha\mathrm{v}_n)\,.$$

Then $\mathbb{F}^n$ is a vector space over $\mathbb{F}$.

**Example 3.4.** Let $\mathbb{F}$ be a scalar field. The collection of $m \times n$ matrices with entries in $\mathbb{F}$ is denoted by $\mathcal{M}(m, n; \mathbb{F})$ or $\mathbb{F}^{m\times n}$; that is, $A \in \mathcal{M}(m, n; \mathbb{F})$ if and only if $A = [a_{ij}]_{1\leqslant i\leqslant m, 1\leqslant j\leqslant n}$ for some $a_{ij} \in \mathbb{F}$. Define the addition $+$ and scalar multiplication $\cdot$ on $\mathcal{M}(m, n; \mathbb{F})$ by

$$A + B = [a_{ij} + b_{ij}] \qquad \text{if} \quad A = [a_{ij}] \text{ and } B = [b_{ij}]$$

and

$$c \cdot A = [c \cdot a_{ij}] \qquad \text{if} \quad A = [a_{ij}]\,.$$

Then $(\mathcal{M}(m, n; \mathbb{F}), +, \cdot)$ is a vector space over $\mathbb{F}$.

**Example 3.5.** Let $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$, and $\mathbb{V}$ be the collection of all real-valued continuous functions on $[0, 1]$. The vector addition $+$ and scalar multiplication $\cdot$ is defined by

$$(f + g)(x) = f(x) + g(x) \qquad \forall\, f, g \in \mathbb{V}\,,$$
$$(\alpha \cdot f)(x) = \alpha f(x) \qquad \forall\, f \in \mathbb{V}, \alpha \in \mathbb{F}\,.$$

Then $\mathbb{V}$ is a vector space over $\mathbb{F}$, and is denoted by $\mathscr{C}([0, 1]; \mathbb{F})$. When the scalar field under consideration is clear, we simply use $\mathscr{C}([0, 1])$ to denote this vector space.

**Definition 3.6** (Vector subspace)**.** Let $\mathbb{V}$ be a vector space over scalar field $\mathbb{F}$. A subset $\mathbb{W} \subseteq \mathbb{V}$ is called a vector subspace of $\mathbb{V}$ if itself is a vector space over $\mathbb{F}$.

**Definition 3.7.** Let $\mathbb{V}$ be a vector space over a scalar field $\mathbb{F}$. $k$ vectors $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, $\cdots$, $\boldsymbol{v}_k$ in $\mathbb{V}$ is said to be ***linearly dependent*** if there exists $(\alpha_1, \cdots, \alpha_k) \subseteq \mathbb{F}^k$, $(\alpha_1, \cdots, \alpha_k) \neq \mathbf{0}$ such that $\alpha_1\boldsymbol{v}_1 + \alpha_2\boldsymbol{v}_2 + \cdots + \alpha_k\boldsymbol{v}_k = \mathbf{0}$. $k$ vectors $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, $\cdots$, $\boldsymbol{v}_k$ in $\mathbb{V}$ is said to be ***linearly independent*** if they are not linearly dependent. In other words, $\{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_k\}$ are linearly independent if

$$\alpha_1\boldsymbol{v}_1 + \alpha_2\boldsymbol{v}_2 + \cdots + \alpha_k\boldsymbol{v}_k = \mathbf{0} \quad \Rightarrow \quad \alpha_1 = \alpha_2 = \cdots = \alpha_k = 0\,.$$

**Example 3.8.** The $k$ vectors $\{1, x, x^2, \cdots, x^{k-1}\}$ are linearly independent in $\mathscr{C}([0,1])$ for all $k \in \mathbb{N}$.

**Definition 3.9.** The ***dimension*** of a vector space $\mathbb{V}$ is the number of maximum linearly independent set in $\mathbb{V}$, and in such case $\mathbb{V}$ is called an $n$-dimensional vector space, where $n$ is the dimension of $\mathbb{V}$. If for every number $n \in \mathbb{N}$ there exists $n$ linearly independent vectors in $\mathbb{V}$, the vector space $\mathbb{V}$ is said to be infinitely dimensional.

**Example 3.10.** The space $\mathbb{F}^n$ is $n$-dimensional, and $\mathscr{C}([0,1])$ is infinitely dimensional (since $1, x, \cdots, x^{n-1}$ are $n$ linearly independent vectors in $\mathscr{C}([0,1])$).

**Definition 3.11** (Basis)**.** Let $\mathbb{V}$ be a vector space over $\mathbb{F}$. A collection of vectors $\{\boldsymbol{v}_i\}_{i\in\mathcal{I}}$ in $\mathbb{V}$ is called a ***basis*** of $\mathbb{V}$ if for every $\boldsymbol{v} \in \mathbb{V}$, there exists a unique $\{\alpha_i\}_{i\in\mathcal{I}} \subseteq \mathbb{F}$ such that

$$\boldsymbol{v} = \sum_{\alpha\in\mathcal{I}} \alpha_i \boldsymbol{v}_i \,.$$

For a given basis $\mathcal{B} = \{\boldsymbol{v}_i\}_{i\in\mathcal{I}}$, the coefficients $\{\alpha_i\}_{i\in\mathcal{I}}$ given in the above relation is denoted by $[\boldsymbol{v}]_{\mathcal{B}}$.

**Example 3.12** (Standard Basis of $\mathbb{F}^n$)**.** Let $\mathbf{e}_i = (0, , \cdots, 0, 1, 0, \cdots, 0)$, where 1 locates at the $i$-th slot. Then the collection $\{\mathbf{e}_i\}_{i=1}^n$ is a basis of the vector space $\mathbb{F}^n$ over $\mathbb{F}$ since

$$(\alpha_1, \cdots, \alpha_n) = \sum_{i=1}^{n} \alpha_i \mathbf{e}_i \qquad \forall\, \alpha_i \in \mathbb{F}.$$

The collection $\{\mathbf{e}_i\}_{i=1}^n$ is called the ***standard basis*** of $\mathbb{F}^n$.

**Example 3.13.** Even though $\left\{1, x, \cdots, x^k, \cdots\right\}$ is a set of linearly independent vectors, it is not a basis of $\mathscr{C}([0,1])$. However, let $\mathscr{P}([0,1])$ be the collection of polynomials defined on $[0,1]$. Then $\mathscr{P}([0,1])$ is still a vector space, and $\left\{1, x, \cdots, x^k, \cdots\right\}$ is a basis of $\mathscr{P}([0,1])$.

### 3.1.2 Linear maps and their matrix representation

**Definition 3.14.** Let $\mathbb{V}, \mathbb{W}$ be vector spaces over a common scalar field $\mathbb{F}$. A map $L$ from $\mathbb{V}$ to $\mathbb{W}$ is said to be ***linear*** if $L(c\boldsymbol{v}_1 + \boldsymbol{v}_2) = cL(\boldsymbol{v}_1) + L(\boldsymbol{v}_2)$ for all $\boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{V}$ and $c \in \mathbb{F}$. We often write $L\boldsymbol{v}$ instead of $L(\boldsymbol{v})$, and the collection of all linear maps from $\mathbb{V}$ to $\mathbb{W}$ is denoted by $\mathscr{L}(\mathbb{V}; \mathbb{W})$. We also write $\mathscr{L}(\mathbb{V})$ instead of $\mathscr{L}(\mathbb{V}; \mathbb{V})$ if $\mathbb{W} = \mathbb{V}$. An element in $\mathscr{L}(\mathbb{V}; \mathbb{F})$ is called a linear functional on $\mathbb{V}$.

**Proposition 3.15.** *Let $\mathbb{V}$ and $\mathbb{W}$ be vector spaces over a common scalar field $\mathbb{F}$. Then $\mathscr{L}(\mathbb{V};\mathbb{W})$ is a vector space over $\mathbb{F}$.*

**Example 3.16.** Let $\mathbb{F}$ be a scalar field, and $A = [a_{ij}]_{1\leqslant i\leqslant m, 1\leqslant j\leqslant n} \in \mathcal{M}(m,n;\mathbb{F})$ be an $m\times n$ matrix. Define a vector-valued function $L:\mathbb{F}^n\to\mathbb{F}^m$ by

$$L(x_1,\cdots,x_n) = \Big(\sum_{j=1}^n a_{1j}x_j, \sum_{j=1}^n a_{2j}x_j, \cdots, \sum_{j=1}^n a_{mj}x_j\Big)\,.$$

Then $L\in\mathscr{L}(\mathbb{F}^n,\mathbb{F}^m)$.

From Example 3.16, we see that any $m\times n$ matrix is associated with a linear map. Now suppose that $\mathbb{V}$ and $\mathbb{W}$ are vector spaces over a common scalar field $\mathbb{F}$, $\mathbb{V}$ is a $n$-dimensional vector space with basis $\mathcal{B} = \{\boldsymbol{v}_j\}_{j=1}^n$, and $\mathbb{W}$ is a $m$-dimensional vector space with basis $\widetilde{\mathcal{B}} = \{\boldsymbol{w}_i\}_{i=1}^m$. Let $L\in\mathscr{L}(\mathbb{V};\mathbb{W})$. Since $\widetilde{B}$ is a basis of $\mathbb{W}$, for each $1\leqslant j\leqslant n$ there exist unique $a_{1j}$, $a_{2j}$, $\cdots$, $a_{mj}\in\mathbb{F}$ such that $L\boldsymbol{v}_j = \sum\limits_{i=1}^m a_{ij}\boldsymbol{w}_i$. Moreover, if $\boldsymbol{u}\in\mathbb{V}$, then there exist $c_1,\cdots,c_n\in\mathbb{F}$ such that

$$\boldsymbol{u} = \sum_{j=1}^n c_j\boldsymbol{v}_j \qquad \text{or} \qquad \boldsymbol{c} = [\boldsymbol{u}]_{\mathcal{B}}\,,$$

and by the linearity of $L$,

$$L\boldsymbol{u} = L\Big(\sum_{j=1}^n c_j\boldsymbol{v}_j\Big) = \sum_{j=1}^n c_jL\boldsymbol{v}_j = \sum_{j=1}^n\sum_{i=1}^m c_ja_{ij}\boldsymbol{w}_i = \sum_{i=1}^m\Big(\sum_{j=1}^n a_{ij}c_j\Big)\boldsymbol{w}_i\,.$$

Let $b_i = \sum\limits_{j=1}^n a_{ij}c_j$, and $\boldsymbol{b} = [b_1,\cdots,b_m]^{\mathrm{T}}$. Then with $A$ denoting the $m\times n$ matrix $[a_{ij}]_{1\leqslant i\leqslant m,1\leqslant j\leqslant n}$,

$$[L\boldsymbol{u}]_{\widetilde{B}} = \boldsymbol{b} = A\boldsymbol{c} = A[\boldsymbol{u}]_{\mathcal{B}}\,.$$

The discussion above induces the following

**Definition 3.17.** Let $\mathbb{V},\mathbb{W}$ be two vector spaces, $\dim(\mathbb{V}) = n$ and $\dim(\mathbb{W}) = m$, and $\mathcal{B},\widetilde{\mathcal{B}}$ are basis of $\mathbb{V},\mathbb{W}$, respectively. For $L\in\mathscr{L}(\mathbb{V};\mathbb{W})$, the ***matrix representation*** of $L$ relative to bases $\mathcal{B}$ and $\widetilde{\mathcal{B}}$, denoted by $[L]_{\widetilde{\mathcal{B}},\mathcal{B}}$, is the matrix satisfying

$$[L\boldsymbol{u}]_{\widetilde{B}} = [L]_{\widetilde{\mathcal{B}},\mathcal{B}}[\boldsymbol{u}]_{\mathcal{B}} \qquad \forall\,\boldsymbol{u}\in\mathbb{V}\,.$$

If $L\in\mathscr{L}(\mathbb{V};\mathbb{V})$, we simply use $[L]_{\mathcal{B}}$ to denote $[L]_{\mathcal{B},\mathcal{B}}$.

**Example 3.18.** Let $\mathbb{V} = \text{span}(1, x, \cdots, x^{n-1})$ and $\mathbb{W} = \text{span}(1, x, \cdots, x^{m-1})$ with $m \geqslant n-1$. Then $\dfrac{d}{dx} : \mathbb{V} \to \mathbb{W}$ defined by

$$\frac{d}{dx}\Big(\sum_{k=1}^{n} a_k x^{k-1}\Big) = \sum_{k=1}^{n} a_k (k-1) x^{k-2}$$

is linear, and the matrix representation of $\dfrac{d}{dx}$ (relative to the standard basis of $\mathbb{V}$ and $\mathbb{W}$) is

$$m\text{-rows}\left\{\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & 0 & 2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & 0 & (n-1) \\ \vdots & & & \ddots & 0 \\ \vdots & & & & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix}\right. .$$

**Theorem 3.19.** *Let $\mathbb{V}_1, \mathbb{V}_2, \mathbb{V}_3$ be finite dimensional vector spaces, and $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$ be basis of $\mathbb{V}_1$, $\mathbb{V}_2$, $\mathbb{V}_3$, respectively. Then*

$$[TS]_{\mathcal{B}_3,\mathcal{B}_1} = [T]_{\mathcal{B}_3,\mathcal{B}_2}[S]_{\mathcal{B}_2,\mathcal{B}_1} \qquad \forall\, S \in \mathscr{L}(\mathbb{V}_1;\mathbb{V}_2), T \in \mathscr{L}(\mathbb{V}_2;\mathbb{V}_3)\,.$$

*Proof.* Let $S \in \mathscr{L}(\mathbb{V}_1;\mathbb{V}_2)$ and $T \in \mathscr{L}(\mathbb{V}_2;\mathbb{V}_3)$ be given. For all $\boldsymbol{v} \in \mathbb{V}_1$,

$$[TS\boldsymbol{v}]_{\mathcal{B}_3} = [T]_{\mathcal{B}_3,\mathcal{B}_2}[S\boldsymbol{v}]_{\mathcal{B}_2} = [T]_{\mathcal{B}_3,\mathcal{B}_2}[S]_{\mathcal{B}_2,\mathcal{B}_1}[\boldsymbol{v}]_{\mathcal{B}_1} \qquad \text{and} \qquad [TS\boldsymbol{v}]_{\mathcal{B}_3} = [TS]_{\mathcal{B}_3,\mathcal{B}_1}[\boldsymbol{v}]_{\mathcal{B}_1}\,.$$

Therefore,

$$[T]_{\mathcal{B}_3,\mathcal{B}_2}[S]_{\mathcal{B}_2,\mathcal{B}_1}[\boldsymbol{v}]_{\mathcal{B}_1} = [TS]_{\mathcal{B}_3,\mathcal{B}_1}[\boldsymbol{v}]_{\mathcal{B}_1} \qquad \forall\, \boldsymbol{v} \in \mathbb{V}_1\,;$$

thus letting $\boldsymbol{v}$ be any basis vector implies that

$$[TS]_{\mathcal{B}_3,\mathcal{B}_1} = [T]_{\mathcal{B}_3,\mathcal{B}_2}[S]_{\mathcal{B}_2,\mathcal{B}_1}\,. \qquad \square$$

### 3.1.3 Algebraic dual spaces

In the rest of this section, we will only focus on the space $\mathscr{L}(\mathbb{V};\mathbb{W})$ for the case that the codomain $\mathbb{W}$ is the underlying scalar field $\mathbb{F}$. In such a case, we use $\mathbb{V}'$ to denote $\mathscr{L}(\mathbb{V};\mathbb{F})$, called the *algebraic* dual space of $\mathbb{V}$.

**Example 3.20.** Let $\mathbb{V} = \mathbb{R}^n$. From Linear Algebra we know that $\mathbb{V}' = \mathbb{R}^n$ in the sense that every $f \in \mathbb{V}'$ corresponds to a unique matrix $\boldsymbol{a} \equiv [a_1, \cdots, a_n] \in \mathbb{R}^n$ such that

$$f(\boldsymbol{x}) = \boldsymbol{a} \cdot \boldsymbol{x}$$

so that we identify $f$ as the vector $\boldsymbol{a} \in \mathbb{R}^n$ to obtain that $\mathbb{V}'$ "=" $\mathbb{V}$.

A bit more generalized version of the result above is that $(\mathbb{C}^n)' = \mathbb{C}^n$ in the sense that every $f \in (\mathbb{C}^n)'$ corresponds to a unique vector $\boldsymbol{a} \equiv (c_1, \cdots, c_n) \in \mathbb{C}^n$ such that

$$f(\boldsymbol{z}) = \boldsymbol{c} \cdot \boldsymbol{z} = \sum_{j=1}^{n} \overline{c}_j z_j \,,$$

where $\overline{c}_j$ is the complex conjugate of $c_j$.

We note that the example above shows that $\dim(\mathbb{R}^n) = \dim\big((\mathbb{R}^n)'\big)$ and $\dim(\mathbb{C}^n) = \dim\big((\mathbb{C}^n)'\big)$. In general, we have the following

**Proposition 3.21.** *Let $\mathbb{V}$ be a finite dimensional vector space over field $\mathbb{F}$. Then* $\dim(\mathbb{V}') = \dim(\mathbb{V})$.

*Proof.* Let $\dim(\mathbb{V}) = n$ and $\{\mathbf{e}_1, \cdots, \mathbf{e}_n\}$ be a basis of $\mathbb{V}$. Define $\varphi_1, \cdots \varphi_n$ by

$$\varphi_i\Big(\sum_{j=1}^{n} c_j \mathbf{e}_j\Big) = \sum_{j=1}^{n} c_j \delta_{ij} \qquad \forall\, 1 \leqslant i \leqslant n \;\text{ and }\; c_j \in \mathbb{F}\,. \tag{3.1}$$

Then $\varphi_1, \cdots, \varphi_n \in \mathbb{V}'$. Moreover, the collection $\{\varphi_1, \cdots, \varphi_n\}$ are linearly independent since if $\alpha_1, \cdots, \alpha_n \in \mathbb{F}$ verify that

$$\alpha_1\varphi_1 + \alpha_2\varphi_2 + \cdots + \alpha_n\varphi_n = 0 \;\; \text{(the zero function)}\,,$$

we must have

$$(\alpha_1\varphi_1 + \alpha_2\varphi_2 + \cdots + \alpha_n\varphi_n)(\mathbf{e}_j) = 0 \qquad \forall\, 1 \leqslant j \leqslant n$$

which, using (3.1), implies that $\alpha_j = 0$ for all $1 \leqslant j \leqslant n$. Therefore, $\dim(\mathbb{V}') \geqslant n$.

On the other hand, suppose that $f \in \mathbb{V}'$ and $f(\mathbf{e}_j) = d_j$. If $\boldsymbol{x} = x_1\mathbf{e}_1 + \cdots + x_n\mathbf{e}_n$, the linearity of $g$ implies that

$$f(\boldsymbol{x}) = f(x_1\mathbf{e}_1 + \cdots + x_n\mathbf{e}_n) = x_1 f(\mathbf{e}_1) + \cdots + x_n f(\mathbf{e}_n) = d_1x_1 + \cdots + d_nx_n$$

and (3.1) shows that

$$(d_1\varphi_1+\cdots+d_n\varphi_n)(\boldsymbol{x}) = (d_1\varphi_1+\cdots+d_n\varphi_n)(x_1\mathbf{e}_1+\cdots+x_n\mathbf{e}_n) = \sum_{i=1}^{n} d_i\varphi_i\Big(\sum_{j=1}^{n} x_j\mathbf{e}_j\Big)$$
$$= \sum_{i=1}^{n}\sum_{j=1}^{n} d_i x_j\delta_{ij} = \sum_{i=1}^{n} d_i x_i = d_1x_1+\cdots+d_nx_n\,.$$

Therefore, $f = d_1\varphi_1+\cdots+d_n\varphi_n$ which implies that $\mathbb{V}' = \text{span}(\varphi_1,\cdots,\varphi_n)$. This establishes that $\dim(\mathbb{V}') = n$. □

## 3.2 Direct Sum of Vector Spaces and Multi-Linear Maps

### 3.2.1 Direct sum of vector spaces

**Definition 3.22.** Given sets $A$ and $B$, the Cartesian product of $A$ and $B$, denoted by $A\times B$, is the set of all ordered pairs $(a,b)$ with $a\in A$ and $b\in B$; that is, $A\times B = \big\{(a,b)\,\big|\,a\in A \text{ and } b\in B\big\}$. The Cartesian of three or more sets are defined similarly.

Let $X$ and $Y$ be vector spaces over a common scalar field $\mathbb{F}$. The ***direct sum*** of $X$ and $Y$, denoted by $X\oplus Y$, is $X\times Y$ with the following vector space structure:

$$\lambda\cdot(\boldsymbol{x}_1,\boldsymbol{y}_1)+(\boldsymbol{x}_2,\boldsymbol{y}_2) = (\lambda\cdot\boldsymbol{x}_1+\boldsymbol{x}_2,\lambda\cdot\boldsymbol{y}_1+\boldsymbol{y}_2) \qquad \forall\,\lambda\in\mathbb{F},\boldsymbol{x}_1,\boldsymbol{x}_2\in X \text{ and } \boldsymbol{y}_1,\boldsymbol{y}_2\in Y\,.$$

For $\boldsymbol{x}\in X$ and $\boldsymbol{y}\in Y$, the ordered pair $(\boldsymbol{x},\boldsymbol{y})$ is also written as $\boldsymbol{x}\oplus\boldsymbol{y}$.

**Remark 3.23.** 1. The direct sum is a way of getting a new big vector space from two (or more) smaller vector spaces in the simplest way one can imagine: you just line them up.

2. Let $X,Y$ be finite dimensional vector spaces over a scalar field $\mathbb{F}$, where $\mathbb{F}=\mathbb{R}$ or $\mathbb{C}$. Then $X\oplus Y$ is a finite dimensional vector space over $\mathbb{F}$ and $\dim(X\oplus Y) = \dim(X)+\dim(Y)$ for $X\oplus Y$ has a basis $\big\{\boldsymbol{x}_1\oplus\mathbf{0},\boldsymbol{x}_2\oplus\mathbf{0},\cdots,\boldsymbol{x}_m\oplus\mathbf{0},\mathbf{0}\oplus\boldsymbol{y}_1,\mathbf{0}\oplus\boldsymbol{y}_2,\cdots,\mathbf{0}\oplus\boldsymbol{y}_n\big\}$, where $\big\{\boldsymbol{x}_1,\cdots,\boldsymbol{x}_m\big\}$ is a basis of $X$ and $\{\boldsymbol{y}_1,\cdots,\boldsymbol{y}_n\}$ is a basis of $Y$.

**Definition 3.24.** Let $X$, $Y$, $Z$, $W$ be vector spaces over field $\mathbb{F}$, and $A\in\mathscr{L}(X;Z)$, $B\in\mathscr{L}(Y;W)$. The direct sum of $A$ and $B$, denoted by $A\oplus B$, is a linear map in $\mathscr{L}(X\oplus Y,Z\oplus W)$ satisfying that

$$(A\oplus B)(\boldsymbol{x}\oplus\boldsymbol{y}) = (A\boldsymbol{x})\oplus(B\boldsymbol{y}) \qquad \forall\,\boldsymbol{x}\in X,\boldsymbol{y}\in Y\,.$$

**Theorem 3.25.** *Let $X_1, X_2, X_3, Y_1, Y_2, Y_3$ be vectors spaces over field $\mathbb{F}$, and $A_1 \in \mathscr{L}(X_1; X_2)$, $A_2 \in \mathscr{L}(X_2; X_3)$, $B_1 \in \mathscr{L}(Y_1; Y_2)$, $B_2 \in \mathscr{L}(Y_2; Y_3)$. Then*

$$(A_2 \oplus B_2)(A_1 \oplus B_1) = (A_2 A_1) \oplus (B_2 B_1)\,.$$

*Proof.* Let $\boldsymbol{x} \in X_1$ and $\boldsymbol{y} \in Y_1$. Then by the definition of the tensor product of linear maps,

$$\begin{aligned}(A_2 \oplus B_2)(A_1 \oplus B_1)(\boldsymbol{x} \oplus \boldsymbol{y}) &= (A_2 \oplus B_2)(A_1\boldsymbol{x} \oplus B_1\boldsymbol{y}) = (A_2A_1\boldsymbol{x} \oplus B_2B_1\boldsymbol{y}) \\ &= (A_2A_1 \oplus B_2B_1)(\boldsymbol{x} \oplus \boldsymbol{y})\,.\end{aligned}$$

□

**Theorem 3.26.** *Let $X_1$, $X_2$, $Y_1$, $Y_2$ be finite dimensional vector spaces over field $\mathbb{F}$, and $A \in \mathscr{L}(X_1; X_2)$, $B \in \mathscr{L}(Y_1; Y_2)$. Suppose that relative to given basis of $X_1, X_2, Y_1, Y_2$, the matrix representation of $A$ and $B$ are $[A]$ and $[B]$, respectively. Then relative to the basis of $X_1 \oplus Y_1$ and $X_2 \oplus Y_2$ associated with given basis of $X_1, X_2, Y_1, Y_2$ (explained in Remark 3.23), the matrix representation of $A \oplus B$ is*

$$\begin{bmatrix} [A] & \mathbf{0} \\ \mathbf{0} & [B] \end{bmatrix}.$$

### 3.2.2 Multi-linear maps

Multi-linearity is an extension of linearity of maps.

**Definition 3.27.** Let $\mathbb{V}_1, \mathbb{V}_2, \mathbb{W}$ be vector spaces over a common scalar field $\mathbb{F}$. A map $L : \mathbb{V}_1 \oplus \mathbb{V}_2 \to \mathbb{W}$ is said to be bilinear map provided that

$$L(c\boldsymbol{u} + \boldsymbol{v}, \boldsymbol{w}) = cL(\boldsymbol{u}, \boldsymbol{w}) + L(\boldsymbol{v}, \boldsymbol{w}) \qquad \forall\, \boldsymbol{u}, \boldsymbol{v} \in \mathbb{V}_1, \boldsymbol{w} \in \mathbb{V}_2 \text{ and } c \in \mathbb{F}\,, \tag{3.2a}$$

$$L(\boldsymbol{u}, c\boldsymbol{v} + \boldsymbol{w}) = cL(\boldsymbol{u}, \boldsymbol{v}) + L(\boldsymbol{u}, \boldsymbol{w}) \qquad \forall\, \boldsymbol{u} \in \mathbb{V}_1, \boldsymbol{v}, \boldsymbol{w} \in \mathbb{V}_2 \text{ and } c \in \mathbb{F}\,. \tag{3.2b}$$

The collection of all maps $L : X \oplus Y \to Z$ satisfying (3.2) is denoted by $\mathscr{L}(\mathbb{V}_1, \mathbb{V}_2; \mathbb{W})$.

We can extend the bilinearity to multi-linearity easily through the following

**Definition 3.28.** Let $\mathbb{V}_1, \cdots, \mathbb{V}_n, \mathbb{W}$ be vector spaces over a common scalar field $\mathbb{F}$. A map $L : \mathbb{V}_1 \oplus \cdots \oplus \mathbb{V}_n \to \mathbb{W}$ is said to be multi-linear, denoted by $L \in \mathscr{L}(\mathbb{V}_1, \cdots, \mathbb{V}_n; \mathbb{W})$, provided that

$$\begin{aligned}&L(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_{j-1}, c\boldsymbol{v}_j + \boldsymbol{w}_j, \boldsymbol{u}_{j+1}, \cdots, \boldsymbol{u}_n) \\ &\quad = cL(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_{j-1}, \boldsymbol{v}_j, \boldsymbol{u}_{j+1}, \cdots, \boldsymbol{u}_n) + L(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_j, \boldsymbol{w}_{j-1}, \boldsymbol{u}_{j+1}, \cdots, \boldsymbol{u}_n)\end{aligned}$$

for all $1 \leqslant j \leqslant n$ and $c \in \mathbb{F}$, and $\boldsymbol{u}_\ell \in \mathbb{V}_\ell$ for all $\ell \neq j$, $\boldsymbol{v}_j, \boldsymbol{w}_j \in \mathbb{V}_j$.

**Proposition 3.29.** *Let $\mathbb{V}_1, \cdots, \mathbb{V}_n, \mathbb{W}$ be vector spaces over a common scalar field $\mathbb{F}$. Then $\mathscr{L}(\mathbb{V}_1, \cdots, \mathbb{V}_n; \mathbb{W})$ is a vector space over $\mathbb{F}$.*

*Proof.* Let $f, g \in \mathscr{L}(\mathbb{V}_1, \cdots, \mathbb{V}_n)$, and $\alpha \in \mathbb{F}$. Then if $1 \leqslant j \leqslant n$, $c \in \mathbb{F}$, and $\boldsymbol{u}_\ell \in \mathbb{V}_\ell$ for all $\ell \neq j$, $\boldsymbol{v}_j, \boldsymbol{w}_j \in \mathbb{V}_j$, we have

$$
\begin{aligned}
&(\alpha f + g)(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_{j-1}, c\boldsymbol{v}_j + \boldsymbol{w}_j, \boldsymbol{u}_{j+1}, \cdots, \boldsymbol{u}_n) \\
&\quad = \alpha f(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_{j-1}, c\boldsymbol{v}_j + \boldsymbol{w}_j, \boldsymbol{u}_{j+1}, \cdots, \boldsymbol{u}_n) + g(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_{j-1}, c\boldsymbol{v}_j + \boldsymbol{w}_j, \boldsymbol{u}_{j+1}, \cdots, \boldsymbol{u}_n) \\
&\quad = \alpha\big[cf(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_{j-1}, \boldsymbol{v}_j, \boldsymbol{u}_{j+1}, \cdots, \boldsymbol{u}_n) + f(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_{j-1}, \boldsymbol{w}_j, \boldsymbol{u}_{j+1}, \cdots, \boldsymbol{u}_n)\big] \\
&\qquad + cg(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_{j-1}, \boldsymbol{v}_j, \boldsymbol{u}_{j+1}, \cdots, \boldsymbol{u}_n) + g(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_{j-1}, \boldsymbol{w}_j, \boldsymbol{u}_{j+1}, \cdots, \boldsymbol{u}_n) \\
&\quad = c(\alpha f + g)(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_{j-1}, \boldsymbol{v}_j, \boldsymbol{u}_{j+1}, \cdots, \boldsymbol{u}_n) + (\alpha f + g)(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_{j-1}, \boldsymbol{w}_j, \boldsymbol{u}_{j+1}, \cdots, \boldsymbol{u}_n)\,.
\end{aligned}
$$

Therefore, $\alpha f + g \in \mathscr{L}(\mathbb{V}_1, \cdots, \mathbb{V}_n; \mathbb{W})$. ▫

## 3.3 Inner Product Spaces and Hilbert Spaces

**Definition 3.30.** An ***inner product space*** $\big(\mathbb{V}, \langle \cdot, \cdot \rangle\big)$ is a vector space $\mathbb{V}$ over a scalar field $\mathbb{F}$ (where $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$) associated with a function $\langle \cdot, \cdot \rangle : \mathbb{V} \times \mathbb{V} \to \mathbb{F}$ such that

(1) $\langle \boldsymbol{x}, \boldsymbol{x} \rangle \geqslant 0, \ \forall\, \boldsymbol{x} \in \mathbb{V}$.

(2) $\langle \boldsymbol{x}, \boldsymbol{x} \rangle = 0$ if and only if $\boldsymbol{x} = 0$.

(3) $\langle \boldsymbol{x}, \boldsymbol{y} + \boldsymbol{z} \rangle = \langle \boldsymbol{x}, \boldsymbol{y} \rangle + \langle \boldsymbol{x}, \boldsymbol{z} \rangle$ for all $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{V}$.

(4) $\langle \boldsymbol{x}, \lambda \boldsymbol{y} \rangle = \lambda \langle \boldsymbol{x}, \boldsymbol{y} \rangle$ for all $\lambda \in \mathbb{F}$ and $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{V}$.

(5) $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \overline{\langle \boldsymbol{y}, \boldsymbol{x} \rangle}$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{V}$, where $\overline{c}$ denotes the complex conjugate of $c$.

A function $\langle \cdot, \cdot \rangle$ satisfying (1)-(5) is called an ***inner product*** on $\mathbb{V}$.

**Proposition 3.31.** *Let $\langle \cdot, \cdot \rangle$ be an inner product on a vector space $\mathbb{V}$ over a scalar field $\mathbb{F}$. Then*

1. $\langle \boldsymbol{u}, \lambda \boldsymbol{v} + \mu \boldsymbol{w} \rangle = \lambda \langle \boldsymbol{u}, \boldsymbol{v} \rangle + \mu \langle \boldsymbol{u}, \boldsymbol{w} \rangle$ *for all* $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in \mathbb{V}$ *and* $\lambda, \mu \in \mathbb{F}$.
2. $\langle \lambda \boldsymbol{v} + \mu \boldsymbol{w}, \boldsymbol{u} \rangle = \overline{\lambda} \langle \boldsymbol{v}, \boldsymbol{u} \rangle + \overline{\mu} \langle \boldsymbol{w}, \boldsymbol{u} \rangle$ *for all* $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in \mathbb{V}$ *and* $\lambda, \mu \in \mathbb{F}$.
3. $\langle \boldsymbol{0}, \boldsymbol{w} \rangle = \langle \boldsymbol{w}, \boldsymbol{0} \rangle = 0$ *for all* $\boldsymbol{w} \in \mathbb{V}$.

**Theorem 3.32.** *The inner product $\langle \cdot, \cdot \rangle$ on a vector space $\mathbb{V}$ over scalar field $\mathbb{F}$ satisfies the* ***Cauchy-Schwarz inequality***

$$\big|\langle \boldsymbol{x}, \boldsymbol{y} \rangle\big| \leqslant \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}\sqrt{\langle \boldsymbol{y}, \boldsymbol{y} \rangle} \qquad \forall\, \boldsymbol{x}, \boldsymbol{y} \in \mathbb{V}\,. \tag{3.3}$$

*Moreover, for non-zero vectors $\boldsymbol{x}, \boldsymbol{y}$, the equality holds if and only if there exists $\gamma \in \mathbb{F}$ such that $\boldsymbol{x} = \gamma \boldsymbol{y}$.*

*Proof.* Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{V}$. Define $\alpha = \langle \boldsymbol{x}, \boldsymbol{y} \rangle$. W.L.O.G. we can assume that $\alpha \neq 0$ (for otherwise (3.3) holds trivially). Then there exists $\beta \in \mathbb{F}$ such that $\alpha \cdot \beta = |\alpha|$ (so $|\beta| = 1$). For any $\lambda \in \mathbb{R}$,

$$\begin{aligned} 0 &\leqslant \langle \lambda\beta\boldsymbol{x} + \boldsymbol{y}, \lambda\beta\boldsymbol{x} + \boldsymbol{y} \rangle = \lambda^2|\beta|^2\langle \boldsymbol{x}, \boldsymbol{x} \rangle^2 + \langle \lambda\beta\boldsymbol{x}, \boldsymbol{y} \rangle + \langle \boldsymbol{y}, \lambda\beta\boldsymbol{x} \rangle + \langle \boldsymbol{y}, \boldsymbol{y} \rangle^2 \\ &= \lambda^2\langle \boldsymbol{x}, \boldsymbol{x} \rangle^2 + \lambda\beta\langle \boldsymbol{x}, \boldsymbol{y} \rangle + \lambda\overline{\langle \beta\boldsymbol{x}, \boldsymbol{y} \rangle} + \langle \boldsymbol{y}, \boldsymbol{y} \rangle^2 \\ &= \lambda^2\langle \boldsymbol{x}, \boldsymbol{x} \rangle^2 + 2\lambda\big|\langle \boldsymbol{x}, \boldsymbol{y} \rangle\big| + \langle \boldsymbol{y}, \boldsymbol{y} \rangle^2\,. \end{aligned} \tag{3.4}$$

Since the right-hand side in the inequality above is always non-negative for all real $\lambda$, we must have

$$\big|\langle \boldsymbol{x}, \boldsymbol{y} \rangle\big|^2 - \langle \boldsymbol{x}, \boldsymbol{x} \rangle \cdot \langle \boldsymbol{y}, \boldsymbol{y} \rangle \leqslant 0$$

which implies (3.3).

Finally, suppose that $\boldsymbol{x}, \boldsymbol{y} \neq \boldsymbol{0}$ and $\big|\langle \boldsymbol{x}, \boldsymbol{y} \rangle\big| = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}\sqrt{\langle \boldsymbol{y}, \boldsymbol{y} \rangle}$. Then with $\lambda \in \mathbb{F}$ given by $\lambda = -\sqrt{\dfrac{\langle \boldsymbol{y}, \boldsymbol{y} \rangle}{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}}$, (3.4) shows that

$$\begin{aligned} 0 &\leqslant \langle \lambda\beta\boldsymbol{x} + \boldsymbol{y}, \lambda\beta\boldsymbol{x} \rangle = \lambda^2\langle \boldsymbol{x}, \boldsymbol{x} \rangle + 2\lambda\sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}\sqrt{\langle \boldsymbol{y}, \boldsymbol{y} \rangle} + \langle \boldsymbol{y}, \boldsymbol{y} \rangle \\ &= \big(\lambda\sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle} + \sqrt{\langle \boldsymbol{y}, \boldsymbol{y} \rangle}\big)^2 = 0\,; \end{aligned}$$

thus $\lambda\beta\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{0}$. ▫

**Definition 3.33.** A ***normed vector space*** (or simply ***normed space***) $(\mathbb{V}, \|\cdot\|)$ is a vector space $\mathbb{V}$ over a scalar field $\mathbb{F}$, where $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$, associated with a function $\|\cdot\| : \mathbb{V} \to \mathbb{R}$ such that

(a) $\|\boldsymbol{x}\| \geqslant 0$ for all $\boldsymbol{x} \in \mathbb{V}$.

(b) $\|\boldsymbol{x}\| = 0$ if and only if $\boldsymbol{x} = \boldsymbol{0}$.

(c) $\|\lambda \cdot \boldsymbol{x}\| = |\lambda| \cdot \|\boldsymbol{x}\|$ for all $\lambda \in \mathbb{F}$ and $\boldsymbol{x} \in \mathbb{V}$.

(d) $\|\boldsymbol{x}+\boldsymbol{y}\| \leqslant \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{V}$.

A function $\|\cdot\|$ satisfying (a)-(d) is called a ***norm*** on $\mathbb{V}$.

**Theorem 3.34.** *The inner product $\langle\cdot,\cdot\rangle$ on a vector space $\mathbb{V}$ (over scalar field $\mathbb{F}$) induces a norm $\|\cdot\|$ given by $\|\boldsymbol{x}\| = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x}\rangle}$.*

*Proof.* It should be clear that (a)-(c) in Definition 3.33 are satisfied. To show that $\|\cdot\|$ satisfies the triangle inequality, by (3.3) we find that

$$\begin{aligned}\big(\|\boldsymbol{x}\| + \|\boldsymbol{y}\|\big)^2 - \|\boldsymbol{x}+\boldsymbol{y}\|^2 &= \|\boldsymbol{x}\|^2 + 2\|\boldsymbol{x}\|\|\boldsymbol{y}\| + \|\boldsymbol{y}\|^2 - \langle \boldsymbol{x}+\boldsymbol{y}, \boldsymbol{x}+\boldsymbol{y}\rangle \\ &= 2\big(\|\boldsymbol{x}\|\|\boldsymbol{y}\| - \mathrm{Re}\langle \boldsymbol{x}, \boldsymbol{y}\rangle\big) \geqslant 2\big(\|\boldsymbol{x}\|\|\boldsymbol{y}\| - \big|\langle \boldsymbol{x}, \boldsymbol{y}\rangle\big|\big) \geqslant 0\,;\end{aligned}$$

thus the triangle inequality is also valid. ▫

Having introduced the induced norm of inner product spaces, it is easy to see the following two propositions and the proof of the propositions is left to the readers.

**Proposition 3.35** (Parallelogram Law)**.** *Let $(\mathbb{V}, \langle\cdot,\cdot\rangle)$ be an inner product space, and $\|\cdot\|$ be the norm induced by the inner product. Then*

$$\|\boldsymbol{x}-\boldsymbol{y}\|^2 + \|\boldsymbol{x}+\boldsymbol{y}\|^2 = 2\big(\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2\big) \qquad \forall\, \boldsymbol{x}, \boldsymbol{y} \in \mathbb{V}\,.$$

**Proposition 3.36** (Polarization Identity)**.** *Let $(\mathbb{V}, \langle\cdot,\cdot\rangle)$ be an inner product space over $\mathbb{F}$, and $\|\cdot\|$ be the norm induced by the inner product.*

1. *If $\mathbb{F} = \mathbb{R}$, then $\langle \boldsymbol{x}, \boldsymbol{y}\rangle = \dfrac{1}{4}\big[\|\boldsymbol{x}+\boldsymbol{y}\|^2 - \|\boldsymbol{x}-\boldsymbol{y}\|^2\big]$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{V}$.*
2. *If $\mathbb{F} = \mathbb{C}$, then $\langle \boldsymbol{x}, \boldsymbol{y}\rangle = \dfrac{1}{4}\big[\|\boldsymbol{x}+\boldsymbol{y}\|^2 - \|\boldsymbol{x}-\boldsymbol{y}\|^2 - i\|\boldsymbol{x}+i\boldsymbol{y}\|^2 + i\|\boldsymbol{x}-i\boldsymbol{y}\|^2\big]$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{V}$.*

We remark here that the polarization identity provides a way to reconstruct the inner product once you only have the induced norm. The polarization identity provides a way to verify if the norm of a normed space is induced by some inner product.

**Definition 3.37.** A ***Banach space*** is a complete normed vector space, and a ***Hilbert space*** is a complete inner product space (that is, a Banach space whose norm is induced by the inner product).

**Remark 3.38.** In the definition above, the completeness of a normed vector space is defined as follows.

1. A sequence $\{\boldsymbol{x}_n\}_{n=1}^{\infty}$ is called a Cauchy sequence in a normed vector space $(\mathbb{V}, \|\cdot\|)$ if
$$(\forall\, \varepsilon > 0)(\exists\, N > 0)(n, m \geqslant N \Rightarrow \|x_n - x_m\| < \varepsilon)\,.$$

2. A normed space $(\mathbb{V}, \|\cdot\|)$ is complete if every Cauchy sequence in $\mathbb{V}$ converges; that is, if $\{\boldsymbol{x}_n\}_{n=1}^{\infty}$ is a Cauchy sequence in $\mathbb{V}$, then there exists $\boldsymbol{x} \in \mathbb{V}$ such that $\lim\limits_{n\to\infty} \|\boldsymbol{x}_n - \boldsymbol{x}\| = 0$.

**Theorem 3.39.** *Let $(\mathbb{H}, \langle\cdot,\cdot\rangle)$ be a Hilbert space, and $E$ be a closed convex subset of $\mathbb{H}$. If $\boldsymbol{x} \notin E$, there exists a unique $\boldsymbol{x}_0 \in E$ such that*
$$\|\boldsymbol{x} - \boldsymbol{x}_0\| = \operatorname{dist}(\boldsymbol{x}, E) = \inf\big\{\|\boldsymbol{x} - \boldsymbol{y}\| \,\big|\, \boldsymbol{y} \in E\big\}.$$

*Proof.* By the definition of the infimum, there exists a sequence $\{\boldsymbol{x}_n\}_{n=1}^{\infty} \subseteq E$ such that
$$\operatorname{dist}(\boldsymbol{x}, E)^2 \leqslant \|\boldsymbol{x} - \boldsymbol{x}_n\|^2 < \operatorname{dist}(\boldsymbol{x}, E)^2 + \frac{1}{n}\,.$$
By the parallelogram law (Proposition 3.35),
$$\|\boldsymbol{b} - \boldsymbol{c}\|^2 + 4\Big\|\boldsymbol{a} - \frac{\boldsymbol{b}+\boldsymbol{c}}{2}\Big\|^2 = 2\|\boldsymbol{a} - \boldsymbol{b}\|^2 + 2\|\boldsymbol{a} - \boldsymbol{c}\|^2 \qquad \forall\, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathbb{H}\,. \tag{3.5}$$
Let $\boldsymbol{a} = \boldsymbol{x}$, $\boldsymbol{b} = \boldsymbol{x}_n$ and $\boldsymbol{c} = \boldsymbol{x}_m$ in (3.5). The convexity of $E$ implies that $\dfrac{\boldsymbol{x}_n + \boldsymbol{x}_m}{2} \in E$; thus $\Big\|\boldsymbol{x} - \dfrac{\boldsymbol{x}_n + \boldsymbol{x}_m}{2}\Big\| \geqslant \operatorname{dist}(x, E)$. Therefore, for all $n, \boldsymbol{m} \in \mathbb{N}$,
$$\|\boldsymbol{x}_n - \boldsymbol{x}_m\|^2 \leqslant 2\big(\|\boldsymbol{x} - \boldsymbol{x}_n\|^2 + \|\boldsymbol{x} - \boldsymbol{x}_m\|^2\big) - 4\operatorname{dist}(\boldsymbol{x}, E)^2 < \frac{2}{n} + \frac{2}{m}\,.$$
The inequality above shows that $\{\boldsymbol{x}_n\}_{n=1}^{\infty}$ is a Cauchy sequence in $E$. Since $\mathbb{H}$ is complete, $\{\boldsymbol{x}_n\}_{n=1}^{\infty}$ converges to some point $\boldsymbol{x}_0 \in \mathbb{H}$, and the closedness of $E$ implies that $\boldsymbol{x}_0 \in E$. Moreover,
$$\|\boldsymbol{x} - \boldsymbol{x}_0\| = \lim_{n\to\infty} \|\boldsymbol{x} - \boldsymbol{x}_n\| = \operatorname{dist}(x, E)\,.$$
For the uniqueness of such a closest point, suppose that $\boldsymbol{y}_1, \boldsymbol{y}_2 \in E$ are distinct and satisfy
$$\|\boldsymbol{x} - \boldsymbol{y}_1\| = \|\boldsymbol{x} - \boldsymbol{y}_2\| = \operatorname{dist}(\boldsymbol{x}, E)\,.$$
Then (3.5) implies that
$$\Big\|\boldsymbol{x} - \frac{\boldsymbol{y}_1 + \boldsymbol{y}_2}{2}\Big\|^2 = \frac{1}{4}\big[2\|\boldsymbol{x} - \boldsymbol{y}_1\|^2 + 2\|\boldsymbol{x} - \boldsymbol{y}_2\|^2 - \|\boldsymbol{y}_1 - \boldsymbol{y}_2\|^2\big] < \operatorname{dist}(\boldsymbol{x}, E)^2\,.$$
By the convexity of $E$, we have $\dfrac{\boldsymbol{y}_1 + \boldsymbol{y}_2}{2} \in E$; thus the above ineqaulity implies that $\boldsymbol{y}_1, \boldsymbol{y}_2$ cannot be the closest point of $E$ to point $\boldsymbol{x}$, a contradiction. ▫

**Definition 3.40.** Let $(\mathbb{H}, \langle\cdot,\cdot\rangle)$ be a Hilbert space, and $W$ be a subspace of $\mathbb{H}$. The orthogonal complement of $W$, denoted by $W^\perp$, is the set

$$W^\perp = \left\{ \boldsymbol{x} \in \mathbb{H} \,\middle|\, \langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0 \text{ for all } \boldsymbol{y} \in W \right\}.$$

**Proposition 3.41.** *Let $(\mathbb{H}, \langle\cdot,\cdot\rangle)$ be a Hilbert space, and $W$ be a subspace of $\mathbb{H}$. Then $E^\perp$ is closed in $\mathbb{H}$.*

*Proof.* The valid of the proposition follows from that

$$E^\perp = \bigcap_{\boldsymbol{y} \in E} \left\{ \boldsymbol{x} \in \mathbb{H} \,\middle|\, \langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0 \right\}$$

and the fact that the set $\left\{ \boldsymbol{x} \in \mathbb{H} \,\middle|\, \langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0 \right\}$ is closed for each $\boldsymbol{y} \in \mathbb{H}$. ▫

**Lemma 3.42.** *Let $(\mathbb{H}, \langle\cdot,\cdot\rangle)$ be a Hilbert space, and $W$ be a subspace of $\mathbb{H}$. Then*

$$(W^\perp)^\perp = \overline{W}.$$

*Proof.* First we note that if $\boldsymbol{x} \in \mathcal{W}$, then $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0$ for all $\boldsymbol{y} \in W^\perp$. This implies that $\boldsymbol{x} \in (W^\perp)^\perp$. Therefore, $W \subseteq (W^\perp)^\perp$. By Proposition 3.41, we must have $\overline{W} \subseteq (W^\perp)^\perp$.

Suppose that $\overline{W} \subsetneq (W^\perp)^\perp$. Then there exists $\boldsymbol{x} \in (W^\perp)^\perp \cap \overline{W}^\complement$. Since $\overline{W}$ is a closed subspace of $\mathbb{H}$, $\overline{W}$ must be convex; thus Theorem 3.39 implies that there exists a unique $\boldsymbol{x}_0 \in \overline{W}$ such that

$$\|\boldsymbol{x} - \boldsymbol{x}_0\| = \operatorname{dist}(x, \overline{W}).$$

Note that $\boldsymbol{x} - \boldsymbol{x}_0 \in (W^\perp)^\perp$. On the other hand, $\boldsymbol{x} - \boldsymbol{x}_0 \in \overline{W}^\perp \subseteq W^\perp$ since $\boldsymbol{x}_0$ is the closed point to $\boldsymbol{x}$ in $\overline{W}$. Therefore,

$$\boldsymbol{x} - \boldsymbol{x}_0 \in (W^\perp)^\perp \cap W^\perp$$

which implies that $\boldsymbol{x} - \boldsymbol{x}_0 = \boldsymbol{0}$, a contradiction to that $\boldsymbol{x} \notin \overline{W}$. ▫

**Corollary 3.43.** *Let $(\mathbb{H}, \langle\cdot,\cdot\rangle)$ be a Hilbert space, and $W$ be a closed subspace of $\mathbb{H}$. If $W \neq \mathbb{H}$, then $W^\perp \neq \varnothing$.*

**• Orthonormal basis**

**Definition 3.44.** Let $(\mathbb{V}, \langle\cdot,\cdot\rangle)$ be a finite dimensional inner product space. A basis $\mathcal{B} = \{\mathbf{v}_1, \cdots, \mathbf{v}_n\}$ of $\mathbb{V}$ is said to be orthonormal if $\langle \mathbf{v}_i, \mathbf{v}_j \rangle = \delta_{ij}$ for all $1 \leqslant i, j \leqslant n$, where $\delta_{ij}$ is the Kronecker delta.

Let $(\mathbb{V}, \langle\cdot,\cdot\rangle_{\mathbb{V}})$ and $(\mathbb{W}, \langle\cdot,\cdot\rangle_{\mathbb{W}})$ be two finite dimensional inner product spaces over $\mathbb{C}$, $\mathcal{B} = \{\mathbf{v}_1, \cdots, \mathbf{v}_n\}$ and $\widetilde{\mathcal{B}} = \{\mathbf{w}_1, \cdots, \mathbf{w}_m\}$ be orthonormal basis of $\mathbb{V}$ and $\mathbb{W}$, respectively, and $L \in \mathscr{L}(\mathbb{V}; \mathbb{W})$. If $A = [L]_{B,\widetilde{\mathcal{B}}} = [a_{ij}]_{m\times n}$ be the matrix representation of $L$ relative to $B$ and $\widetilde{\mathcal{B}}$, then by the fact that

$$L\boldsymbol{v} = \sum_{i=1}^{m} \Big( \sum_{j=1}^{n} a_{ij} v_j \Big) \mathbf{w}_i$$

we find that

$$\begin{aligned} \langle \boldsymbol{w}, L\boldsymbol{v} \rangle_{\mathbb{W}} &= \Big\langle \sum_{k=1}^{m} w_k \mathbf{w}_k, \sum_{i=1}^{m} \Big( \sum_{j=1}^{n} a_{ij} v_j \Big) \mathbf{w}_i \Big\rangle_{\mathbb{W}} = \sum_{j=1}^{n} \sum_{i,k=1}^{m} a_{ij} v_j \overline{w_k} \langle \mathbf{w}_k, \mathbf{w}_i \rangle_{\mathbb{W}} = \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} v_j \overline{w}_i \\ &= \big\langle [\boldsymbol{w}]_{\widetilde{\mathcal{B}}}, A[\boldsymbol{v}]_{\mathcal{B}} \big\rangle_{\mathbb{C}^m} = \big\langle [\boldsymbol{w}]_{\widetilde{\mathcal{B}}}, [L]_{\widetilde{\mathcal{B}},\mathcal{B}} [\boldsymbol{v}]_{\mathcal{B}} \big\rangle_{\mathbb{C}^m} = \big\langle [\boldsymbol{w}]_{\widetilde{\mathcal{B}}}, [L\boldsymbol{v}]_{\widetilde{\mathcal{B}}} \big\rangle_{\mathbb{C}^m} . \end{aligned}$$

The identity above converts the computation of the inner product of $\boldsymbol{w}$ and $L\boldsymbol{v}$ in $\mathbb{W}$ in terms of the inner product of $[\boldsymbol{w}]_{\widetilde{\mathcal{B}}}$ and $[L\boldsymbol{v}]_{\widetilde{\mathcal{B}}} (= [L]_{\widetilde{\mathcal{B}},\mathcal{B}} [\boldsymbol{v}]_{\mathcal{B}})$ in $\mathbb{C}^m$ using the matrix representation of $L$ and matrix multiplications.

In general, if $L_1, L_2 \in \mathscr{L}(\mathbb{V})$ and $\mathcal{B}$ is an orthonormal basis of $\mathbb{V}$, then

$$\langle L_1\boldsymbol{u}, L_2\boldsymbol{v} \rangle_{\mathbb{V}} = \big\langle [L_1]_{\mathcal{B}} [\boldsymbol{u}]_{\mathcal{B}}, [L_2]_{\mathcal{B}} [\boldsymbol{v}]_{\mathcal{B}} \big\rangle_{\mathbb{C}^n} \qquad \forall\, \boldsymbol{u}, \boldsymbol{v} \in \mathbb{V} \tag{3.6}$$

since

$$\langle L_1\boldsymbol{u}, L_2\boldsymbol{v} \rangle_{\mathbb{V}} = \big\langle [L_1\boldsymbol{u}]_{\mathcal{B}}, [L_2]_{\mathcal{B}} [\boldsymbol{v}]_{\mathcal{B}} \big\rangle_{\mathbb{C}^n} = \big\langle [L_1]_{\mathcal{B}} [\boldsymbol{u}]_{\mathcal{B}}, [L_2]_{\mathcal{B}} [\boldsymbol{v}]_{\mathcal{B}} \big\rangle_{\mathbb{C}^n} .$$

## 3.4 Dual Spaces and Adjoint Operators

**Definition 3.45.** Let $\mathbb{V}$ and $\mathbb{W}$ be vector spaces over a common scalar field $\mathbb{F}$ equipped with norms $\|\cdot\|_{\mathbb{V}}$ and $\|\cdot\|_{\mathbb{W}}$, respectively. A linear map $L : \mathbb{V} \to \mathbb{W}$ is said to be bounded if the number

$$\|L\|_{\mathscr{B}(\mathbb{V},\mathbb{W})} \equiv \sup_{\|\boldsymbol{x}\|_{\mathbb{V}}=1} \|L\boldsymbol{x}\|_{\mathbb{W}} < \infty .$$

The collection of all bounded linear maps from $\mathbb{V}$ to $\mathbb{W}$ is denoted by $\mathscr{B}(\mathbb{V}, \mathbb{W})$. When $\mathbb{V} = \mathbb{W}$, we write $\mathscr{B}(\mathbb{V})$ instead of $\mathscr{B}(\mathbb{V}, \mathbb{V})$. When the underlying spaces $\mathbb{V}, \mathbb{W}$ are clear to us, sometimes we simply use $\|L\|$ to denote the norm $\|L\|_{\mathscr{B}(\mathbb{V},\mathbb{W})}$.

**Definition 3.46** (Dual Spaces)**.** Let $(X, \|\cdot\|_X)$ be a Banach space over scalar field $\mathbb{F}$. The (continuous) dual space of $X$, denoted by $X^*$, is the collection of all bounded linear functionals on $X$; that is,

$$X^* = \Big\{ L \in \mathscr{L}(X, \mathbb{F}) \,\Big|\, \sup_{\|\boldsymbol{x}\|_X=1} \big|L(\boldsymbol{x})\big| < \infty \Big\} .$$

**Theorem 3.47.** *If* $\big(\mathbb{H}, \langle\cdot,\cdot\rangle\big)$ *is a finite dimensional Hilbert space over field* $\mathbb{F}$*, then* $\mathbb{H}^*$ *is also finite dimensional and* $\dim(\mathbb{H}) = \dim(\mathbb{H}^*)$*.*

*Proof.* Let $\{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_n\}$ be an orthonormal basis of $\mathbb{H}$ (one can always find an orthonormal basis through the Gram-Schmidt process). For each $1 \leqslant k \leqslant n$, define $\varphi_k : \mathbb{H} \to \mathbb{F}$ by

$$\varphi_k(\boldsymbol{x}) = \langle \mathbf{e}_k, \boldsymbol{x} \rangle .$$

The Cauchy-Schwarz inequality (3.3) then implies that

$$\big|\varphi_k(\boldsymbol{x})\big| \leqslant \|\mathbf{e}_k\| \cdot \|\boldsymbol{x}\| = \|\boldsymbol{x}\| \qquad \forall\, \boldsymbol{x} \in \mathbb{H} ;$$

thus $\varphi_k \in \mathbb{H}^*$ for each $1 \leqslant k \leqslant n$. Moreover, if $\alpha_1, \cdots, \alpha_n$ are numbers in $\mathbb{F}$ and $\alpha_1\varphi_1 + \alpha_2\varphi_2 + \cdots + \alpha_n\varphi_n = 0$ or to be more precise,

$$\alpha_1\varphi_1(\boldsymbol{x}) + \alpha_2\varphi_2(\boldsymbol{x}) + \cdots + \alpha_n\varphi_n(\boldsymbol{x}) = 0 \qquad \forall\, \boldsymbol{x} \in \mathbb{H} ,$$

then for each $1 \leqslant j \leqslant n$, the fact that $\varphi_k(\mathbf{e}_j) = \delta_{jk}$ (the Kronecker delta) implies that

$$0 = \alpha_1\varphi_1(\mathbf{e}_j) + \alpha_2\varphi_2(\mathbf{e}_j) + \cdots + \alpha_n\varphi_n(\mathbf{e}_j) = \sum_{k=1}^{n} \alpha_k\delta_{jk} = \alpha_k .$$

Therefore, $\{\varphi_1, \varphi_2, \cdots, \varphi_n\}$ is a linear independent set.

Finally, by the fact that $\boldsymbol{x} = \sum\limits_{k=1}^{n} \langle \mathbf{e}_k, \boldsymbol{x} \rangle\, \mathbf{e}_k$ for all $\boldsymbol{x} \in \mathbb{H}$, we find that for $f \in \mathbb{H}^*$,

$$f(\boldsymbol{x}) = f\Big( \sum_{k=1}^{n} \langle \mathbf{e}_k, \boldsymbol{x} \rangle \mathbf{e}_k \Big) = \sum_{k=1}^{n} f(\mathbf{e}_k)\varphi_k(\boldsymbol{x}) \qquad \forall\, \boldsymbol{x} \in \mathbb{H} .$$

This implies that $\{\varphi_1, \varphi_2, \cdots, \varphi_n\}$ is a basis of $\mathbb{H}^*$; thus $\dim(\mathbb{H}^*) = n$. ▫

**Theorem 3.48** (Riesz Representation)**.** *Let* $(\mathbb{H}, \langle\cdot,\cdot\rangle)$ *be a Hilbert space. Then every* $L \in \mathbb{H}^*$ *corresponds to a unique* $\boldsymbol{y} \in \mathbb{H}$ *such that* $L(\boldsymbol{x}) = \langle \boldsymbol{y}, \boldsymbol{x} \rangle$ *for all* $\boldsymbol{x} \in \mathbb{H}$*. In other words, there exists a bijection* $\varphi : \mathbb{H}^* \to \mathbb{H}$ *such that*

$$L(\boldsymbol{x}) = \langle \varphi(L), \boldsymbol{x} \rangle \qquad \forall\, \boldsymbol{x} \in \mathbb{H} .$$

*Moreover,* $\|\varphi(L)\| = \|L\|_{\mathscr{B}(\mathbb{H},\mathbb{F})}$ *for all* $L \in \mathbb{H}^*$*.*

*Proof.* W.L.O.G., we assume that $L$ is not the zero map (for otherwise we can choose $\boldsymbol{y} = \boldsymbol{0}$). Let $\|\cdot\|$ denote the norm induced by the inner product; that is, $\|\boldsymbol{v}\| = \sqrt{\langle \boldsymbol{v}, \boldsymbol{v}\rangle}$ for all $\boldsymbol{v} \in \mathbb{H}$.

Let $N$ be the null space of $L$; that is, $N = L^{-1}(\{0\})$. Then $N$ is closed, so Corollary 3.43 implies that $N^{\perp}$, the orthogonal complement of $N$, has a non-zero element $\boldsymbol{z}$ with $\|\boldsymbol{z}\| = 1$. Such $z$ verifies the identity that

$$L\big(L(\boldsymbol{x})\boldsymbol{z} - L(\boldsymbol{z})\boldsymbol{x}\big) = L(\boldsymbol{x})L(\boldsymbol{z}) - L(\boldsymbol{z})L(\boldsymbol{x}) = 0 \qquad \forall\, \boldsymbol{x} \in \mathbb{H}\,.$$

In other words, the vector $L(\boldsymbol{x})\boldsymbol{z} - L(\boldsymbol{z})\boldsymbol{x} \in N$ for all $\boldsymbol{x} \in \mathbb{H}$. Therefore, for each $\boldsymbol{x} \in \mathbb{H}$,

$$0 = \langle \boldsymbol{z}, L(\boldsymbol{x})\boldsymbol{z} - L(\boldsymbol{z})\boldsymbol{x}\rangle = L(\boldsymbol{x})\|\boldsymbol{z}\|^2 - L(\boldsymbol{z})\langle \boldsymbol{z}, \boldsymbol{x}\rangle = L(\boldsymbol{x}) - L(\boldsymbol{z})\langle \boldsymbol{z}, \boldsymbol{x}\rangle$$

so that letting $\boldsymbol{y} = \overline{L(\boldsymbol{z})}\boldsymbol{z}$, we have

$$L(\boldsymbol{x}) = \langle \boldsymbol{y}, \boldsymbol{x}\rangle \qquad \forall\, \boldsymbol{x} \in \mathbb{H}\,.$$

Suppose that $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathbb{H}$ satisfy $L(\boldsymbol{x}) = \langle \boldsymbol{y}_1, \boldsymbol{x}\rangle = \langle \boldsymbol{y}_2, \boldsymbol{x}\rangle$ for all $\boldsymbol{x} \in \mathbb{H}$. Then

$$\langle \boldsymbol{y}_1 - \boldsymbol{y}_2, \boldsymbol{x}\rangle = 0 \qquad \forall\, \boldsymbol{x} \in \mathbb{H}\,.$$

In particular, letting $\boldsymbol{x} = \boldsymbol{y}_1 - \boldsymbol{y}_2$ in the identity above we find that $\|\boldsymbol{y}_1 - \boldsymbol{y}_2\| = 0$; thus the property of norms shows that $\boldsymbol{y}_1 = \boldsymbol{y}_2$. Therefore, each $L \in \mathbb{H}^*$ corresponds to a unique $\boldsymbol{y} \in \mathbb{H}$ satisfying $L(\boldsymbol{x}) = \langle \boldsymbol{y}, \boldsymbol{x}\rangle$ for all $\boldsymbol{x} \in \mathbb{H}$.

Finally, using the identity that $\|\boldsymbol{y}\| = \sup\limits_{\|\boldsymbol{x}\|=1} \big|\langle \boldsymbol{y}, \boldsymbol{x}\rangle\big|$, we find that

$$\|\varphi(L)\| = \sup_{\|\boldsymbol{x}\|=1} \big|\langle \varphi(L), \boldsymbol{x}\rangle\big| = \sup_{\|\boldsymbol{x}\|=1} \big|L(\boldsymbol{x})\big| = \|L\|_{\mathscr{B}(\mathbb{H},\mathbb{F})}\,. \qquad \square$$

**Remark 3.49.** Let $(\mathbb{H}, \langle\cdot,\cdot\rangle)$ be a Hilbert space, and $\varphi$ be the map given in Theorem 3.48. Define

$$\langle L_1, L_2\rangle_{\mathbb{H}^*} = \big\langle \varphi(L_1), \varphi(L_2)\big\rangle \qquad \forall\, L_1, L_2 \in \mathbb{H}^*\,.$$

Then $\big(\mathbb{H}^*, \langle\cdot,\cdot\rangle_{\mathbb{H}^*}\big)$ is a Hilbert space, and $\|\cdot\|_{\mathscr{B}(\mathbb{H},\mathbb{F})}$ is the norm induced by the inner product given above. The operator norm $\|\cdot\|_{\mathscr{B}(\mathbb{H},\mathbb{F})}$ sometimes is denoted by $\|\cdot\|_{H^*}$.

Let $(\mathbb{V}, \langle\cdot,\cdot\rangle_{\mathbb{V}})$ and $(\mathbb{W}, \langle\cdot,\cdot\rangle_{\mathbb{W}})$ be Hilbert spaces over a common scalar field $\mathbb{F}$, where $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$, and $A \in \mathscr{B}(\mathbb{V}, \mathbb{W})$. Note that the boundedness of $A$ implies that

$$\|A\boldsymbol{v}\|_{\mathbb{W}} \leqslant \|A\|_{\mathscr{B}(\mathbb{V},\mathbb{W})}\|\boldsymbol{v}\|_{\mathbb{V}} < \infty \qquad \forall\, \boldsymbol{v} \in \mathbb{V}\,.$$

For a given $\boldsymbol{w} \in \mathbb{W}$, define $L : \mathbb{V} \to \mathbb{F}$ by

$$L(\boldsymbol{v}) = \langle \boldsymbol{w}, A\boldsymbol{v} \rangle_{\mathbb{W}} .$$

Then $L \in \mathbb{V}'$ (the algebraic dual space of $\mathbb{W}$) and the Cauchy-Schwarz inequality (3.3) implies that

$$\big|L(\boldsymbol{v})\big| \leqslant \|\boldsymbol{w}\|_{\mathbb{W}} \|A\boldsymbol{v}\|_{\mathbb{W}} \leqslant \|\boldsymbol{w}\|_{\mathbb{W}} \|A\|_{\mathscr{B}(\mathbb{V},\mathbb{W})} \|\boldsymbol{v}\|_{\mathbb{V}}$$

so that

$$\sup_{\|\boldsymbol{v}\|_{\mathbb{V}}=1} \big|L(\boldsymbol{v})\big| \leqslant \|A\|_{\mathscr{B}(\mathbb{V},\mathbb{W})} \|\boldsymbol{w}\|_{\mathbb{W}} < \infty .$$

Therefore, $L \in \mathbb{V}^*$ (the continuous dual space of $\mathbb{W}$). By the Riesz representation theorem (Theorem 3.48), there exists a unique vector $\boldsymbol{u} \in \mathbb{V}$ such that

$$L(\boldsymbol{v}) = \langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\mathbb{V}} \qquad \forall\, \boldsymbol{v} \in \mathbb{V} .$$

The map $\boldsymbol{w} \mapsto \boldsymbol{u}$ is denoted by $A^*$ (so $A^* : \mathbb{W} \to \mathbb{V}$), and $A^*$ is called the adjoint operator of $A$. We note that $A^*$ satisfies that

$$\langle \boldsymbol{w}, A\boldsymbol{v} \rangle_{\mathbb{W}} = \langle A^*(\boldsymbol{w}), \boldsymbol{v} \rangle_{\mathbb{V}} \qquad \forall\, \boldsymbol{v} \in \mathbb{V}, \boldsymbol{w} \in \mathbb{W}$$

so that for all $\boldsymbol{v} \in \mathbb{V}$ and $\boldsymbol{w}_1, \boldsymbol{w}_2$ in $\mathbb{W}$,

$$\begin{aligned} \big\langle A^*(\lambda \boldsymbol{w}_1 + \mu \boldsymbol{w}_2), \boldsymbol{v} \big\rangle_{\mathbb{V}} &= \langle \lambda \boldsymbol{w}_1 + \mu \boldsymbol{w}_2, A\boldsymbol{v} \rangle_{\mathbb{W}} = \bar{\lambda} \langle \boldsymbol{w}_1, A\boldsymbol{v} \rangle_{\mathbb{W}} + \bar{\mu} \langle \boldsymbol{w}_2, A\boldsymbol{v} \rangle_{\mathbb{W}} \\ &= \bar{\lambda} \big\langle A^*(\boldsymbol{w}_1), \boldsymbol{v} \big\rangle_{\mathbb{V}} + \bar{\mu} \big\langle A^*(\boldsymbol{w}_2), \boldsymbol{v} \big\rangle_{\mathbb{V}} = \big\langle \lambda A^*(\boldsymbol{w}_1) + \mu A^*(\boldsymbol{w}_2), \boldsymbol{v} \big\rangle_{\mathbb{V}} . \end{aligned}$$

Therefore,

$$A^*(\lambda \boldsymbol{w}_1 + \mu \boldsymbol{w}_2) = \lambda A^*(\boldsymbol{w}_1) + \mu A^*(\boldsymbol{w}_2) \qquad \forall\, \boldsymbol{w}_1, \boldsymbol{w}_2 \in \mathbb{H} ;$$

thus $A^* \in \mathscr{L}(\mathbb{W}, \mathbb{V})$. Moreover,

$$\begin{aligned} \|A^*\|_{\mathscr{B}(\mathbb{W},\mathbb{V})} &= \sup_{\|\boldsymbol{w}\|_{\mathbb{W}}=1} \sup_{\|\boldsymbol{v}\|_{\mathbb{V}}=1} \big|\langle A^*\boldsymbol{w}, \boldsymbol{v} \rangle_{\mathbb{V}}\big| = \sup_{\|\boldsymbol{w}\|_{\mathbb{W}}=1} \sup_{\|\boldsymbol{v}\|_{\mathbb{V}}=1} \big|\langle \boldsymbol{w}, A\boldsymbol{v} \rangle_{\mathbb{W}}\big| \\ &= \sup_{\|\boldsymbol{v}\|_{\mathbb{V}}=1} \sup_{\|\boldsymbol{w}\|_{\mathbb{W}}=1} \big|\langle \boldsymbol{w}, A\boldsymbol{v} \rangle_{\mathbb{W}}\big| = \|A\|_{\mathscr{B}(\mathbb{V},\mathbb{W})} ; \end{aligned}$$

thus $A^*$ is indeed bounded.

**Definition 3.50.** Let $(\mathbb{V}, \langle \cdot, \cdot \rangle_{\mathbb{V}})$ and $(\mathbb{W}, \langle \cdot, \cdot \rangle_{\mathbb{W}})$ be Hilbert spaces over a common scalar field $\mathbb{F}$, where $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$, and $A \in \mathscr{B}(\mathbb{V}, \mathbb{W})$. The adjoint operator of $A$, denoted by $A^*$, is the unique element in $\mathscr{B}(\mathbb{W}, \mathbb{V})$ satisfying that

$$\langle \boldsymbol{w}, A\boldsymbol{v} \rangle_{\mathbb{W}} = \langle A^*\boldsymbol{w}, \boldsymbol{v} \rangle_{\mathbb{V}} \qquad \forall\, \boldsymbol{v} \in \mathbb{V}, \boldsymbol{w} \in \mathbb{W} .$$

**Remark 3.51.** The adjoint operator can be defined for general linear operator (which may be unbounded) as follows. Let $(X, \|\cdot\|_X)$ and $(Y, \|\cdot\|_Y)$ be normed spaces, and $A : D(A) \to Y$, where $D(A)$ is a dense subset of $X$ called the domain of $A$. The adjoint operator of $A$, denoted by $A^*$, is an operator from a subset of $Y^*$ to $X^*$ satisfying

$$\langle A^* \boldsymbol{y}^*, \boldsymbol{x} \rangle = \langle y^*, A\boldsymbol{x} \rangle \qquad \forall\, \boldsymbol{x} \in D(A), \boldsymbol{y}^* \in D(A^*)\,,$$

where $\langle \cdot, \cdot \rangle$ denotes the duality pairing (a fancy way to express linear functionals in functional analysis), and $D(A^*)$, the domain of $A^*$, is the set

$$D(A^*) = \left\{ \boldsymbol{y}^* \in Y^* \,\middle|\, \exists\, C > 0 \ni \left| \langle \boldsymbol{y}^*, A\boldsymbol{x} \rangle \right| \leqslant C \|\boldsymbol{x}\|_X \text{ for all } \boldsymbol{x} \in D(A) \right\}.$$

**Remark 3.52.** Let $(\mathbb{V}, \langle \cdot, \cdot \rangle_{\mathbb{V}})$ and $(\mathbb{W}, \langle \cdot, \cdot \rangle_{\mathbb{W}})$ be Hilbert spaces over a common scalar field $\mathbb{F}$, where $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$, and $A \in \mathscr{B}(\mathbb{V}, \mathbb{W})$. By the property of inner product,

$$\langle A\boldsymbol{v}, \boldsymbol{w} \rangle_{\mathbb{W}} = \overline{\langle \boldsymbol{w}, A\boldsymbol{v} \rangle_{\mathbb{W}}} = \overline{\langle A^*\boldsymbol{w}, \boldsymbol{v} \rangle_{\mathbb{V}}} = \langle \boldsymbol{v}, A^*\boldsymbol{w} \rangle_{\mathbb{V}} \quad \forall\, \boldsymbol{v} \in \mathbb{V}, \boldsymbol{w} \in \mathbb{W}\,.$$

Therefore, the adjoint operator $A^*$ of $A$ satisfies

$$\langle \boldsymbol{w}, A\boldsymbol{v} \rangle_{\mathbb{W}} = \langle A^*\boldsymbol{w}, \boldsymbol{v} \rangle_{\mathbb{V}} \quad \text{and} \quad \langle A\boldsymbol{v}, \boldsymbol{w} \rangle_{\mathbb{W}} = \langle \boldsymbol{v}, A^*\boldsymbol{w} \rangle_{\mathbb{V}} \qquad \forall\, \boldsymbol{w}, \boldsymbol{v} \in \mathbb{H}\,. \tag{3.7}$$

**Proposition 3.53.** *Let $(\mathbb{V}, \langle \cdot, \cdot \rangle_{\mathbb{V}})$ and $(\mathbb{W}, \langle \cdot, \cdot \rangle_{\mathbb{W}})$ be Hilbert spaces over a common scalar field $\mathbb{F}$, where $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$, and $A \in \mathscr{B}(\mathbb{V}, \mathbb{W})$. Then $(A^*)^* = A$.*

*Proof.* Let $\boldsymbol{v} \in \mathbb{V}$ be given. Then if $\boldsymbol{w} \in \mathbb{W}$, using (3.7) we find that

$$\langle A\boldsymbol{v}, \boldsymbol{w} \rangle_{\mathbb{W}} = \langle \boldsymbol{v}, A^*\boldsymbol{w} \rangle_{\mathbb{V}} = \left\langle (A^*)^*\boldsymbol{v}, \boldsymbol{w} \right\rangle_{\mathbb{W}}.$$

Therefore, $A\boldsymbol{v} = (A^*)^*\boldsymbol{v}$ for all $\boldsymbol{v} \in \mathbb{H}$; thus $A = (A^*)^*$. ▫

**• Matrix representation of adjoint operators**

Let $(\mathbb{V}, \langle \cdot, \cdot \rangle_{\mathbb{V}})$ and $(\mathbb{W}, \langle \cdot, \cdot \rangle_{\mathbb{W}})$ be two finite dimensional inner product spaces over $\mathbb{C}$, $\mathcal{B} = \{\mathbf{v}_1, \cdots, \mathbf{v}_n\}$ and $\widetilde{\mathcal{B}} = \{\mathbf{w}_1, \cdots, \mathbf{w}_m\}$ be orthonormal basis of $\mathbb{V}$ and $\mathbb{W}$, respectively, and $L \in \mathscr{L}(\mathbb{V}; \mathbb{W})$. Using (3.6), we find that the matrix representation of $L$ and $L^*$ satisfy

$$\begin{aligned} &\text{the } (i,j)\text{-entry of } [L^*]_{\mathcal{B},\widetilde{\mathcal{B}}} \\ &\quad = \left\langle [\mathbf{v}_i]_{\mathcal{B}}, [L^*]_{\mathcal{B},\widetilde{\mathcal{B}}} [\mathbf{w}_j]_{\widetilde{\mathcal{B}}} \right\rangle_{\mathbb{C}^n} = \langle \mathbf{v}_i, L^*\mathbf{w}_j \rangle_{\mathbb{V}} = \langle L\mathbf{v}_i, \mathbf{w}_j \rangle_{\mathbb{W}} = \overline{\langle \mathbf{w}_j, L\mathbf{v}_i \rangle}_{\mathbb{W}} \\ &\quad = \text{the complex conjugate of the } (j,i)\text{-entry of } [L]_{\widetilde{\mathcal{B}},\mathcal{B}}. \end{aligned}$$

This observation motivates the following

**Definition 3.54** (Conjugate transpose of matrices)**.** Let $A = [a_{ij}]_{m\times n}$ be an $m \times n$ complex matrix. The conjugate transpose of $A$, denoted by $A^{\mathrm{H}}$, $A^*$ or $A^\dagger$ (the last one is often used in quantum mechanics), is an $n \times m$ matrix $[b_{ij}]_{n\times m}$ given by $b_{ij} = \overline{a_{ji}}$. In other words,

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}^\dagger = \begin{bmatrix} \overline{a_{11}} & \overline{a_{21}} & \cdots & \overline{a_{m1}} \\ \overline{a_{12}} & \overline{a_{22}} & \cdots & \overline{a_{m2}} \\ \vdots & & & \vdots \\ \overline{a_{1n}} & \overline{a_{2n}} & \cdots & \overline{a_{mn}} \end{bmatrix}.$$

**Remark 3.55.** For real matrices, the conjugate transpose is just the transpose.

**Theorem 3.56.** *Let $(\mathbb{V}, \langle\cdot,\cdot\rangle_{\mathbb{V}})$ and $(\mathbb{W}, \langle\cdot,\cdot\rangle_{\mathbb{W}})$ be finite dimensional inner product spaces over $\mathbb{C}$, $\mathcal{B}$ and $\widetilde{\mathcal{B}}$ be orthonormal basis of $\mathbb{V}$ and $\mathbb{W}$, respectively. If $L \in \mathscr{L}(\mathbb{V};\mathbb{W})$, then $[L^*]_{\mathcal{B},\widetilde{\mathcal{B}}} = [L]^\dagger_{\widetilde{\mathcal{B}},\mathcal{B}}$.*

**Definition 3.57.** Let $A = [a_{ij}]$ be a square matrix.

1. $A$ is said to be ***Hermitian*** if $A = A^\dagger$.
2. $A$ is said to be ***skew Hermitian*** if $A^\dagger = -A$.
3. $A$ is said to be ***normal*** if $AA^\dagger = A^\dagger A$.
4. $A$ is said to be ***unitary*** if $A^{-1} = A^\dagger$ (explained in Section 3.5).

## 3.5 Unitary Operators and Unitary Matrices

The concept of unitary operators is a generalization of orthogonal matrices that have the property that $\mathrm{O}^{-1} = \mathrm{O}^{\mathrm{T}}$.

### 3.5.1 Unitary operators

**Definition 3.58.** Let $\big(\mathbb{H}, \langle\cdot,\cdot\rangle\big)$ be a Hilbert space, and $U \in \mathscr{B}(\mathbb{H})$.

1. $U$ is said to be ***self-adjoint*** if $U^* = U$.
2. $U$ is said to be ***unitary*** if $UU^* = U^*U = \mathrm{Id}$, where Id denotes the identity map on $\mathbb{H}$.

The collection of self-adjoint operators on $\mathbb{H}$ is denoted by $\mathscr{B}_{\mathrm{sa}}(\mathbb{H})$, and the collection of unitary operators on $\mathbb{H}$ is denoted by $\mathrm{U}(\mathbb{H})$.

**Remark 3.59.** Let $\big(\mathbb{H}, \langle\cdot,\cdot\rangle\big)$ be a Hilbert space over field $\mathbb{F}$, and $U \in \mathscr{B}(\mathbb{H})$.

1. If $\mathbb{F} = \mathbb{R}$, then $U$ satisfying $UU^* = U^*U = \mathrm{Id}$ is often called **orthogonal** instead of unitary. Therefore, when the term "unitary" is used, we often assume that $\mathbb{F} = \mathbb{C}$.

2. If $U$ is unitary, then $U^*$ is also unitary.

**Theorem 3.60.** *Let $\big(\mathbb{H}, \langle\cdot,\cdot\rangle\big)$ be a Hilbert space over field $\mathbb{C}$, and $U \in \mathscr{B}(\mathbb{H})$. The following three statements are equivalent.*

1. *$U$ is unitary.*

2. *$U$ is surjective and $\|U\boldsymbol{x}\| = \|\boldsymbol{x}\|$ for all $\boldsymbol{x} \in \mathbb{H}$.*

3. *$U$ is surjective and $\langle U\boldsymbol{x}, U\boldsymbol{y}\rangle = \langle \boldsymbol{x}, \boldsymbol{y}\rangle$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{H}$.*

*Proof.* "1 $\Rightarrow$ 2": Let $\boldsymbol{z} \in \mathbb{H}$ be given. Then $\boldsymbol{y} = U^*\boldsymbol{z} \in \mathbb{H}$ satisfies $U\boldsymbol{y} = \boldsymbol{z}$. This implies that $U$ is surjective. Moreover, if $\boldsymbol{x} \in \mathbb{H}$ be given, then

$$\|\boldsymbol{x}\|^2 = \langle \boldsymbol{x}, \boldsymbol{x}\rangle = \langle \boldsymbol{x}, U^*U\boldsymbol{x}\rangle = \langle U\boldsymbol{x}, U\boldsymbol{x}\rangle = \|U\boldsymbol{x}\|^2 ;$$

thus $\|U\boldsymbol{x}\| = \|\boldsymbol{x}\|$ for all $\boldsymbol{x} \in \mathbb{H}$.

"2 $\Rightarrow$ 3": Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{H}$. Then

$$\begin{aligned}
\|U(\boldsymbol{x}+\boldsymbol{y})\|^2 &= \langle U(\boldsymbol{x}+\boldsymbol{y}), U(\boldsymbol{x}+\boldsymbol{y})\rangle = \|U\boldsymbol{x}\|^2 + \langle U\boldsymbol{x}, U\boldsymbol{y}\rangle + \langle U\boldsymbol{y}, U\boldsymbol{x}\rangle + \|U\boldsymbol{y}\|^2 \\
&= \|U\boldsymbol{x}\|^2 + 2\mathrm{Re}(\langle U\boldsymbol{x}, U\boldsymbol{y}\rangle) + \|U\boldsymbol{y}\|^2 , \\
\|\boldsymbol{x}+\boldsymbol{y}\|^2 &= \|\boldsymbol{x}\|^2 + \langle \boldsymbol{x}, \boldsymbol{y}\rangle + \langle \boldsymbol{y}, \boldsymbol{x}\rangle + \|\boldsymbol{y}\|^2 = \|\boldsymbol{x}\|^2 + 2\mathrm{Re}(\langle \boldsymbol{x}, \boldsymbol{y}\rangle) + \|\boldsymbol{y}\|^2 ,
\end{aligned}$$

and

$$\begin{aligned}
\|U(\boldsymbol{x}+i\boldsymbol{y})\|^2 &= \langle U(\boldsymbol{x}+i\boldsymbol{y}), U(\boldsymbol{x}+i\boldsymbol{y})\rangle = \|U\boldsymbol{x}\|^2 + i\langle U\boldsymbol{x}, U\boldsymbol{y}\rangle - i\langle U\boldsymbol{y}, U\boldsymbol{x}\rangle + \|U\boldsymbol{y}\|^2 \\
&= \|U\boldsymbol{x}\|^2 - 2\mathrm{Im}(\langle U\boldsymbol{x}, U\boldsymbol{y}\rangle) + \|U\boldsymbol{y}\|^2 , \\
\|\boldsymbol{x}+i\boldsymbol{y}\|^2 &= \|\boldsymbol{x}\|^2 + i\langle \boldsymbol{x}, \boldsymbol{y}\rangle - i\langle \boldsymbol{y}, \boldsymbol{x}\rangle + \|\boldsymbol{y}\|^2 = \|\boldsymbol{x}\|^2 - 2\mathrm{Im}(\langle \boldsymbol{x}, \boldsymbol{y}\rangle) + \|\boldsymbol{y}\|^2 .
\end{aligned}$$

Since $\|U\boldsymbol{x}\| = \|\boldsymbol{x}\|$ for all $\boldsymbol{x} \in \mathbb{H}$, we have

$$\mathrm{Re}(\langle U\boldsymbol{x}, U\boldsymbol{y}\rangle) = \mathrm{Re}(\langle \boldsymbol{x}, \boldsymbol{y}\rangle) \qquad \text{and} \qquad \mathrm{Im}(\langle U\boldsymbol{x}, U\boldsymbol{y}\rangle) = \mathrm{Im}(\langle \boldsymbol{x}, \boldsymbol{y}\rangle) .$$

Therefore, $\langle U\boldsymbol{x}, U\boldsymbol{y}\rangle = \langle \boldsymbol{x}, \boldsymbol{y}\rangle$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{H}$.

"3 $\Rightarrow$ 1": Let $\boldsymbol{x} \in \mathbb{H}$ be given. Then

$$\langle U^*U\boldsymbol{x}, \boldsymbol{y}\rangle = \langle U\boldsymbol{x}, U\boldsymbol{y}\rangle = \langle \boldsymbol{x}, \boldsymbol{y}\rangle \qquad \forall\, \boldsymbol{y} \in \mathbb{H}\,;$$

thus $U^*U\boldsymbol{x} = \boldsymbol{x}$. This implies that $U^*U = \mathrm{Id}$ on $\mathbb{H}$.

On the other hand, since $U$ is surjective, for each $\boldsymbol{y} \in \mathbb{H}$ there exists $\boldsymbol{x} \in \mathbb{H}$ such that $U\boldsymbol{x} = \boldsymbol{y}$. Using $U^*U = \mathrm{Id}$, this $\boldsymbol{x}$ must be $U^*\boldsymbol{y}$; thus $UU^*\boldsymbol{y} = \boldsymbol{y}$ for all $\boldsymbol{y} \in \mathbb{H}$. This shows that $UU^* = \mathrm{Id}$ on $\mathbb{H}$; thus $U$ is unitary. ▫

**Corollary 3.61.** *Let $\big(\mathbb{H}, \langle\cdot,\cdot\rangle\big)$ be a Hilbert space over field $\mathbb{C}$. If $U \in \mathrm{U}(\mathbb{H})$, then $\|U\| = 1$.*

**Definition 3.62.** Let $(X, \|\cdot\|)$ be a Banach space, and $T \in \mathscr{B}(X)$. The spectrum of $T$, denoted by $\sigma(T)$, is the collection of all $\lambda \in \mathbb{C}$ for which the operator $T - \lambda\mathrm{Id}$ is not invertible. In other words,

$$\sigma(T) = \big\{\lambda \in \mathbb{C} \,\big|\, (T - \lambda\mathrm{Id}) \text{ is not bijective}\big\}\,.$$

A number $\lambda \in \sigma(T)$ is called an eigenvalue of $T$ if $T - \lambda\mathrm{Id}$ is not one-to-one. The collection of all eigenvalues of $T$ is called the point spectrum of $T$ and is denoted by $\sigma_p(T)$.

**Theorem 3.63.** *Let $\big(\mathbb{H}, \langle\cdot,\cdot\rangle\big)$ be a Hilbert space, $U \in \mathrm{U}(\mathbb{H})$, and $\lambda$ be an eigenvalue of $U$. Then $|\lambda| = 1$.*

*Proof.* Let $\lambda$ be an eigenvalue of $U$. Then there exists a non-zero vector $\boldsymbol{x} \in \mathbb{H}$ such that $U\boldsymbol{x} = \lambda\boldsymbol{x}$. Therefore,

$$\|U\boldsymbol{x}\| = |\lambda|\|\boldsymbol{x}\|\,,$$

and the theorem is concluded by Theorem 3.60 and the fact that $\boldsymbol{x} \neq \boldsymbol{0}$. ▫

### 3.5.2 Unitary matrices

**Definition 3.64.** A unitary matrix $A$ is the matrix representation of some unitary map $U : \mathbb{H} \to \mathbb{H}$, where $\mathbb{H}$ is a finite dimensional inner product space over $\mathbb{C}$, relative to an orthonormal basis of $\mathbb{H}$.

By the definition of unitary maps, Theorem 3.19 and 3.56 provide an alternative definition of unitary matrices that we state as follows.

**Definition 3.65** (Alternative Definition of Unitary Matrices)**.** A square matrix $A$ is said to be unitary if $AA^\dagger = A^\dagger A = \mathrm{I}$. The collection of all $n \times n$ unitary matrices is denoted by $\mathrm{U}(n)$.

**Corollary 3.66.** *If $A \in \mathrm{U}(n)$, then $A^{-1} = A^{\dagger}$.*

**Corollary 3.67.** *If $A \in \mathrm{U}(n)$, then $\left|\det(A)\right| = 1$.*

**Definition 3.68.** The special unitary group of degree $n$, denoted by $\mathrm{SU}(n)$, is the collection of $n \times n$ unitary matrices with determinant 1. An element in $\mathrm{SU}(n)$ is called a special unitary matrix.

**Definition 3.69** (Walsh-Hadamard Matrix)**.** For $m \in \mathbb{N} \cup \{0\}$, the Walsh-Hadamard matrix $\mathrm{H}_m$ is a $2^m \times 2^m$ matrix defined recursively by

1. $\mathrm{H}_0 = 1$; 2. $\mathrm{H}_m = \dfrac{1}{\sqrt{2}} \begin{bmatrix} \mathrm{H}_{m-1} & \mathrm{H}_{m-1} \\ \mathrm{H}_{m-1} & -\mathrm{H}_{m-1} \end{bmatrix}$ for all $m \in \mathbb{N}$.

We note that $\mathrm{H}_m$ is symmetric and orthogonal/unitary for all $m \in \mathbb{N}$ (which can be proved by induction).

**Remark 3.70.** The original definition of the Hadamard matrix (of order $2^m$), denoted by $\mathrm{H}_m$, is a $2^m \times 2^m$ matrix defined recursively by

1. $\mathrm{H}_0 = 1$; 2. $\mathrm{H}_m = \begin{bmatrix} \mathrm{H}_{m-1} & \mathrm{H}_{m-1} \\ \mathrm{H}_{m-1} & -\mathrm{H}_{m-1} \end{bmatrix}$ for all $m \in \mathbb{N}$.

However, in quantum computing we usually only consider unitary matrices, so the factor $\dfrac{1}{\sqrt{2}}$ is to normalized the original Hadamard matrices so that the norm of each colum (and also each row) all become 1. Therefore, the Hadamard matrices given in Definition 3.69 is sometimes called the normalized Hadamard matrices.

**Remark 3.71.** Let the $(k, \ell)$-entry of $\mathrm{H}_m$ be denoted by $h_{k\ell}$; that is, $\mathrm{H}_m = [h_{k\ell}]_{1 \leqslant k, \ell \leqslant 2^m}$. Then

$$h_{k\ell} = 2^{-\frac{m}{2}} (-1)^{(k-1) \,\bullet\, (\ell-1)},$$

where the bitwise dot product $\bullet$ of two numbers $k$ and $\ell$ is given by

$$k \bullet \ell = \sum_{j=1}^{m} k_j \ell_j = k_1\ell_1 + k_2\ell_2 + \cdots + k_m\ell_m \tag{3.8}$$

if $k = (k_1 k_2 \cdots k_m)_2$ and $\ell = (\ell_1 \ell_2 \cdots \ell_m)_2$.

In matlab®, the bitwise dot product of $x$ and $y$ can be computed by

$$x \bullet y = \mathbf{de2bi}(x, n) * \mathbf{de2bi}(y, n)'$$

if both $x$ and $y$ can be expressed as $n$ bits binary numbers.

**Exercise 3.72.** For matrices $A = [a_{k\ell}]$ and $B = [b_{k\ell}]$ of the same size $m \times n$, define the Hadamard product of $A$ and $B$, denoted by $A \odot B$, as an $m \times n$ matrix whose $(k, \ell)$-entry is give by $a_{k\ell} b_{k\ell}$; that is,

$$C = A \odot B\,, \quad C = [c_{k\ell}]\,, \quad c_{k\ell} = a_{k\ell} b_{k\ell}\,. \tag{3.9}$$

In matlab®, the Hadamard product of $A$ and $B$ can be computed by $A \odot B = A .* B$. **In the following, we will always use $.*$ to denote the Hadamard product.**

Let $\mathrm{M}_n = \sqrt{2}^n \mathrm{H}_n$ be the **unnormalized** Hadamard matrix whose $(k, \ell)$-entry is given by $(-1)^{(k-1)\bullet(\ell-1)}$, and $\boldsymbol{r}_j$ be the $(j+1)$-th row of $\mathrm{M}_n$. Define $\varphi : \{0,1\}^n \to \{\boldsymbol{r}_0, \boldsymbol{r}_1, \cdots, \boldsymbol{r}_{2^n-1}\}$ by

$$\varphi(j_1, j_2, \cdots, j_n) = \boldsymbol{r}_j \quad \text{if} \quad j = (j_1 j_2 \cdots j_n)_2\,.$$

Show that $\varphi : (\{0,1\}^n, \oplus) \to \big(\{\boldsymbol{r}_0, \boldsymbol{r}_1, \cdots, \boldsymbol{r}_{2^n-1}\},\ .*\big)$ is a group isomorphism, where $\oplus$ is the element-wise addition in $\mathbb{Z}_2$; that is,

$$(x_1, x_2, \cdots, x_n) \oplus (y_1, y_2, \cdots, y_n) = (x_1 \oplus y_1, x_2 \oplus y_2, \cdots, x_n \oplus y_n)\,.$$

In other words, show that $\varphi : \{0,1\}^n \to \{\boldsymbol{r}_0, \boldsymbol{r}_1, \cdots, \boldsymbol{r}_{2^n-1}\}$ defined above is a bijection and

$$\varphi\big((k_1, \cdots, k_n) \oplus (\ell_1, \cdots, \ell_n)\big) = \boldsymbol{r}_k .* \boldsymbol{r}_\ell \qquad \forall\, k = (k_1 k_2 \cdots k_n)_2 \text{ and } \ell = (\ell_1 \ell_2 \cdots \ell_n)_2\,.$$

## 3.6 Quantum Mechanics

One should treat this section as an independent section.

**Definition 3.73.** A linear map $A : \mathbb{H} \to \mathbb{H}$ is called an operator on the Hilbert space $\mathbb{H}$. The set of all operators on $H$ is denoted by $\mathscr{L}(\mathbb{H})$. A linear map $T : \mathscr{L}(\mathbb{H}) \to \mathscr{L}(\mathbb{H})$; that is, an operator acting on operators, is called a super-operator. The operator $A^* : \mathbb{H} \to \mathbb{H}$ that satisfies

$$\langle A^*\psi | \phi \rangle = \langle \psi | A\phi \rangle \qquad \forall\, |\psi\rangle, |\phi\rangle \in \mathbb{H}$$

is called the adjoint operator of $A$. $A$ is called self-adjoint if $A^* = A$, and the collection of all self-adjoint operators on $\mathbb{H}$ is denoted by $\mathscr{B}_{sa}(\mathbb{H})$.

**Definition 3.74** (Spectrum)**.** Let $A$ be an operator on a Hilbert space $\mathbb{H}$. A vector $|\psi\rangle \in \mathbb{H}\backslash\{0\}$ is called an eigenvector of $A$ with eigenvalue $\lambda \in \mathbb{C}$ if

$$A|\psi\rangle = \lambda|\psi\rangle.$$

The linear subspace that is spanned by all eigenvectors for a given eigenvalue $\lambda$ of an operator $A$ is called the eigenspace of $\lambda$ and denoted by $\mathrm{Eig}(A,\lambda)$. An eigenvalue $\lambda$ is called non-degenerate if its eigenspace is one-dimensional. Otherwise, $\lambda$ is called degenerate. The set $\big\{\lambda \in \mathbb{C} \,\big|\, (A-\lambda \mathrm{I})^{-1} \text{ does not exist}\big\}$ is called the spectrum of the operator $A$ and is denoted by $\sigma(A)$.

**Definition 3.75.** Let $\mathbb{H}$ be a Hilbert space. An operator $\mathrm{P} \in \mathscr{L}(\mathbb{H})$ satisfying $\mathrm{P}^2 = \mathrm{P}$ is called a projection or projector. If in addition $\mathrm{P}^* = \mathrm{P}$, then P is called an orthogonal projection.

Let $\mathbb{H}_{\mathrm{sub}}$ be a subspace of $\mathbb{H}$. If $\mathrm{P}_{\mathrm{sub}}$ is an orthogonal projection and satisfies $\mathrm{P}_{\mathrm{sub}}|\psi\rangle = |\psi\rangle$ for all $|\psi\rangle \in \mathbb{H}_{\mathrm{sub}}$ we call $\mathrm{P}_{\mathrm{sub}}$ the projection onto this subspace.

Let $A \in \mathscr{L}(\mathbb{H})$. The projector onto the eigenspace $\mathrm{Eig}(A,\lambda)$ of $\lambda$ is denoted by $\mathrm{P}_\lambda$ (here $A$ is not presented in the expression since we usually only focus on one particular $A$).

**Definition 3.76** (Observables and Pure States)**.** An observable; that is, a physically measurable quantity of a quantum system, is represented by a self-adjoint operator on a Hilbert space $\mathbb{H}$. If the preparation of a statistical ensemble is such that for any observable represented by its self-adjoint operator $A$ the mean value of the observable can be calculated with the help of a vector $|\psi\rangle \in \mathbb{H}$ satisfying $\||\psi\rangle\|_{\mathbb{H}} = 1$ as

$$\langle A\rangle_\psi = \langle \psi | A\psi\rangle\,, \tag{3.10}$$

then the preparation is said to be described by a pure state represented by the vector $|\psi\rangle \in \mathbb{H}$. One calls $|\psi\rangle$ the state vector or simply the state, and $\langle A\rangle_\psi$ is called the (quantum mechanical) expectation value of the observable $A$ in the pure state $|\psi\rangle$.

The space $\mathbb{H}$ is said to be the Hilbert space of the quantum system.

Using the diagonal representation of any self-adjoint operator $A$ in terms of its eigenbasis $\{\mathbf{e}_1, \cdots, \mathbf{e}_n\}$, the expectation value of the observable represented by $A$ becomes

$$\langle A\rangle_\psi = \langle \psi | A\psi\rangle = \Big\langle \psi \Big| \sum_j \lambda_j \mathbf{e}_j \Big\rangle \langle \mathbf{e}_j | \psi\rangle = \sum_j \lambda_j \langle \psi | \mathbf{e}_j\rangle \langle \mathbf{e}_j | \psi\rangle = \sum_j \lambda_j \big|\langle \psi | \mathbf{e}_j\rangle\big|^2 .$$

In measurements one always observes an element of the spectrum (see Definition 3.74) of the associated operator. Since we restrict ourselves here exclusively to finite dimensional systems, for our purposes we can thus identify the eigenvalues $\{\lambda_j\}$ of a self-adjoint operator $A$ as the possible measurement results of the associated observable. In the case of a purely

non-degenerate spectrum the positive numbers $\left|\langle\psi|\mathbf{e}_j\rangle\right|^2$ are interpreted as the probabilities with which the respective value $\lambda_j$ is observed. This is formalized more generally in the following postulate.

**Postulate 3.77.** In a quantum system with Hilbert space $\mathbb{H}$ the possible measurement values of an observable are given by the spectrum $\sigma(A)$ of the operator $A \in \mathscr{B}_{sa}(\mathbb{H})$ associated with the observable. The probability $\boldsymbol{P}_\psi(\lambda)$ that for a quantum system in the pure state $|\psi\rangle \in \mathbb{H}$ a measurement of the observable yields the eigenvalue $\lambda$ of $A$ is given with the help of the projection $\mathrm{P}_\lambda$ onto the eigenspace $\mathrm{Eig}(A,\lambda)$ of $\lambda$ as

$$\boldsymbol{P}_\psi(\lambda) = \left\|\mathrm{P}_\lambda|\psi\rangle\right\|_{\mathbb{H}}^2 .$$

Let $A$ be an observable (with spectrum $\sigma(A) = \{\lambda_1, \cdots, \lambda_n\}$ and corresponding eigenbasis $\{\mathbf{e}_1, \cdots, \mathbf{e}_n\}$), $|\psi\rangle \in \mathbb{H}$ be a state vector, and $\alpha \in \mathbb{R}$. Then

$$\langle A\rangle_{e^{i\alpha}\psi} = \langle e^{i\alpha}\psi|Ae^{i\alpha}\psi\rangle = e^{-i\alpha}e^{i\alpha}\langle\psi|A\psi\rangle = \langle A\rangle_\psi ;$$

that is, the expectation values of any observable $A$ in the state $e^{i\alpha}|\psi\rangle$ and in the state $|\psi\rangle$ are the same. Since

$$\left|\langle e^{i\alpha}\psi|\mathbf{e}_j\rangle\right|^2 = \left|e^{-i\alpha}\langle\psi|\mathbf{e}_j\rangle\right|^2 = \left|\langle\psi|\mathbf{e}_j\rangle\right|^2 ,$$

the measurement probabilities in the two states are also the same. This means that physically the state $e^{i\alpha}|\psi\rangle \in \mathbb{H}$ and the state $|\psi\rangle \in \mathbb{H}$ are indistinguishable. In other words, they describe the same state.

**Definition 3.78.** Let $\mathbb{H}$ be a Hilbert space. For every $|\psi\rangle \in \mathbb{H}$ with $\||\psi\rangle\|_{\mathbb{H}} = 1$ the set

$$S_\psi \equiv \left\{e^{i\alpha}|\psi\rangle \,\middle|\, \alpha \in \mathbb{R}\right\}$$

is called a ray in $\mathbb{H}$ with $|\psi\rangle$ as a representative.

Every element of a ray $S_\psi$ describes the same physical situation. The phase $\alpha \in \mathbb{R}$ in $e^{i\alpha}$ can be arbitrarily chosen. More precisely, pure states are thus described by a representative $|\psi\rangle$ of a ray $S_\psi$ in the Hilbert space. In the designation of a state one uses only the symbol $|\psi\rangle$ of a representative of the ray, keeping in mind that $|\psi\rangle$ and $e^{i\alpha}|\psi\rangle$ are physically indistinguishable. We shall use this fact explicitly on several occasions.

Conversely, every unit vector in a Hilbert space $\mathbb{H}$ corresponds to a physical state, in other words, describes the statistics of a quantum mechanical system. If $|\phi\rangle$, $|\psi\rangle \in \mathbb{H}$ are

states, then $a|\phi\rangle + b|\psi\rangle \in \mathbb{H}$ for $a, b \in \mathbb{C}$ with $\|a|\phi\rangle + b|\psi\rangle\|_{\mathbb{H}} = 1$ is a state as well. This is the quantum mechanical superposition principle: any normalized linear combination of states is again a state and thus (in principle) a physically realizable preparation.

**Postulate 3.79.** In a quantum system with Hilbert space $\mathbb{H}$ every change of a pure state over time

$$|\psi_0\rangle : \text{state at time } t_0 \overset{\text{no measurement}}{\longrightarrow} |\psi(t)\rangle : \text{state at time } t$$

that has not been caused by a measurement is described by the time evolution operator $U(t, t_0) \in U(\mathbb{H})$. The time-evolved state $|\psi(t)\rangle$ originating from $|\psi_0\rangle$ is then given by

$$|\psi(t)\rangle = U(t, t_0)|\psi(t_0)\rangle\,. \tag{3.11}$$

The time evolution operator $U(t, t_0)$ is the solution of the initial value problem

$$i\frac{d}{dt}U(t, t_0) = H(t)U(t, t_0)\,, \tag{3.12a}$$

$$U(t_0, t_0) = \mathrm{I}\,, \tag{3.12b}$$

where $H(t)$ is the self-adjoint Hamilton operator (a.k.a. Hamiltonian), which is said to generate the time evolution of the quantum system.

The operator version of time evolution given in Postulate 3.79 is completely equivalent to the well-known Schrödinger equation

$$i\frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle \tag{3.13}$$

which describes the time evolution of pure states as expressed by its effect on the state vectors. This is because application of (3.12) to (3.11) results in the Schrödinger equation (3.13), and, conversely, any solution of the Schrödinger equation for arbitrary initial states $|\psi(t_0)\rangle$ yields a solution for $U(t, t_0)$. The formulation of the time evolution making use of the time evolution operator $U(t, t_0)$ given in Postulate 3.79 has the advantage over the Schrödinger equation that it can be used for mixed states (which we will not talk about) as well. The operator $H(t)$ corresponds to the observable energy of the quantum system. Hence, the expectation value $\langle H(t)\rangle_\psi$ of the Hamiltonian gives the expectation value for the energy of the system in the state $|\psi\rangle$. If $H$ is time-independent; that is, $\frac{d}{dt}H(t) = 0$, then the energy of the system is constant and is given by the eigenvalues $\{E_j \,|\, j \in I\}$ of

$\mathbb{H}$. The fact that these eigenvalues are discrete for certain Hamiltonians is at the heart of the designation "quantum". It was Planck's assumption that the energy of a black body can only be integer multiples of a fixed quantum of energy, which helped him derive the correct radiation formula. But the origins of this assumptions were not understood at the time. Only quantum mechanics subsequently provided a theoretical and mathematical theory delivering a proof for discrete energy levels.

The Hamilton operator $H(t)$ not only corresponds to the energy observable of the system, but also determines the time evolution of the system. The specific form of the operator $H(t)$ is determined by the internal and external interactions to which the quantum system is exposed. Circuits in quantum computers are built up from elementary gates that act as unitary operators $V$ on the states. In order to implement such gates one then tries to create Hamilton operators that generate a time evolution $U(t, t_0)$ implementing the desired gate; that is, one attempts to find $H(t)$ and $t$ such that $V = U(t, t_0)$.

## 3.7 Tensor Product of Vector Spaces

Motivated by Section 2.3.1, in the following we investigate the general properties of tensor product of vectors.

**Caution**: The definition of the tensor product given below is purely mathematics. You do not need to understand fully in order to learn quantum computing; however, we encourage you to go through this once for it will explain a lot of things that normal textbooks for quantum computing will not talk about.

### 3.7.1 Tensor product

Let $\mathbb{V}$ be a vector space over a scalar field $\mathbb{F}$. By Proposition 3.29, we find that $\mathbb{V}'$ is a vector space over $\mathbb{F}$ as well. This enables us to consider $(\mathbb{V}')'$, the dual space of $\mathbb{V}'$. In the example above, we have that $[(\mathbb{R}^n)']' = \mathbb{R}^n$. In general, $(\mathbb{V}')' = \mathbb{V}$ is not true, but there is an injection $\iota : \mathbb{V} \to (\mathbb{V}')'$ in the sense that

$$\iota(\boldsymbol{v})(f) \equiv f(\boldsymbol{v}) \qquad \forall\, f \in \mathbb{V}'\,. \tag{3.14}$$

The linear embedding $\iota : \mathbb{V} \to (\mathbb{V}')'$ is a natural vector space isomorphism provided, again, $\dim(\mathbb{V})$ is finite, the proof being evident as the embedding is a linear and injective map between spaces with equal finite dimension (Proposition 3.21).

The embedding (3.14) permits us to define a vector space

$$\mathbb{V}_1 \otimes \mathbb{V}_2 \otimes \cdots \otimes \mathbb{V}_n \,,$$

called the tensor product of vector spaces $\mathbb{V}_1, \mathbb{V}_2, \cdots, \mathbb{V}_n$ with a common scalar field $\mathbb{F}$. Before proceeding to the definition of the tensor product of vector spaces, we first look at the tensor product of vectors.

**Definition 3.80.** Let $\mathbb{V}_1, \cdots, \mathbb{V}_n$ be vector spaces over a common scalar field $\mathbb{F}$, and $\boldsymbol{v}_j \in \mathbb{V}_j$ be given for $1 \leqslant j \leqslant n$. The tensor product $\boldsymbol{v}_1 \otimes \cdots \otimes \boldsymbol{v}_n$ is a function from $\mathbb{V}_1' \oplus \cdots \oplus \mathbb{V}_n'$ to $\mathbb{F}$ defined by

$$(\boldsymbol{v}_1 \otimes \cdots \otimes \boldsymbol{v}_n)(f_1, ..., f_n) = \prod_{j=1}^{n} f_j(\boldsymbol{v}_j) \equiv f_1(\boldsymbol{v}_1) \cdot \cdots \cdot f_n(\boldsymbol{v}_n) \,. \tag{3.15}$$

The associativity and distributive law of the scalar field of $\mathbb{F}$ imply the following three propositions.

**Proposition 3.81.** *Let $\mathbb{U}, \mathbb{V}, \mathbb{W}$ be vector spaces over a common scalar field $\mathbb{F}$, and $\boldsymbol{u} \in \mathbb{U}$, $\boldsymbol{v} \in \mathbb{V}$ and $\boldsymbol{w} \in \mathbb{W}$. Then*

$$\boldsymbol{u} \otimes \boldsymbol{v} \otimes \boldsymbol{w} = (\boldsymbol{u} \otimes \boldsymbol{v}) \otimes \boldsymbol{w} = \boldsymbol{u} \otimes (\boldsymbol{v} \otimes \boldsymbol{w}) \,.$$

*Proof.* Let $f \in \mathbb{U}'$, $g \in \mathbb{V}'$ and $h \in \mathbb{W}'$. Then

$$\begin{aligned}(\boldsymbol{u} \otimes \boldsymbol{v} \otimes \boldsymbol{w})(f, g, h) &= f(\boldsymbol{u}) \cdot g(\boldsymbol{v}) \cdot h(\boldsymbol{w}) = \big[f(\boldsymbol{u}) \cdot g(\boldsymbol{v})\big] \cdot h(\boldsymbol{w}) = (\boldsymbol{u} \otimes \boldsymbol{v})(f, g) \cdot h(\boldsymbol{w}) \\ &= \big[(\boldsymbol{u} \otimes \boldsymbol{v}) \otimes \boldsymbol{w}\big]\big((f, g), h\big) = \big[(\boldsymbol{u} \otimes \boldsymbol{v}) \otimes \boldsymbol{w}\big](f, g, h)\end{aligned}$$

so that $\boldsymbol{u} \otimes \boldsymbol{v} \otimes \boldsymbol{w} = (\boldsymbol{u} \otimes \boldsymbol{v}) \otimes \boldsymbol{w}$. The identity $\boldsymbol{u} \otimes \boldsymbol{v} \otimes \boldsymbol{w} = \boldsymbol{u} \otimes (\boldsymbol{v} \otimes \boldsymbol{w})$ can be proved in the similar fashion, and the proof is left to the reader. □

**Proposition 3.82.** *Let $\mathbb{V}_1, \cdots, \mathbb{V}_n$ be vector spaces over a common scalar field $\mathbb{F}$. For $1 \leqslant j \leqslant n$, let $\boldsymbol{v}_\ell \in \mathbb{V}_\ell$ for $\ell \neq j$, $\boldsymbol{u}_j, \boldsymbol{w}_j \in \mathbb{V}_j$, and $c \in \mathbb{F}$. Then*

$$\begin{aligned}&\boldsymbol{v}_1 \otimes \cdots \otimes \boldsymbol{v}_{j-1} \otimes (c\boldsymbol{u}_j + \boldsymbol{w}_j) \otimes \boldsymbol{v}_{j+1} \otimes \cdots \otimes \boldsymbol{v}_n \\ &\quad = c\,(\boldsymbol{v}_1 \otimes \cdots \otimes \boldsymbol{v}_{j-1} \otimes \boldsymbol{u}_j \otimes \boldsymbol{v}_{j+1} \otimes \cdots \otimes \boldsymbol{v}_n) + (\boldsymbol{v}_1 \otimes \cdots \otimes \boldsymbol{v}_{j-1} \otimes \boldsymbol{w}_j \otimes \boldsymbol{v}_{j+1} \otimes \cdots \otimes \boldsymbol{v}_n) \,.\end{aligned}$$

*Proof.* Let $f_\ell \in \mathbb{V}'_\ell$ for $1 \leqslant \ell \leqslant n$ be given. Then

$$\begin{aligned}
&\big(\boldsymbol{v}_1 \otimes \cdots \otimes \boldsymbol{v}_{j-1} \otimes (c\boldsymbol{u}_j + \boldsymbol{w}_j) \otimes \boldsymbol{v}_{j+1} \otimes \cdots \otimes \boldsymbol{v}_n\big)(f_1, \cdots, f_n) \\
&\quad = f_1(\boldsymbol{v}_1) \cdots f_{j-1}(\boldsymbol{v}_{j-1}) \cdot f_j(c\boldsymbol{u}_j + \boldsymbol{w}_j) \cdot f_{j+1}(\boldsymbol{v}_j) \cdots f_n(\boldsymbol{v}_n) \\
&\quad = c f_1(\boldsymbol{v}_1) \cdots f_{j-1}(\boldsymbol{v}_{j-1}) \cdot f_j(\boldsymbol{u}_j) \cdot f_{j+1}(\boldsymbol{v}_j) \cdots f_n(\boldsymbol{v}_n) \\
&\qquad + f_1(\boldsymbol{v}_1) \cdots f_{j-1}(\boldsymbol{v}_{j-1}) \cdot f_j(\boldsymbol{w}_j) \cdot f_{j+1}(\boldsymbol{v}_j) \cdots f_n(\boldsymbol{v}_n) \\
&\quad = c(\boldsymbol{v}_1 \otimes \cdots \otimes \boldsymbol{v}_{j-1} \otimes \boldsymbol{u}_j \otimes \boldsymbol{v}_{j+1} \otimes \cdots \otimes \boldsymbol{v}_n)(f_1, \cdots, f_n) \\
&\qquad + (\boldsymbol{v}_1 \otimes \cdots \otimes \boldsymbol{v}_{j-1} \otimes \boldsymbol{w}_j \otimes \boldsymbol{v}_{j+1} \otimes \cdots \otimes \boldsymbol{v}_n)(f_1, \cdots, f_n)
\end{aligned}$$

which establishes the proposition. □

**Proposition 3.83.** *Let $\mathbb{V}_1, \cdots, \mathbb{V}_n$ be vector spaces over a common scalar field $\mathbb{F}$, and $\boldsymbol{v}_j \in \mathbb{V}_j$ be given for $1 \leqslant j \leqslant n$. Then $\boldsymbol{v}_1 \otimes \cdots \otimes \boldsymbol{v}_n \in \mathscr{L}(\mathbb{V}'_1, \cdots, \mathbb{V}'_n; \mathbb{F})$.*

*Proof.* Let $1 \leqslant j \leqslant n$, $f_\ell \in \mathbb{V}'_\ell$ for $\ell \neq j$, $g_j$, $h_j \in \mathbb{V}'_j$ and $c \in \mathbb{F}$ be given. Then

$$\begin{aligned}
&(\boldsymbol{v}_1 \otimes \cdots \otimes \boldsymbol{v}_n)(f_1, \cdots, f_{j-1}, cg_j + h_j, f_{j+1}, \cdots, f_n) \\
&\quad = f_1(\boldsymbol{v}_1) \cdot \cdots\cdot f_{j-1}(\boldsymbol{v}_{j-1}) \cdot \big[cg_j(\boldsymbol{v}_j) + h_j(\boldsymbol{v}_j)\big] \cdot f_{j+1}(\boldsymbol{v}_{j+1}) \cdot \cdots\cdot f_n(\boldsymbol{v}_n) \\
&\quad = c\, f_1(\boldsymbol{v}_1) \cdot \cdots\cdot f_{j-1}(\boldsymbol{v}_{j-1}) \cdot g_j(\boldsymbol{v}_j) \cdot f_{j+1}(\boldsymbol{v}_{j+1}) \cdot \cdots\cdot f_n(\boldsymbol{v}_n) \\
&\qquad + f_1(\boldsymbol{v}_1) \cdot \cdots\cdot f_{j-1}(\boldsymbol{v}_{j-1}) \cdot h_j(\boldsymbol{v}_j) \cdot f_{j+1}(\boldsymbol{v}_{j+1}) \cdot \cdots\cdot f_n(\boldsymbol{v}_n) \\
&\quad = c(\boldsymbol{v}_1 \otimes \cdots \otimes \boldsymbol{v}_n)(f_1, \cdots, f_{j-1}, g_j, f_{j+1}, \cdots, f_n) \\
&\qquad + (\boldsymbol{v}_1 \otimes \cdots \otimes \boldsymbol{v}_n)(f_1, \cdots, f_{j-1}, h_j, f_{j+1}, \cdots, f_n)
\end{aligned}$$

which shows that $\boldsymbol{v}_1 \otimes \cdots \otimes \boldsymbol{v}_n$ satisfies the multi-linearity. □

The fact that $\mathscr{L}(\mathbb{V}'_1, \cdots, \mathbb{V}'_n; \mathbb{F})$ is a vector space over $\mathbb{F}$ motivates the definition of the tensor product of vector spaces.

**Definition 3.84.** Let $\mathbb{V}_1$, $\cdots$, $\mathbb{V}_n$ be vector spaces over a common scalar field $\mathbb{F}$. The tensor product space $\mathbb{V}_1 \otimes \cdots \otimes \mathbb{V}_n$ is the subspace of $\mathscr{L}(\mathbb{V}'_1, \cdots, \mathbb{V}'_n; \mathbb{F})$ spanned by all finite linear combinations of tensor products $\boldsymbol{v}_1 \otimes \cdots \otimes \boldsymbol{v}_n$, where $\boldsymbol{v}_j \in \mathbb{V}_j$ for $1 \leqslant j \leqslant n$ and $\boldsymbol{v}_1 \otimes \cdots \otimes \boldsymbol{v}_n \in \mathscr{L}(\mathbb{V}'_1, \cdots, \mathbb{V}'_n; \mathbb{F})$ is given by (3.15).

The following proposition is a direct consequence of Proposition 3.81.

**Proposition 3.85.** *Let $\mathbb{U}, \mathbb{V}, \mathbb{W}$ be vector spaces over a common scalar field $\mathbb{F}$. Then*

$$\mathbb{U} \otimes \mathbb{V} \otimes \mathbb{W} = (\mathbb{U} \otimes \mathbb{V}) \otimes \mathbb{W} = \mathbb{U} \otimes (\mathbb{V} \otimes \mathbb{W}).$$

**Proposition 3.86.** *Let $\mathbb{V}_1, \cdots, \mathbb{V}_n$ be finite-dimensional vector spaces over a common scalar field $\mathbb{F}$. Then $\mathbb{V}_1 \otimes \cdots \mathbb{V}_n$ is finite-dimensional and*

$$\dim\big(\mathbb{V}_1 \otimes \cdots \otimes \mathbb{V}_n\big) = \prod_{j=1}^{n} \dim(\mathbb{V}_j) = \dim(\mathbb{V}_1) \cdot \cdots \cdot \dim(\mathbb{V}_n)\,.$$

*Proof.* By Proposition 3.85, it suffices to show the case $n = 2$.

Let $\{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_n\}$ and $\{\widetilde{\mathbf{e}}_1, \widetilde{\mathbf{e}}_2, \cdots, \widetilde{\mathbf{e}}_m\}$ be basis of $\mathbb{V}_1$ and $\mathbb{V}_2$, respectively. For $\boldsymbol{x} \in \mathbb{V}_1$ and $\boldsymbol{y} \in \mathbb{V}_2$, write $\boldsymbol{x} = \sum_{k=1}^{n} x_k \mathbf{e}_k$ and $\boldsymbol{y} = \sum_{\ell=1}^{m} y_\ell \widetilde{\mathbf{e}}_\ell$. Then Proposition 3.82 implies that

$$\boldsymbol{x} \otimes \boldsymbol{y} = \Big(\sum_{k=1}^{n} x_k \mathbf{e}_k\Big) \otimes \Big(\sum_{\ell=1}^{m} y_\ell \widetilde{\mathbf{e}}_\ell\Big) = \sum_{k=1}^{n}\sum_{\ell=1}^{m} x_k y_\ell (\mathbf{e}_k \otimes \widetilde{\mathbf{e}}_\ell)\,.$$

Since vectors in $\mathbb{V}_1 \otimes \mathbb{V}_2$ can be expressed as a linear combination of vectors of the form $\boldsymbol{x} \otimes \boldsymbol{y}$, we find that every vectors in $\mathbb{V}_1 \otimes \mathbb{V}_2$ can be expressed as a linear combination of vectors from the set $\mathcal{B} \equiv \big\{\mathbf{e}_k \otimes \widetilde{\mathbf{e}}_\ell \,\big|\, 1 \leqslant k \leqslant n, 1 \leqslant \ell \leqslant m\big\}$. Since $\#\mathcal{B} = nm$, it suffices to show that $\mathcal{B}$ is a linearly independent set.

Let $\big\{c_{k\ell}\big\}_{1\leqslant k\leqslant n, 1\leqslant \ell\leqslant m}$ be a collection of scalars in $\mathbb{F}$ such that

$$\sum_{k=1}^{n}\sum_{\ell=1}^{m} c_{k\ell}\mathbf{e}_k \otimes \widetilde{\mathbf{e}}_\ell = 0 \quad \text{(the zero vector in } \mathbb{V}_1 \otimes \mathbb{V}_2\text{)}.$$

Let $f_i \in \mathbb{V}_1'$ and $g_j \in \mathbb{V}_2'$ satisfy

$$f_i(\mathbf{e}_k) = \delta_{ik} \qquad \text{and} \qquad g_j(\widetilde{\mathbf{e}}_\ell) = \delta_{j\ell}\,,$$

where $\delta_{\cdot\cdot}$ are the Kronecker delta. Then for each $1 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant m$,

$$0 = \Big(\sum_{k=1}^{n}\sum_{\ell=1}^{m} c_{k\ell}\mathbf{e}_k \otimes \widetilde{\mathbf{e}}_\ell\Big)(f_i, g_j) = \sum_{k=1}^{n}\sum_{\ell=1}^{m} c_{k\ell}\delta_{ik}\delta_{j\ell} = c_{ij}\,.$$

This implies that $\mathcal{B}$ is a linearly independent set; thus $\dim(\mathbb{V}_1 \otimes \mathbb{V}_2) = \#\mathcal{B} = nm$. ▫

Next we consider the (matrix/coordinate) representation of tensor product of vectors. This amounts to choose an ordered basis in the tensor product space. We start with the following

**Example 3.87.** Let $\mathbb{U}$ and $\mathbb{V}$ be vector spaces over a common scalar field $\mathbb{F}$, and $\mathcal{B}_\mathbb{U} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ and $\mathcal{B}_\mathbb{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2\}$ be basis of $\mathbb{U}$ and $\mathbb{V}$, respectively. For $\boldsymbol{x} \in \mathbb{U}$ and $\boldsymbol{y} \in \mathbb{V}$, there exist unique $x_1, x_2, x_3, y_1, y_2 \in \mathbb{F}$ such that

$$\boldsymbol{x} = x_1\boldsymbol{u}_1 + x_2\boldsymbol{u}_2 + x_3\boldsymbol{u}_3 \quad \text{and} \quad \boldsymbol{y} = y_1\boldsymbol{v}_1 + y_2\boldsymbol{v}_2\,,$$

By the property of the tensor product of vectors,

$$\boldsymbol{x}\otimes\boldsymbol{y}=\sum_{i=1}^{3}\sum_{j=1}^{2}x_iy_j(\boldsymbol{u}_i\otimes\boldsymbol{v}_j)$$

so that the coordinate (the collection of coefficients) of $\boldsymbol{x}\otimes\boldsymbol{y}$ relative to the ordered basis $\mathcal{B}=\left\{\boldsymbol{u}_1\otimes\boldsymbol{v}_1,\boldsymbol{u}_1\otimes\boldsymbol{v}_2,\boldsymbol{u}_2\otimes\boldsymbol{v}_1,\boldsymbol{u}_2\otimes\boldsymbol{v}_2,\boldsymbol{u}_3\otimes\boldsymbol{v}_1,\boldsymbol{u}_3\otimes\boldsymbol{v}_2\right\}$ of $\mathbb{U}\otimes\mathbb{V}$ is given by

$$(x_1y_1,x_1y_2,x_2y_1,x_2y_2,x_3y_1,x_3y_2)\,.$$

Writing the coordinate in terms of a column vector, we have

$$[\boldsymbol{x}\otimes\boldsymbol{y}]_{\mathcal{B}}=\begin{bmatrix}x_1y_1\\x_1y_2\\x_2y_1\\x_2y_2\\x_3y_1\\x_3y_2\end{bmatrix}=\begin{bmatrix}x_1\begin{bmatrix}y_1\\y_2\end{bmatrix}\\x_2\begin{bmatrix}y_1\\y_2\end{bmatrix}\\x_3\begin{bmatrix}y_1\\y_2\end{bmatrix}\end{bmatrix}\equiv\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}\otimes\begin{bmatrix}y_1\\y_2\end{bmatrix}=[\boldsymbol{x}]_{\mathcal{B}_{\mathbb{U}}}\otimes[\boldsymbol{y}]_{\mathcal{B}_{\mathbb{V}}}\,.$$

The ordered basis $\mathcal{B}$ is called the **induced basis** of the ordered basis $\mathcal{B}_{\mathbb{U}}$ and $\mathcal{B}_{\mathbb{V}}$.

**Remark 3.88.** For given basis of vector spaces, there are two induced basis, one for direct sum of spaces and one for tensor product of spaces. We will abuse the use of the word "induced" but keep in mind that it refers to one particular type.

The example above motivates the formal/computational definition of the tensor product of vectors in $\mathbb{C}^m$ and $\mathbb{C}^n$ as follows. Let $\boldsymbol{x}=[x_1,x_2,\cdots,x_m]^{\mathrm{T}}\in\mathbb{C}^m$ and $\boldsymbol{y}=[y_1,y_2,\cdots,y_n]^{\mathrm{T}}\in\mathbb{C}^n$. The "tensor product" of $\boldsymbol{x}$ and $\boldsymbol{y}$, denoted by $[\boldsymbol{x}\otimes\boldsymbol{y}]$, is a vector in $\mathbb{C}^{mn}$ given by

$$[\boldsymbol{x}\otimes\boldsymbol{y}]=\begin{bmatrix}x_1y_1\\\vdots\\x_1y_n\\\vdots\\x_my_1\\\vdots\\x_my_n\end{bmatrix}=\begin{bmatrix}x_1\begin{bmatrix}y_1\\\vdots\\y_n\end{bmatrix}\\\vdots\\x_m\begin{bmatrix}y_1\\\vdots\\y_n\end{bmatrix}\end{bmatrix}\equiv\begin{bmatrix}x_1\\\vdots\\x_m\end{bmatrix}\otimes\begin{bmatrix}y_1\\\vdots\\y_n\end{bmatrix}=[\boldsymbol{x}]\otimes[\boldsymbol{y}].$$

In fact, $[\boldsymbol{x}\otimes\boldsymbol{y}]\in\mathbb{C}^{mn}$ is indeed the coordinate of $\boldsymbol{x}\otimes\boldsymbol{y}$ relative to the induced ordered basis $\left\{\mathbf{e}_1\otimes\widetilde{\mathbf{e}}_1,\mathbf{e}_1\otimes\widetilde{\mathbf{e}}_2,\cdots,\mathbf{e}_1\otimes\widetilde{\mathbf{e}}_n,\mathbf{e}_2\otimes\widetilde{\mathbf{e}}_1,\mathbf{e}_2\otimes\widetilde{\mathbf{e}}_2,\cdots,\mathbf{e}_2\otimes\widetilde{\mathbf{e}}_n,\cdots,\mathbf{e}_m\otimes\widetilde{\mathbf{e}}_1,\mathbf{e}_m\otimes\widetilde{\mathbf{e}}_2,\cdots,\mathbf{e}_m\otimes\widetilde{\mathbf{e}}_n\right\}$ of $\mathbb{C}^m\otimes\mathbb{C}^n$.

Suppose that $X, Y, Z, W$ are vector spaces over a scalar field $\mathbb{F}$. Then $\mathscr{L}(X;Z)$ and $\mathscr{L}(Y;W)$ are also vector spaces so that $\mathscr{L}(X;Z) \otimes \mathscr{L}(Y;W)$ is a well-defined concept (in the sense of Definition 3.84). In the following, we talk about an alternative definition of $A \otimes B$ if $A \otimes B$ is if $A \in \mathscr{L}(X;Z)$ and $B \in \mathscr{L}(Y;W)$.

**Definition 3.89.** Let $X, Y, Z, W$ be vector spaces over a common scalar field $\mathbb{F}$, and $A \in \mathscr{L}(X;Z)$, $B \in \mathscr{L}(Y;W)$. The tensor product of $A$ and $B$, denoted by $A \otimes B$, is an element in $\mathscr{L}(X \otimes Y, Z \otimes W)$ satisfying that

$$(A \otimes B)(\boldsymbol{x} \otimes \boldsymbol{y}) = (A\boldsymbol{x}) \otimes (B\boldsymbol{y}) \qquad \forall\, \boldsymbol{x} \in X \text{ and } \boldsymbol{y} \in Y\,.$$

**Remark 3.90.** To avoid confusion, instead of treating $A \otimes B$ as the tensor product of $A$ and $B$ one can also treat $A \otimes B$ as a "**new operation**" between linear maps $A$ and $B$ (but with the same notation).

**Proposition 3.91.** *Let $A, B, C$ be linear maps on vector spaces $X, Y, Z$ (over a common scalar field $\mathbb{F}$). Then*

$$(A \otimes B) \otimes C = A \otimes (B \otimes C)\,.$$

- **Matrix representation of tensor product of linear maps**

Suppose that $X, Y, Z, W$ are finite dimensional vector spaces over field $\mathbb{F}$, and $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$, $\{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_\ell\}$, $\{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_m\}$ and $\{\boldsymbol{w}_1, \cdots, \boldsymbol{w}_k\}$ are basis of $X, Y, Z, W$, respectively. Let $A \in \mathscr{L}(X;Z)$, $B \in \mathscr{L}(Y;W)$, and the matrix representations of $A$ and $B$ be $[A] = [a_{ij}]_{1\leqslant i\leqslant m, 1\leqslant j\leqslant n}$ and $[B] = [b_{ij}]_{1\leqslant i\leqslant k, 1\leqslant j\leqslant \ell}$, respectively, so that

$$A(c_1\boldsymbol{x}_1 + \cdots + c_n\boldsymbol{x}_n) = \sum_{i=1}^{m}\Big(\sum_{r=1}^{n} a_{ir}c_r\Big)\boldsymbol{z}_i\,, \quad B(d_1\boldsymbol{y}_1 + \cdots + d_\ell\boldsymbol{y}_\ell) = \sum_{j=1}^{k}\Big(\sum_{s=1}^{\ell} b_{js}d_s\Big)\boldsymbol{w}_j\,.$$

Then

$$\begin{aligned}
&(A \otimes B)\big((c_1\boldsymbol{x}_1 + \cdots + c_n\boldsymbol{x}_n) \otimes (d_1\boldsymbol{y}_1 + \cdots + d_\ell\boldsymbol{y}_\ell)\big)\\
&\quad\equiv \big[A(c_1\boldsymbol{x}_1 + \cdots + c_n\boldsymbol{x}_n)\big] \otimes \big[B(d_1\boldsymbol{y}_1 + \cdots + d_\ell\boldsymbol{y}_\ell)\big]\\
&\quad= \Big[\sum_{i=1}^{m}\Big(\sum_{r=1}^{n} a_{ir}c_r\Big)\boldsymbol{z}_i\Big] \otimes \Big[\sum_{j=1}^{k}\Big(\sum_{s=1}^{\ell} b_{js}d_s\Big)\boldsymbol{w}_j\Big]\\
&\quad= \sum_{i=1}^{m}\sum_{j=1}^{k}\Big(\sum_{r=1}^{n}\sum_{s=1}^{\ell} a_{ir}b_{js}c_rd_s\Big)\boldsymbol{z}_i \otimes \boldsymbol{w}_j\,.
\end{aligned}$$

Since the induced ordered basis of $X \otimes Y$ and $Z \otimes W$ are given respectively by

$$\mathcal{B}_{X\otimes Y} = \left\{ \boldsymbol{x}_1 \otimes \boldsymbol{y}_1, \cdots, \boldsymbol{x}_1 \otimes \boldsymbol{y}_\ell, \boldsymbol{x}_2 \otimes \boldsymbol{y}_1, \cdots, \boldsymbol{x}_2 \otimes \boldsymbol{y}_\ell, \cdots, \boldsymbol{x}_n \otimes \boldsymbol{y}_1, \cdots, \boldsymbol{x}_n \otimes \boldsymbol{y}_\ell \right\},$$
$$\mathcal{B}_{Z\otimes W} = \left\{ \boldsymbol{z}_1 \otimes \boldsymbol{w}_1, \cdots, \boldsymbol{z}_1 \otimes \boldsymbol{w}_k, \boldsymbol{z}_2 \otimes \boldsymbol{w}_1, \cdots, \boldsymbol{z}_2 \otimes \boldsymbol{w}_k, \cdots, \boldsymbol{z}_m \otimes \boldsymbol{w}_1, \cdots, \boldsymbol{z}_m \otimes \boldsymbol{w}_k \right\},$$

the matrix representation of $A \otimes B$ satisfies

$$[A \otimes B] \begin{bmatrix} c_1 d_1 \\ \vdots \\ c_1 d_\ell \\ c_2 d_1 \\ \vdots \\ c_2 d_\ell \\ \vdots \\ c_n d_1 \\ \vdots \\ c_n d_\ell \end{bmatrix}_{n\ell \times 1} = \begin{bmatrix} \sum_{r=1}^n \sum_{s=1}^\ell a_{1r} b_{1s} c_r d_s \\ \vdots \\ \sum_{r=1}^n \sum_{s=1}^\ell a_{1r} b_{ks} c_r d_s \\ \sum_{r=1}^n \sum_{s=1}^\ell a_{2r} b_{1s} c_r d_s \\ \vdots \\ \sum_{r=1}^n \sum_{s=1}^\ell a_{2r} b_{ks} c_r d_s \\ \vdots \\ \sum_{r=1}^n \sum_{s=1}^\ell a_{mr} b_{1s} c_r d_s \\ \vdots \\ \sum_{r=1}^n \sum_{s=1}^\ell a_{mr} b_{ks} c_r d_s \end{bmatrix}_{mk \times 1}$$

for all $c_1, \cdots, c_n$ and $d_1, \cdots, d_\ell$ in $\mathbb{F}$.

Let $c_1 = d_j = 1$, where $1 \leqslant j \leqslant \ell$, and $c_r = d_s = 0$ for other $r, s$, we find that the $j$-th column of $[A \otimes B]$ is given by

$$[A \otimes B](:, j) = \begin{bmatrix} a_{11} b_{1j} \\ \vdots \\ a_{11} b_{kj} \\ a_{21} b_{1j} \\ \vdots \\ a_{21} b_{kj} \\ \vdots \\ a_{m1} b_{1j} \\ \vdots \\ a_{m1} b_{kj} \end{bmatrix} = \begin{bmatrix} a_{11} \begin{bmatrix} b_{1j} \\ \vdots \\ b_{kj} \end{bmatrix} \\ a_{21} \begin{bmatrix} b_{1j} \\ \vdots \\ b_{kj} \end{bmatrix} \\ \vdots \\ a_{m1} \begin{bmatrix} b_{1j} \\ \vdots \\ b_{kj} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} a_{11} \\ \vdots \\ a_{m1} \end{bmatrix} \otimes \begin{bmatrix} b_{1j} \\ \vdots \\ b_{kj} \end{bmatrix}$$

so that the first $\ell$ columns of $[A \otimes B]$ are given by

$$[A \otimes B](:, 1:\ell) = \begin{bmatrix} a_{11}[B] \\ a_{21}[B] \\ \vdots \\ a_{m1}[B] \end{bmatrix} = \begin{bmatrix} a_{11} \\ \vdots \\ a_{m1} \end{bmatrix} \otimes [B].$$

Let $c_2 = d_j = 1$, where $1 \leqslant j \leqslant \ell$, and $c_r = d_s = 0$ for other $r, s$, we find that the $(\ell + j)$-th column of $[A \otimes B]$ is given by

$$[A \otimes B](:, \ell + j) = \begin{bmatrix} a_{12}b_{1j} \\ \vdots \\ a_{12}b_{kj} \\ a_{22}b_{1j} \\ \vdots \\ a_{22}b_{kj} \\ \vdots \\ a_{m2}b_{1j} \\ \vdots \\ a_{m2}b_{kj} \end{bmatrix} = \begin{bmatrix} a_{12} \begin{bmatrix} b_{1j} \\ \vdots \\ b_{kj} \end{bmatrix} \\ a_{22} \begin{bmatrix} b_{1j} \\ \vdots \\ b_{kj} \end{bmatrix} \\ \vdots \\ a_{m2} \begin{bmatrix} b_{1j} \\ \vdots \\ b_{kj} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} a_{12} \\ \vdots \\ a_{m2} \end{bmatrix} \otimes \begin{bmatrix} b_{1j} \\ \vdots \\ b_{kj} \end{bmatrix}$$

so that the $(\ell + 1)$-th to $2\ell$-th columns of $[A \otimes B]$ are given by

$$[A \otimes B](:, \ell 1 : 2\ell) = \begin{bmatrix} a_{12}[B] \\ a_{22}[B] \\ \vdots \\ a_{m2}[B] \end{bmatrix} = \begin{bmatrix} a_{12} \\ \vdots \\ a_{m2} \end{bmatrix} \otimes [B] \,.$$

In general, we can find that the $(p-1)\ell + j$ column of $[A \otimes B]$ by letting $c_p = d_j = 1$ and $c_r = d_s = 0$ for other $r, s$ and obtain that the matrix representation of $A \otimes B$ is then given by

$$[A \otimes B] = \begin{bmatrix} a_{11}[B] & a_{12}[B] & \cdots & a_{1n}[B] \\ a_{21}[B] & a_{22}[B] & \cdots & a_{2n}[B] \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}[B] & a_{m2}[B] & \cdots & a_{mn}[B] \end{bmatrix} \equiv [A] \otimes [B] \,.$$

**Definition 3.92.** Let $A \in \mathcal{M}(m, n)$ and $B \in \mathcal{M}(k, \ell)$ The tensor product of $A$ and $B$, denoted by $A \otimes B$, is an $(mk) \times (n\ell)$ matrix given by

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix} .$$

**Remark 3.93.** In matlab®, the tensor product $A \otimes B$ of two matrices $A$ and $B$ is given by

$$A \otimes B = \mathbf{kron}(A, B) \,.$$

• **Tensor product of Hilbert spaces**

We note that the tensor product spaces defined above does not have an inner product structure. Now let us talk about the Hilbertian tensor product of Hilbert spaces. Consider a finite number of (complex) Hilbert spaces $\mathbb{H}_1, \cdots, \mathbb{H}_n$ with respective Hermitian scalar products $\langle \cdot, \cdot \rangle_1$, $\cdots$, $\langle \cdot, \cdot \rangle_n$. Relying upon the above definition, we can first define their algebraic tensor product

$$\mathbb{H}_1 \otimes \cdots \otimes \mathbb{H}_n \, .$$

This is not a Hilbert space yet. However it is possible (not so easy) to prove that $\mathbb{H}_1 \otimes \cdots \otimes \mathbb{H}_n$ admits an Hermitian scalar product induced by the ones of each $\mathbb{H}_j$. This scalar product $\langle \cdot, \cdot \rangle$ is the unique right-linear (property (3) and (4) in Definition 3.30) and left-antilinear (property (3) and (5) in Definition 3.30) extension of

$$\langle \boldsymbol{u}_1 \otimes \cdots \otimes \boldsymbol{u}_n, \boldsymbol{v}_1 \otimes \cdots \otimes \boldsymbol{v}_n \rangle \equiv \prod_{j=1}^{n} \langle \boldsymbol{u}_j, \boldsymbol{v}_j \rangle_j \quad \text{if} \quad \boldsymbol{u}_j, \boldsymbol{v}_j \in \mathbb{H}_j \text{ for all } 1 \leqslant j \leqslant n \, . \tag{3.16}$$

The (anti)linear extension is necessary because $\boldsymbol{\psi}_1 \otimes \cdots \otimes \boldsymbol{\psi}_n$ is not the generic element of $\mathbb{H}_1 \otimes \cdots \otimes \mathbb{H}_n$, the generic element is a finite linear combination of these elements!

It turns out that the unique (anti)liner extension of (3.16) defines an Hermitian scalar product on $\mathbb{H}_1 \otimes \cdots \otimes \mathbb{H}_n$, in particular the extension is positively defined (property (1) and (2) in Definition 3.30).

**Definition 3.94.** The Hilbertian tensor product of (complex) Hilbert spaces $(\mathbb{H}_1, \langle \cdot, \cdot \rangle_1)$, $\cdots$, $(\mathbb{H}_n, \langle \cdot, \cdot \rangle_n)$ is the (complex) Hilbert space $\mathbb{H}_1 \otimes \cdots \otimes \mathbb{H}_n$ given as the completion of the algebraic tensor product $\mathbb{H}_1 \otimes \cdots \otimes \mathbb{H}_n$ with respect to the Hermitian scalar product $\langle \cdot, \cdot \rangle$ which uniquely (anti)linearly extends (3.16).

The completion $\overline{\mathbb{V}}$ of a vector space $\mathbb{V}$ equipped with an Hermitian scalar product $\langle \cdot, \cdot \rangle$ is the complete (Hilbert) space of the equivalence classes of the Cauchy sequences in $\mathbb{V}$ equipped with the unique continuous extension of $\langle \cdot, \cdot \rangle$. So it is uniquely defined (up to Hilbert space isomorphisms) and $\mathbb{V}$ is dense in $\overline{\mathbb{V}}$. Nevertheless, the Hilbert spaces $\mathbb{H}_j$ in quantum computing are $\mathbb{C}^2$ for all $1 \leqslant j \leqslant n$, so the tensor product spaces $\mathbb{H}_1 \otimes \cdots \otimes \mathbb{H}_n$ along with the norm induced by the inner product defined by (3.16) is again a Hilbert space.

**Theorem 3.95.** *For each $n \in \mathbb{N}$,*

$$\bigotimes_{\ell=1}^{n} \big(|0\rangle + e^{i\phi_\ell}|1\rangle\big) \equiv (|0\rangle + e^{i\phi_1}|1\rangle) \otimes (|0\rangle + e^{i\phi_2}|1\rangle) \otimes \cdots \otimes (|0\rangle + e^{i\phi_n}|1\rangle)$$
$$= \sum_{j=0}^{2^n-1} e^{i(j_1\phi_1 + j_2\phi_2 + \cdots + j_n\phi_n)}|j\rangle\,, \tag{3.17}$$

*where $|j\rangle = |j_1 j_2 \cdots j_n\rangle$ for $j \in \{0,1\}^n$.*

*Proof.* Since

$$|0\rangle + e^{i\phi_\ell}|1\rangle = \sum_{j_\ell=0}^{1} e^{ij_\ell\phi_\ell}|j_\ell\rangle\,,$$

we find that

$$\bigotimes_{\ell=1}^{n} \big(|0\rangle + e^{i\phi_\ell}|1\rangle\big) = \bigotimes_{\ell=1}^{n} \sum_{j_\ell=0}^{1} e^{ij_\ell\phi_\ell}|j_\ell\rangle = \Big(\sum_{j_1=0}^{1} e^{ij_1\phi_1}|j_1\rangle\Big) \otimes \cdots \otimes \Big(\sum_{j_n=0}^{1} e^{ij_n\phi_n}|j_n\rangle\Big)$$
$$= \sum_{j_1=0}^{1}\sum_{j_2=0}^{1}\cdots\sum_{j_n=0}^{1} e^{i(j_1\phi_1 + j_2\phi_2 + \cdots j_n\phi_n)}|j_1\rangle \otimes |j_2\rangle \otimes \cdots \otimes |j_n\rangle$$
$$= \sum_{j_1=0}^{1}\sum_{j_2=0}^{1}\cdots\sum_{j_n=0}^{1} e^{i(j_1\phi_1 + j_2\phi_2 + \cdots j_n\phi_n)}|j_1 j_2 \cdots j_n\rangle$$
$$= \sum_{(j_1,j_2,\cdots,j_n)\in\{0,1\}^n} e^{i(j_1\phi_1 + j_2\phi_2 + \cdots j_n\phi_n)}|j_1 j_2 \cdots j_n\rangle$$

which concludes (3.17). □

**Corollary 3.96.** *For each $n \in \mathbb{N}$ and $j = (j_1 j_2 \cdots j_n)_2$,*

$$\mathrm{H}^{\otimes n}|j\rangle \equiv \mathrm{H}^{\otimes n}|j_1 j_2 \cdots j_n\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} (-1)^{j\bullet k}|k\rangle\,, \tag{3.18}$$

*where we recall that with $k = (k_1 k_2 \cdots k_n)_2$, $j \bullet k \equiv j_1 k_1 + \cdots j_n k_n$.*

*Proof.* Note that for $j_\ell \in \{0,1\}$,

$$\mathrm{H}|j_\ell\rangle = \frac{1}{\sqrt{2}} \sum_{k_\ell=0}^{1} (-1)^{j_\ell k_\ell}|k_\ell\rangle\,.$$

Therefore, using (3.17) for the case $\phi_1 = k_1, \cdots, \phi_n = k_n$ we find that

$$\begin{aligned}
&\mathrm{H}^{\otimes n}|j_1 j_2 \cdots j_n\rangle \\
&\quad = (\mathrm{H}|j_1\rangle) \otimes \cdots \otimes (\mathrm{H}|j_n\rangle) = \Big(\frac{1}{\sqrt{2}} \sum_{k_1=0}^{1} (-1)^{j_1 k_1} |k_1\rangle\Big) \otimes \cdots \otimes \Big(\frac{1}{\sqrt{2}} \sum_{k_n=0}^{1} (-1)^{j_n k_n} |k_n\rangle\Big) \\
&\quad = \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^{1} \sum_{k_2=0}^{1} \cdots \sum_{k_n=0}^{1} (-1)^{j_1 k_1 + j_2 k_2 + \cdots + j_n k_n} |k_1\rangle \otimes |k_2\rangle \otimes \cdots \otimes |k_n\rangle \\
&\quad = \frac{1}{\sqrt{2^n}} \sum_{k = k_1 k_2 \cdots k_n \in \{0,1\}^n} (-1)^{j_1 k_1 + j_2 k_2 + \cdots + j_n k_n} |k_1 k_2 \cdots k_n\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n - 1} (-1)^{j \bullet k} |k\rangle
\end{aligned}$$

which concludes (3.18). □

**Remark 3.97.** Note that the qubit

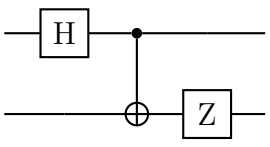
$$\frac{1}{\sqrt{2}}\big(|0\rangle + e^{i\phi}|1\rangle\big) = \mathrm{R}_\phi|+\rangle = \mathrm{R}_\phi \mathrm{H}|0\rangle\,,$$

(3.17) implies that

$$\bigotimes_{\ell=1}^{n} (\mathrm{R}_{\phi_\ell}\mathrm{H})|0\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} e^{i(j_1\phi_1 + j_2\phi_2 + \cdots + j_n\phi_n)} |j\rangle\,.$$

### 3.7.2 Correspondence between tensor product and quantum circuits

The tensor product of quantum gates represents a unitary transformation when these quantum gates are applied in parallel (at the same time), while the ordinary product of quantum gates represents a unitary transformation when these quantum gates are applied sequentially. Using the matrix representation of quantum gates, there is a way to understand the overall effect of all quantum gates applied in a system. For example, the overall unitary transformation given by the quantum circuit




$\big($which produces the entangled quantum state $\frac{1}{\sqrt{2}}\big(|00\rangle - |11\rangle\big)$ when it applies on $|00\rangle$, as

explained in Section 2.4) is $(\mathrm{I}_2 \otimes \mathrm{Z})\mathbf{CNOT}(\mathrm{H} \otimes \mathrm{I}_2)$, so using the matrix representation

$$[\mathrm{H} \otimes \mathrm{I}_2] = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \mathrm{I} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathrm{I}_2 & \mathrm{I}_2 \\ \mathrm{I}_2 & -\mathrm{I}_2 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$

and

$$[\mathrm{I}_2 \otimes \mathrm{Z}] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \mathrm{Z} = \begin{bmatrix} \mathrm{Z} & 0 \\ 0 & \mathrm{Z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix},$$

as well as the matrix representation of **CNOT** we find that the matrix representation of the overall unitary transformation given by the quantum circuit can be computed by

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & -\frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \end{bmatrix}.$$

This matrix representation of the overall unitary transform immediately tells us how to produce the EPR pair $\frac{1}{\sqrt{2}}\big(|00\rangle + |11\rangle\big)$: simply apply this circuit to the state $|10\rangle$ since $[|10\rangle] = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^{\mathrm{T}}$ and

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & -\frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

which corresponds to the EPR pair.

### 3.7.3 More examples

In the following examples, for an $n$-qubit system we always use the ordered basis $\mathcal{B}_n \equiv \{|0\rangle, |1\rangle, |2\rangle, \cdots, |2^n-1\rangle\}$, where, by writting $k$ in terms of binary number $(k_1 k_2 \cdots k_n)_2$; that is,

$$k = 2^{n-1}k_1 + 2^{n-2}k_1 + \cdots + 2^1 k_{n-1} + 2^0 k_n \,,$$

the $k$-th basis vector in $\mathcal{B}_n$ is $|k-1\rangle$.

**Example 3.98** (Matrix representation of swap operation)**.** In an $n$-qubit system, we use $\mathbf{SWAP}_{i,j}$ (here we assume $i < j$ since $\mathbf{SWAP}_{i,j} \equiv \mathbf{SWAP}_{j,i}$) to denote the swap operator that swaps the position of the $i$-th and the $j$-th qubit; that is

$$\begin{aligned}&\mathbf{SWAP}_{i,j}|x_1\rangle\otimes\cdots\otimes|x_n\rangle\\ &\quad= |x_1\rangle\otimes\cdots\otimes|x_{i-1}\rangle\otimes|x_j\rangle\otimes|x_{i+1}\rangle\otimes\cdots\otimes|x_{j-1}\rangle\otimes|x_i\rangle\otimes|x_{j+1}\rangle\otimes\cdots\otimes|x_n\rangle\,.\end{aligned}$$

We note that $\mathbf{SWAP}_{i,j}$ is perfectly defined operator as long as $i \neq j$ and $i, j \leqslant n$. On the other hand, the matrix representation for $\mathbf{SWAP}_{i,j}$ is a $2^n \times 2^n$ matrix which essentially depends on the number of qubits in a qubit system that **SWAP** gate acts on. Therefore, to denote the matrix representation of $\mathbf{SWAP}_{i,j}$ one should use something like $[\mathbf{SWAP}_{i,j}]_n$ to indicate the number $n$ of qubits in the system. In the following, for simplicity instead of $\left[\mathbf{SWAP}_{i,j}\right]_n$ we still use $\mathbf{SWAP}_{i,j}$ to denote the matrix representation of $\mathbf{SWAP}_{i,j}$ without explicitly indicating (but knowing) the number $n$ of qubits in the system under consideration.

We first consider the swap operator on a 2-qubit system, denoted by **SWAP** and defined by

$$\mathbf{SWAP}|x\rangle\otimes|y\rangle = |y\rangle\otimes|x\rangle\,.$$

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

$$\begin{aligned}|x\rangle\otimes|y\rangle &= (\alpha_0|0\rangle + \alpha_1|0\rangle)\otimes(\beta_0|0\rangle + \beta_1|1\rangle)\\ &= \alpha_0\beta_0|0\rangle\otimes|0\rangle + \alpha_0\beta_1|0\rangle\otimes|1\rangle + \alpha_1\beta_0|1\rangle\otimes|0\rangle + \alpha_1\beta_1|1\rangle\otimes|1\rangle\\ &= \alpha_0\beta_0|0\rangle + \alpha_0\beta_1|1\rangle + \alpha_1\beta_0|2\rangle + \alpha_1\beta_1|3\rangle\,,\end{aligned}$$

and

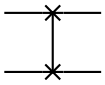
$$\begin{aligned}|y\rangle\otimes|x\rangle &= (\beta_0|0\rangle + \beta_1|1\rangle)\otimes(\alpha_0|0\rangle + \alpha_1|0\rangle)\\ &= \alpha_0\beta_0|0\rangle\otimes|0\rangle + \alpha_0\beta_1|1\rangle\otimes|0\rangle + \alpha_1\beta_0|0\rangle\otimes|1\rangle + \alpha_1\beta_1|1\rangle\otimes|1\rangle\\ &= \alpha_0\beta_0|0\rangle + \alpha_1\beta_0|1\rangle + \alpha_0\beta_1|2\rangle + \alpha_1\beta_1|3\rangle\end{aligned}$$

so that

$$\mathbf{SWAP}\begin{bmatrix}\alpha_0\beta_0\\ \alpha_0\beta_1\\ \alpha_1\beta_0\\ \alpha_1\beta_1\end{bmatrix} = \begin{bmatrix}\alpha_0\beta_0\\ \alpha_1\beta_0\\ \alpha_0\beta_1\\ \alpha_1\beta_1\end{bmatrix}$$

for all $(\alpha_0, \alpha_1), (\beta_0, \beta_1)$ on the Bloch sphere. Therefore, the matrix representation of **SWAP** (relative to the ordered basis $\mathcal{B}_2$) is a $4 \times 4$ matrix given by

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

The quantum circuit symbol for **SWAP** is 

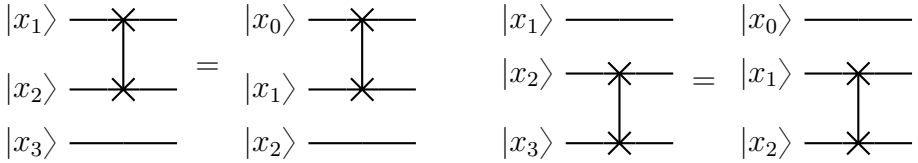
Similarly, on a 3-qubit system, there are three swap operators:

$$\begin{aligned}\mathbf{SWAP}_{1,2}|x\rangle \otimes |y\rangle \otimes |z\rangle &= |y\rangle \otimes |x\rangle \otimes |z\rangle,\\ \mathbf{SWAP}_{2,3}|x\rangle \otimes |y\rangle \otimes |z\rangle &= |x\rangle \otimes |z\rangle \otimes |y\rangle,\\ \mathbf{SWAP}_{1,3}|x\rangle \otimes |y\rangle \otimes |z\rangle &= |z\rangle \otimes |y\rangle \otimes |x\rangle.\end{aligned}$$

Note that

$$\mathbf{SWAP}_{1,2} = \mathbf{SWAP} \otimes \mathrm{I}_2 \qquad \text{and} \qquad \mathbf{SWAP}_{2,3} = \mathrm{I}_2 \otimes \mathbf{SWAP}$$

whose validity can be verifies by the quantum circuits:




By the result of tensor product of linear maps, the matrix representations of $\mathbf{SWAP}_{1,2}$ and $\mathbf{SWAP}_{2,3}$ (relative to the ordered basis $\mathcal{B}_3$) are given by

$$\mathbf{SWAP}_{1,2} = \mathbf{SWAP} \otimes \mathrm{I}_2 = \begin{bmatrix} \mathrm{I}_2 & & \\ & \mathrm{I}_2 & \\ & & \mathrm{I}_2 \end{bmatrix} = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{bmatrix}$$

and

$$\mathbf{SWAP}_{2,3} = \mathrm{I}_2 \otimes \mathbf{SWAP} = \begin{bmatrix} \mathbf{SWAP} & \\ & \mathbf{SWAP} \end{bmatrix} = \begin{bmatrix} 1 & & & & & & & \\ & & 1 & & & & & \\ & 1 & & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & & 1 & \\ & & & & & 1 & & \\ & & & & & & & 1 \end{bmatrix}.$$

To compute the matrix representation of $\mathbf{SWAP}_{1,3}$ (on a 3-qubit system), we write $|x\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$, $|y\rangle = \beta_0|0\rangle + \beta_1|1\rangle$, $|z\rangle = \gamma_0|0\rangle + \gamma_1|1\rangle$ and find that

$$\begin{aligned} \mathbf{SWAP}_{1,3}|x\rangle \otimes |y\rangle \otimes |z\rangle &= |z\rangle \otimes |y\rangle \otimes |x\rangle \\ &= \alpha_0\beta_0\gamma_0|0\rangle + \alpha_1\beta_0\gamma_0|1\rangle + \alpha_0\beta_1\gamma_0|2\rangle + \alpha_1\beta_1\gamma_0|3\rangle \\ &\quad + \alpha_0\beta_0\gamma_1|4\rangle + \alpha_1\beta_0\gamma_1|5\rangle + \alpha_0\beta_1\gamma_1|6\rangle + \alpha_1\beta_1\gamma_1|7\rangle\,; \end{aligned}$$

thus

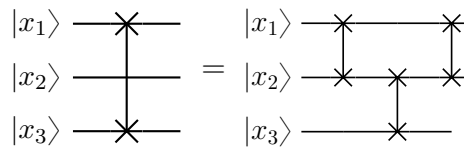
$$\mathbf{SWAP}_{1,3} \begin{bmatrix} \alpha_0\beta_0\gamma_0 \\ \alpha_0\beta_0\gamma_1 \\ \alpha_0\beta_1\gamma_0 \\ \alpha_0\beta_1\gamma_1 \\ \alpha_1\beta_0\gamma_0 \\ \alpha_1\beta_0\gamma_1 \\ \alpha_1\beta_1\gamma_0 \\ \alpha_1\beta_1\gamma_1 \end{bmatrix} = \begin{bmatrix} \alpha_0\beta_0\gamma_0 \\ \alpha_1\beta_0\gamma_0 \\ \alpha_0\beta_1\gamma_0 \\ \alpha_1\beta_1\gamma_0 \\ \alpha_0\beta_0\gamma_1 \\ \alpha_1\beta_0\gamma_1 \\ \alpha_0\beta_1\gamma_1 \\ \alpha_1\beta_1\gamma_1 \end{bmatrix}$$

Therefore, the matrix representation of $\mathbf{SWAP}_{1,3}$ (relative to the ordered basis $\mathcal{B}_3$) is given by

$$\mathbf{SWAP}_{1,3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

We note that the matrix representation of $\mathbf{SWAP}_{1,3}$ can also be computed using the identity

$$\mathbf{SWAP}_{1,3} = \mathbf{SWAP}_{1,2} \cdot \mathbf{SWAP}_{2,3} \cdot \mathbf{SWAP}_{1,2}\,.$$




**Example 3.99** (The controlled-not gate)**.** In quantum computing, the controlled-not gate is a 2-qubit gate defined by

$$|x\rangle \otimes |y\rangle \mapsto \begin{cases} |x\rangle \otimes |y\rangle & \text{if } |x\rangle = |0\rangle\,, \\ |x\rangle \otimes (\mathbf{NOT}|y\rangle) & \text{if } |x\rangle = |1\rangle\,, \end{cases}$$

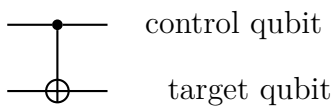
where **NOT** is the NOT gate defined by $\mathbf{NOT}|0\rangle = |1\rangle$ and $\mathbf{NOT}|1\rangle = |0\rangle$. Write $|x\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ and $|y\rangle = \beta_0|0\rangle + \beta_1|1\rangle$. Then the controlled-not gate maps $\big(\alpha_0|0\rangle + \alpha_1|1\rangle\big) \otimes \big(\beta_0|0\rangle + \beta_1|1\rangle\big)$ to

$$\begin{aligned} &\alpha_0|0\rangle \otimes |y\rangle + \alpha_1|1\rangle \otimes (|1\rangle \oplus |y\rangle) = \alpha_0|0\rangle \otimes \big(\beta_0|0\rangle + \beta_1|1\rangle\big) + \alpha_1|1\rangle \otimes \big(\beta_0|1\rangle + \beta_1|0\rangle\big) \\ &\quad = \alpha_0\beta_0|0\rangle \otimes |0\rangle + \alpha_0\beta_1|0\rangle \otimes |1\rangle + \alpha_1\beta_0|1\rangle \otimes |1\rangle + \alpha_1\beta_1|1\rangle \otimes |0\rangle \end{aligned}$$

so that the controlled-not gate, in terms of qubit state vector representation, maps $\begin{bmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_0 \\ \alpha_1\beta_1 \end{bmatrix}$ to $\begin{bmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_1 \\ \alpha_1\beta_0 \end{bmatrix}$ for all $(\alpha_0, \beta_0), (\alpha_1, \beta_1)$ on the Bloch sphere; thus the matrix representation (relative to the ordered basis $\mathcal{B}_2$) is a $4 \times 4$ matrix given by

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

The quantum circuit symbol for **CNOT** is



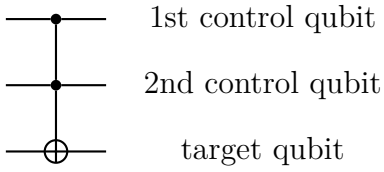

**Example 3.100** (The TOFFOLI gate)**.** In quantum computing, the Toffoli gate, also called the controlled-controlled-not gate, is a 3-qubit gate defined by

$$|x\rangle \otimes |y\rangle \otimes |z\rangle \mapsto \begin{cases} |x\rangle \otimes |y\rangle \otimes (\mathbf{NOT}|z\rangle) & \text{if } |x\rangle = |y\rangle = |1\rangle\,, \\ |x\rangle \otimes |y\rangle \otimes |z\rangle & \text{otherwise}\,. \end{cases}$$

The matrix representation of the Toffoli gate (relative to the ordered basis $\mathcal{B}_3$) is a $8 \times 8$ matrix given by

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

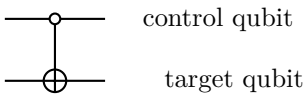
The quantum circuit symbol for **CCNOT** is




We always use $\bullet$ to denote a control qubit that activates the operation on the target qubit when the value is 1. Another kind of control qubit that activates the operation on the target qubit when the value is 0 is denoted by the symbol $\circ$. For example, the 2-qubit quantum gate

$$|x\rangle \otimes |y\rangle \mapsto \begin{cases} |x\rangle \otimes |y\rangle & \text{if } |x\rangle = |1\rangle, \\ |x\rangle \otimes (\mathrm{X}|y\rangle) & \text{if } |x\rangle = |0\rangle, \end{cases}$$

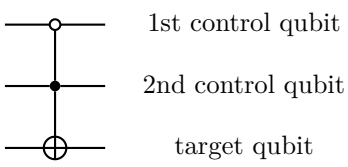
is symbolized by



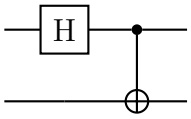

and the 3-qubit quantum gate

$$|x\rangle \otimes |y\rangle \otimes |z\rangle \mapsto \begin{cases} |x\rangle \otimes |y\rangle \otimes (\mathrm{X}|z\rangle) & \text{if } |x\rangle = |0\rangle \text{ and } |y\rangle = |1\rangle, \\ |x\rangle \otimes |y\rangle \otimes |z\rangle & \text{otherwise}, \end{cases}$$

will be symbolized by



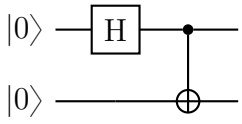

**Example 3.101** (Entanglement)**.** Consider the quantum circuit




The matrix representation of the quantum circuit above is

$$\mathbf{CNOT}(\mathrm{H}\otimes\mathrm{I}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \end{bmatrix}$$

so that the quantum circuit




produces an entangled quantum register $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ since

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}.$$

**Example 3.102** (CNOT gate on $n$-qubit system)**.** In an $n$-qubit system, we use $\mathbf{CNOT}_{i,j}$ to denote the contorlled-not gate whose control qubit is the $i$-th qubit while the target qubit is the $j$-th qubit; that is,

$$\begin{aligned} &\mathbf{CNOT}_{i,j}\big(|x_1\rangle\otimes|x_2\rangle\otimes\cdots\otimes|x_n\rangle\big) \\ &\quad= \begin{cases} |x_1\rangle\otimes\cdots\otimes|x_{j-1}\rangle\otimes(\mathrm{X}|x_j\rangle)\otimes|x_{j+1}\rangle\otimes\cdots\otimes|x_n\rangle & \text{if } |x_i\rangle = |1\rangle, \\ |x_1\rangle\otimes\cdots\otimes|x_n\rangle & \text{if } |x_i\rangle = |0\rangle, \end{cases} \end{aligned}$$

where X is the **NOT** gate. We note that $\mathbf{CNOT}_{i,j}$ is perfectly defined operator as long as $i \neq j$ and $i, j \leqslant n$. On the other hand, the matrix representation for $\mathbf{CNOT}_{i,j}$ is a $2^n \times 2^n$ matrix which essentially depends on the number of qubits in a qubit system that **CNOT** gate acts on. In the following, when talking about the matrix representation of $\mathbf{CNOT}_{i,j}$,

we always assume that it is a $2^k \times 2^k$ matrix, where $k = \max\{i, j\}$, and still use $\mathbf{CNOT}_{i,j}$, instead of $\big[\mathbf{CNOT}_{i,j}\big]$, to denote the matrix representation of $\mathbf{CNOT}_{i,j}$.

We first consider $\mathbf{CNOT}_{i,n}$ on an $n$-qubit system, where $1 \leqslant i < n$. The keys for computing the matrix representation of $\mathbf{CNOT}_{i,n}$ are the two identities

$$\mathbf{CNOT}_{i,n} = \mathrm{I}_2 \otimes \mathbf{CNOT}_{i-1,n-1} = \mathrm{blkdiag}(\mathbf{CNOT}_{i-1,n-1}, \mathbf{CNOT}_{i-1,n-1})\,, \tag{3.19a}$$

$$\mathbf{CNOT}_{1,n} = \mathbf{SWAP}_{1,2} \cdot \mathbf{CNOT}_{2,n} \cdot \mathbf{SWAP}_{1,2}\,. \tag{3.19b}$$

The validity of (3.19) can be easily verifies by the corresponding quantum circuits:

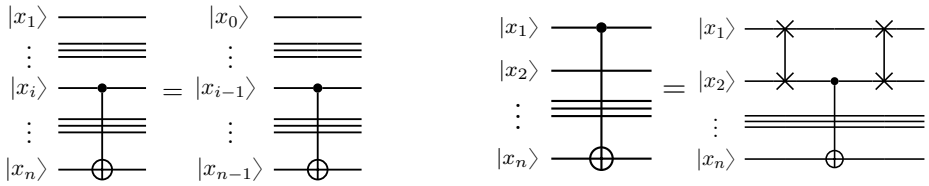



Figure 3.1: The quantum circuits to explain (3.19)

We first show that for all $n \in \mathbb{N}$,

$$\mathbf{CNOT}_{1,n+1} = \mathrm{blkdiag}\big(\underbrace{\mathrm{I}_2, \mathrm{I}_2, \cdots, \mathrm{I}_2}_{2^{n-1}\text{ copies of }\mathrm{I}_2}, \underbrace{\mathrm{X}, \mathrm{X}, \cdots, \mathrm{X}}_{2^{n-1}\text{ copies of X}}\big)\,. \tag{3.20}$$

To see (3.20), we note that $\mathbf{CNOT}_{1,2} = \mathbf{CNOT} = \mathrm{diag}(\mathrm{I}_2, \mathrm{X})$, and (3.19) shows that

$$\begin{aligned}
\mathbf{CNOT}_{1,3} &= \mathbf{SWAP}_{1,2} \cdot \mathbf{CNOT}_{2,3} \cdot \mathbf{SWAP}_{1,2} \\
&= (\mathbf{SWAP} \otimes \mathrm{I}_2) \cdot (\mathrm{I}_2 \otimes \mathbf{CNOT}) \cdot (\mathbf{SWAP} \otimes \mathrm{I}_2) \\
&= \begin{bmatrix} \mathrm{I}_2 & & & \\ & & \mathrm{I}_2 & \\ & \mathrm{I}_2 & & \\ & & & \mathrm{I}_2 \end{bmatrix} \begin{bmatrix} \mathrm{I}_2 & & & \\ & \mathrm{X} & & \\ & & \mathrm{I}_2 & \\ & & & \mathrm{X} \end{bmatrix} \begin{bmatrix} \mathrm{I}_2 & & & \\ & & \mathrm{I}_2 & \\ & \mathrm{I}_2 & & \\ & & & \mathrm{I}_2 \end{bmatrix} = \begin{bmatrix} \mathrm{I}_2 & & & \\ & \mathrm{I}_2 & & \\ & & \mathrm{X} & \\ & & & \mathrm{X} \end{bmatrix} \\
&= \mathrm{diag}(\mathrm{I}_2, \mathrm{I}_2, \mathrm{X}, \mathrm{X})\,.
\end{aligned}$$

Suppose that (3.20) holds for the case $n = m$. If $n = m + 1$, by writing

$$\mathrm{X}_{2^m} = \mathrm{blkdiag}\big(\underbrace{\mathrm{X}, \mathrm{X}, \cdots, \mathrm{X}}_{2^{m-1}\text{ copies of X}}\big)$$

so that $\mathbf{CNOT}_{1,m+1} = \mathrm{blkdiag}\big(\mathrm{I}_{2^m}, \mathrm{X}_{2^m}\big)$, using (3.19) we have

$$\begin{aligned}
\mathbf{CNOT}_{1,n+1} &= \mathbf{SWAP}_{1,2} \cdot \mathbf{CNOT}_{2,n+1} \cdot \mathbf{SWAP}_{1,2} \\
&= (\mathbf{SWAP} \otimes \mathrm{I}_{2^m}) \cdot (\mathrm{I}_2 \otimes \mathbf{CNOT}_{1,m+1}) \cdot (\mathbf{SWAP} \otimes \mathrm{I}_{2^m}) \\
&= \begin{bmatrix} \mathrm{I}_{2^m} & & & \\ & & \mathrm{I}_{2^m} & \\ & \mathrm{I}_{2^m} & & \\ & & & \mathrm{I}_{2^m} \end{bmatrix} \begin{bmatrix} \mathrm{I}_{2^m} & & & \\ & \mathrm{X}_{2^m} & & \\ & & \mathrm{I}_{2^m} & \\ & & & \mathrm{X}_{2^m} \end{bmatrix} \begin{bmatrix} \mathrm{I}_{2^m} & & & \\ & & \mathrm{I}_{2^m} & \\ & \mathrm{I}_{2^m} & & \\ & & & \mathrm{I}_{2^m} \end{bmatrix} \\
&= \begin{bmatrix} \mathrm{I}_{2^m} & & & \\ & \mathrm{I}_{2^m} & & \\ & & \mathrm{X}_{2^m} & \\ & & & \mathrm{X}_{2^m} \end{bmatrix} = \mathrm{blkdiag}(\mathrm{I}_{2^m}, \mathrm{I}_{2^m}, \mathrm{X}_{2^m}, \mathrm{X}_{2^m}) = \mathrm{blkdiag}(\mathrm{I}_{2^{m+1}}, \mathrm{X}_{2^{m+1}})\,.
\end{aligned}$$

Therefore, (3.20) is established by induction.

Using (3.19) we find that

$$\mathbf{CNOT}_{1,3} = \mathrm{blkdiag}(\mathrm{I}_2, \mathrm{I}_2, \mathrm{X}, \mathrm{X}) \qquad \text{and} \qquad \mathbf{CNOT}_{2,3} = \mathrm{blkdiag}(\mathrm{I}_2, \mathrm{X}, \mathrm{I}_2, \mathrm{X})\,.$$

The identities above and (3.20) further imply that

$$\mathbf{CNOT}_{1,4} = \mathrm{blkdiag}(\mathrm{I}_2, \mathrm{I}_2, \mathrm{I}_2, \mathrm{I}_2, \mathrm{X}, \mathrm{X}, \mathrm{X}, \mathrm{X})\,, \tag{3.21a}$$

$$\mathbf{CNOT}_{2,4} = \mathrm{blkdiag}(\mathbf{CNOT}_{1,3}, \mathbf{CNOT}_{1,3}) = \mathrm{blkdiag}(\mathrm{I}_2, \mathrm{I}_2, \mathrm{X}, \mathrm{X}, \mathrm{I}_2, \mathrm{I}_2, \mathrm{X}, \mathrm{X})\,, \tag{3.21b}$$

$$\mathbf{CNOT}_{3,4} = \mathrm{blkdiag}(\mathbf{CNOT}_{2,3}, \mathbf{CNOT}_{2,3}) = \mathrm{blkdiag}(\mathrm{I}_2, \mathrm{X}, \mathrm{I}_2, \mathrm{X}, \mathrm{I}_2, \mathrm{X}, \mathrm{I}_2, \mathrm{X})\,, \tag{3.21c}$$

and

$$\begin{aligned}
\mathbf{CNOT}_{1,5} &= \mathrm{blkdiag}(\mathrm{I}_2, \mathrm{I}_2, \mathrm{I}_2, \mathrm{I}_2, \mathrm{I}_2, \mathrm{I}_2, \mathrm{I}_2, \mathrm{I}_2, \mathrm{X}, \mathrm{X}, \mathrm{X}, \mathrm{X}, \mathrm{X}, \mathrm{X}, \mathrm{X}, \mathrm{X})\,, \\
\mathbf{CNOT}_{2,5} &= \mathrm{blkdiag}(\mathbf{CNOT}_{1,4}, \mathbf{CNOT}_{1,4}) \\
&= \mathrm{blkdiag}(\mathrm{I}_2, \mathrm{I}_2, \mathrm{I}_2, \mathrm{I}_2, \mathrm{X}, \mathrm{X}, \mathrm{X}, \mathrm{X}, \mathrm{I}_2, \mathrm{I}_2, \mathrm{I}_2, \mathrm{I}_2, \mathrm{X}, \mathrm{X}, \mathrm{X}, \mathrm{X})\,, \\
\mathbf{CNOT}_{3,5} &= \mathrm{blkdiag}(\mathbf{CNOT}_{2,4}, \mathbf{CNOT}_{2,4}) \\
&= \mathrm{blkdiag}(\mathrm{I}_2, \mathrm{I}_2, \mathrm{X}, \mathrm{X}, \mathrm{I}_2, \mathrm{I}_2, \mathrm{X}, \mathrm{X}, \mathrm{I}_2, \mathrm{I}_2, \mathrm{X}, \mathrm{X}, \mathrm{I}_2, \mathrm{I}_2, \mathrm{X}, \mathrm{X})\,, \\
\mathbf{CNOT}_{4,5} &= \mathrm{blkdiag}(\mathbf{CNOT}_{3,4}, \mathbf{CNOT}_{3,4}) \\
&= \mathrm{blkdiag}(\mathrm{I}_2, \mathrm{X}, \mathrm{I}_2, \mathrm{X}, \mathrm{I}_2, \mathrm{X}, \mathrm{I}_2, \mathrm{X}, \mathrm{I}_2, \mathrm{X}, \mathrm{I}_2, \mathrm{X}, \mathrm{I}_2, \mathrm{X}, \mathrm{I}_2, \mathrm{X})\,.
\end{aligned}$$

In general,

$$\mathbf{CNOT}_{i,n+1} = \mathrm{blkdiag}\big(\underbrace{\mathrm{IX}_{n-i-1}, \cdots, \mathrm{IX}_{n-i-1}}_{2^{i-1} \text{ copies of } \mathrm{IX}_{n-i-1}}\big)\,,*$$

where $\mathrm{IX}_k = \mathrm{blkdiag}\big(\underbrace{\mathrm{I}_2, \cdots, \mathrm{I}_2}_{2^k \text{ copies of } \mathrm{I}_2}, \underbrace{\mathrm{X}, \cdots, \mathrm{X}}_{2^k \text{ copies of X}}\big)$.

**Remark 3.103.** For $n \geqslant 1$, let $\big[\sigma_{i1}, \sigma_{i2}, \cdots, \sigma_{i2^n}\big]$ be the $(2^{n-i}+1)$-th row of the unnormalized Walsh-Hadamard matrix $\mathrm{H}_n$ given in Definition 3.69, then

$$\mathbf{CNOT}_{i,n+1} = \mathrm{blkdiag}\big(\mathrm{X}^{(1-\sigma_{i1})/2}, \mathrm{X}^{(1-\sigma_{i2})/2}, \cdots, \mathrm{X}^{(1-\sigma_{i2^n})/2}\big)\,,$$

where $\mathrm{X}^0 \equiv \mathrm{I}_2$, or with $f$ denotes the matrix-valued function $f(1) = \mathrm{I}_2$ and $f(-1) = \mathrm{X}$,

$$\mathbf{CNOT}_{i,n+1} = \mathrm{blkdiag}\big(f(\sigma_{i1}), f(\sigma_{i2}), \cdots, f(\sigma_{i2^n})\big)\,.$$

For example, note that

$$\mathrm{H}_3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}.$$

Since the $(2^{3-1}+1)$-th, $(2^{3-2}+1)$-th and $(2^{3-3}+1)$-th row of $\mathrm{H}_3$ are given by

$$\begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \end{bmatrix},$$
$$\begin{bmatrix} 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \end{bmatrix},$$
$$\begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix},$$

respectively, we find that

$$\mathbf{CNOT}_{1,4} = \mathrm{blkdiag}(\mathrm{I}_2, \mathrm{I}_2, \mathrm{I}_2, \mathrm{I}_2, \mathrm{X}, \mathrm{X}, \mathrm{X}, \mathrm{X})\,,$$
$$\mathbf{CNOT}_{2,4} = \mathrm{blkdiag}(\mathrm{I}_2, \mathrm{I}_2, \mathrm{X}, \mathrm{X}, \mathrm{I}_2, \mathrm{I}_2, \mathrm{X}, \mathrm{X})\,,$$
$$\mathbf{CNOT}_{3,4} = \mathrm{blkdiag}(\mathrm{I}_2, \mathrm{X}, \mathrm{I}_2, \mathrm{X}, \mathrm{I}_2, \mathrm{X}, \mathrm{I}_2, \mathrm{X})\,.$$

The row vector $\big[\sigma_{i1}, \sigma_{i2}, \cdots, \sigma_{i2^n}\big]$ defined above is called the symbol of $\mathbf{CNOT}_{i,n+1}$ in this lecture note.

**Exercise 3.104.** For given $j, n \in \mathbb{N}$ with $j < n$, write a matlab® function which generates $\mathbf{CNOT}_{j,n}$ given above. You may also want to try $\mathbf{CNOT}_{i,j;n}$ which is the matrix representation of an $n$-qubit system with $i$-th bit as the controlled bit and $j$-th bit as the target bit (where $1 \leqslant i, j \leqslant n$ but $i$ is not necessary smaller than $j$).

**Definition 3.105.** A quantum gate is called a **multi-controlled gate** if there exists some qubits, called control qubits, such that each value of the control qubits corresponds to a quantum gate acting on the rest of qubits, the target qubits.

Suppose a multi-controlled gate is an $n$-qubit gate with $m$ control bits and $(n-m)$ target bits, where $1 \leqslant k < n$. Rather than just applying a gate when all control bits are zero or one, in a multi-controlled gate the applied operation to the target qubits can be different for each of the $2^m$ possible classical values of the control qubits.

In the following examples, we consider the matrix representation of some special multi-controlled gates.

**Example 3.106.** Consider a multi-controlled gate given by

$$A(|x\rangle \otimes |y\rangle) = \begin{cases} |x\rangle \otimes U|y\rangle & \text{if } |x\rangle = |0\rangle, \\ |x\rangle \otimes V|y\rangle & \text{if } |x\rangle = |1\rangle. \end{cases}$$

Here the first qubit $|x\rangle$ is the control qubit, while the second qubit $|y\rangle$ is the target qubit. Write $|x\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ and $|y\rangle = \beta_0|0\rangle + \beta_1|1\rangle$. Then with "probability" $|\alpha_0|^2$ $A$ maps $|x\rangle \otimes |y\rangle$ to $|0\rangle \otimes (U|y\rangle)$ and with "probability" $|\alpha_1|^2$ $A$ maps $|x\rangle \otimes |y\rangle$ to $|1\rangle \otimes (V|y\rangle)$; thus

$$\begin{aligned} A(|x\rangle \otimes |y\rangle) &= \alpha_0|0\rangle \otimes (U|y\rangle) + \alpha_1|1\rangle \otimes (V|y\rangle) \\ &= \alpha_0|0\rangle \otimes \big[(u_{11}\beta_0 + u_{12}\beta_1)|0\rangle + (u_{21}\beta_0 + u_{22}\beta_1)|1\rangle\big] \\ &\quad + \alpha_1|1\rangle \otimes \big[(v_{11}\beta_0 + v_{12}\beta_1)|0\rangle + (v_{21}\beta_0 + v_{22}\beta_1)|1\rangle\big] \\ &= \alpha_0(u_{11}\beta_0 + u_{12}\beta_1)|0\rangle \otimes |0\rangle + \alpha_0(u_{21}\beta_0 + u_{22}\beta_1)|0\rangle \otimes |1\rangle \\ &\quad + \alpha_1(v_{11}\beta_0 + v_{12}\beta_1)|1\rangle \otimes |0\rangle + \alpha_1(v_{21}\beta_0 + v_{22}\beta_1)|1\rangle \otimes |1\rangle; \end{aligned}$$

thus the qubit state vector representation of $A(|x\rangle \otimes |y\rangle)$ is given by

$$\big[A(|x\rangle \otimes |y\rangle)\big] = \begin{bmatrix} \alpha_0(u_{11}\beta_0 + u_{12}\beta_1) \\ \alpha_0(u_{21}\beta_0 + u_{22}\beta_1) \\ \alpha_1(v_{11}\beta_0 + v_{12}\beta_1) \\ \alpha_1(v_{21}\alpha_1\beta_0 + v_{22}\beta_1) \end{bmatrix} = \begin{bmatrix} u_{11} & u_{12} & & \\ u_{21} & u_{22} & & \\ & & v_{11} & v_{12} \\ & & v_{21} & v_{22} \end{bmatrix} \begin{bmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_0 \\ \alpha_1\beta_1 \end{bmatrix}.$$

In other words, the matrix representation (relative to the ordered basis) is

$$[A] = \begin{bmatrix} u_{11} & u_{12} & & \\ u_{21} & u_{22} & & \\ & & v_{11} & v_{12} \\ & & v_{21} & v_{22} \end{bmatrix} = \text{blkdiag}(U, V)\,.$$

Since $U$ and $V$ are unitary, we find that $A$ is a 2-qubit gate. This kind of qubit gate is called a quantum multiplexer.

Next we consider a multi-controlled 3-qubit gate in which the control qubits are the first 2 qubits. We first consider the map A defined by

$$\mathrm{A}(|x\rangle \otimes |y\rangle \otimes |z\rangle) = \begin{cases} |x\rangle \otimes |y\rangle \otimes (U_{00}|z\rangle) & \text{if } |x\rangle \otimes |y\rangle = |0\rangle \otimes |0\rangle\,, \\ |x\rangle \otimes |y\rangle \otimes (U_{01}|z\rangle) & \text{if } |x\rangle \otimes |y\rangle = |0\rangle \otimes |1\rangle\,, \\ |x\rangle \otimes |y\rangle \otimes (U_{10}|z\rangle) & \text{if } |x\rangle \otimes |y\rangle = |1\rangle \otimes |0\rangle\,, \\ |x\rangle \otimes |y\rangle \otimes (U_{11}|z\rangle) & \text{if } |x\rangle \otimes |y\rangle = |1\rangle \otimes |1\rangle\,, \end{cases}$$

where $|x\rangle$, $|y\rangle$, $|z\rangle$ are all 1-qubit, and $U_{00}$, $U_{01}$, $U_{10}$, $U_{11}$ are $2 \times 2$ unitary matrices. Similar to the computation above, if $|x\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$, $|y\rangle = \beta_0|0\rangle + \beta_1|1\rangle$, and $|z\rangle = \gamma_0|0\rangle + \gamma_1|1\rangle$, we have

$$\mathrm{A}(|x\rangle \otimes |y\rangle \otimes |z\rangle) = \sum_{i,j=0}^{1} \alpha_i \beta_j |i\rangle \otimes |j\rangle \otimes (U_{ij}|z\rangle)$$

where the qubit state vector representation of the right-hand side is

$$\begin{bmatrix} \alpha_0\beta_0 U_{00}|z\rangle \\ \alpha_0\beta_1 U_{01}|z\rangle \\ \alpha_1\beta_0 U_{10}|z\rangle \\ \alpha_1\beta_1 U_{11}|z\rangle \end{bmatrix} = \begin{bmatrix} U_{00} & & & \\ & U_{01} & & \\ & & U_{10} & \\ & & & U_{11} \end{bmatrix} \begin{bmatrix} \alpha_0\beta_0 \begin{bmatrix} \gamma_0 \\ \gamma_1 \end{bmatrix} \\ \alpha_0\beta_1 \begin{bmatrix} \gamma_0 \\ \gamma_1 \end{bmatrix} \\ \alpha_1\beta_0 \begin{bmatrix} \gamma_0 \\ \gamma_1 \end{bmatrix} \\ \alpha_1\beta_1 \begin{bmatrix} \gamma_0 \\ \gamma_1 \end{bmatrix} \end{bmatrix}.$$

Therefore, the matrix representation of A is

$$\begin{bmatrix} U_{00} & & & \\ & U_{01} & & \\ & & U_{10} & \\ & & & U_{11} \end{bmatrix}.$$

In general, we can consider the multi-controlled $(n+1)$-qubit gate $L$ given by

$$L(|x_1\rangle\otimes\cdots\otimes|x_n\rangle\otimes|x_{n+1}\rangle)$$
$$= \begin{cases} |x_1\rangle\otimes\cdots\otimes|x_n\rangle\otimes(U_{0\cdots0}|x_{n+1}\rangle) & \text{if } |x_1\rangle\otimes\cdots\otimes|x_n\rangle = |0\rangle\otimes\cdots\otimes|0\rangle\,, \\ \vdots & \vdots \\ |x_1\rangle\otimes\cdots\otimes|x_n\rangle\otimes(U_{1\cdots1}|x_{n+1}\rangle) & \text{if } |x_1\rangle\otimes\cdots\otimes|x_n\rangle = |1\rangle\otimes\cdots\otimes|1\rangle\,. \end{cases}$$

where $U_{j_1\cdots j_n}$'s are $2\times 2$ unitary matrices for all $j_1,\cdots,j_n\in\{0,1\}^n$, and the control qubits are the first $n$ qubits. By identifying $(j_1\cdots j_n)_2$ with $j$ or more precisely,

$$j = (j_1\cdots j_n)_2 = 2^{n-1}j_1+\cdots+2j_{n-1}+j_n\,,$$

we write $U_{j_1\cdots j_n}$ as $U_j$ and $|j_1\rangle\otimes\cdots\otimes|j_n\rangle$ as $|j\rangle$ so that $L$ can be simply written as

$$L(|x\rangle\otimes|x_{n+1}\rangle) = |j\rangle\otimes(U_j|x_{n+1}\rangle) \qquad \text{if } \ |x\rangle = |j\rangle\,.$$

Suppose that $U_j = \begin{bmatrix} u_{11}^{(j)} & u_{12}^{(j)} \\ u_{21}^{(j)} & u_{22}^{(j)} \end{bmatrix}$, $|x\rangle = \alpha_0|0\rangle+\cdots+\alpha_{2^n-1}|2^n-1\rangle$ and $|x_{n+1}\rangle = \beta_0|0\rangle+\beta_1|1\rangle$. Then for $0\leqslant j\leqslant 2^n-1$,

$$U_j|x_{n+1}\rangle = (u_{11}^{(j)}\beta_0+u_{12}^{(j)}\beta_1)|0\rangle+(u_{21}^{(j)}\beta_0+u_{22}^{(j)}\beta_1)|1\rangle$$

which implies that

$$[L]:\begin{bmatrix} \alpha_0\begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix} \\ \alpha_1\begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix} \\ \vdots \\ \alpha_{2^n-1}\begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix}\end{bmatrix} \mapsto \begin{bmatrix} \alpha_0\big(u_{11}^{(0)}\beta_0+u_{12}^{(0)}\beta_1\big) \\ \alpha_0\big(u_{21}^{(0)}\beta_0+u_{22}^{(0)}\beta_1\big) \\ \alpha_1\big(u_{11}^{(1)}\beta_0+u_{12}^{(1)}\beta_1\big) \\ \alpha_1\big(u_{21}^{(1)}\beta_0+u_{22}^{(1)}\beta_1\big) \\ \vdots \\ \alpha_j\big(u_{11}^{(j)}\beta_0+u_{12}^{(j)}\beta_1\big) \\ \alpha_j\big(u_{21}^{(j)}\beta_0+u_{22}^{(j)}\beta_1\big) \\ \vdots \\ \alpha_{2^n-1}\big(u_{11}^{(2^n-1)}\beta_0+u_{12}^{(2^n-1)}\beta_1\big) \\ \alpha_{2^n-1}\big(u_{21}^{(2^n-1)}\beta_0+u_{22}^{(2^n-1)}\beta_1\big)\end{bmatrix}.$$

Therefore,

$$L(|x\rangle\otimes|x_{n+1}\rangle)=\begin{bmatrix} u_{11}^{(0)} & u_{12}^{(0)} & & & & & & \\ u_{21}^{(0)} & u_{22}^{(0)} & & & & & & \\ & & u_{11}^{(1)} & u_{12}^{(1)} & & & & \\ & & u_{21}^{(1)} & u_{22}^{(1)} & & & & \\ & & & & \ddots & & & \\ & & & & & \ddots & & \\ & & & & & & u_{11}^{(2^n-1)} & u_{12}^{(2^n-1)} \\ & & & & & & u_{21}^{(2^n-1)} & u_{22}^{(2^n-1)} \end{bmatrix}\begin{bmatrix} \alpha_0\begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix} \\ \alpha_1\begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix} \\ \vdots \\ \vdots \\ \alpha_{2^n-1}\begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix}\end{bmatrix}.$$

The $2^{n+1}\times 2^{n+1}$ matrix is the matrix representation of $L$.

**Example 3.107.** Similar to the previous example, in this example we consider a multi-controlled gate given by

$$L(|x\rangle\otimes|y\rangle)=\begin{cases}|x\rangle\otimes U|y\rangle & \text{if } |x\rangle=|0\rangle,\\ |x\rangle\otimes V|y\rangle & \text{if } |x\rangle=|1\rangle.\end{cases}$$

where the control qubit $|x\rangle$ is a 1-qubit state, the target qubit $|y\rangle$ is an $n$-qubit state, and $U$, $V$ are both $n$-qubit gates (so that $[U]$ and $[V]$ are $2^n\times 2^n$ unitary matrix).

Write $|x\rangle=\alpha_0|0\rangle+\alpha_1|1\rangle$, $|y\rangle=\beta_0|0\rangle+\cdots+\beta_{2^n-1}|2^n-1\rangle$, and $|\psi\rangle=|x\rangle\otimes|y\rangle$. Then

$$\begin{aligned}L|\psi\rangle=\alpha_0|0\rangle\otimes\left[U\left(\beta_0|0\rangle+\cdots+\beta_{2^n-1}|2^n-1\rangle\right)\right]\\ +\alpha_1|1\rangle\otimes\left[V\left(\beta_0|0\rangle+\cdots+\beta_{2^n-1}|2^n-1\rangle\right)\right]\end{aligned}$$

Since the matrix representation of $L$ satisfies

$$[L]:\begin{bmatrix}\alpha_0\begin{bmatrix}\beta_0\\ \vdots\\ \beta_{2^n-1}\end{bmatrix}\\ \alpha_1\begin{bmatrix}\beta_0\\ \vdots\\ \beta_{2^n-1}\end{bmatrix}\end{bmatrix}\mapsto\begin{bmatrix}\alpha_0[U]\begin{bmatrix}\beta_0\\ \vdots\\ \beta_{2^n-1}\end{bmatrix}\\ \alpha_1[V]\begin{bmatrix}\beta_0\\ \vdots\\ \beta_{2^n-1}\end{bmatrix}\end{bmatrix}.$$

to find the matrix representation of $L$, we let $\alpha_0=\beta_{\ell-1}=1$ for some fixed $\ell$ while $\alpha_i=\beta_j=0$ if $i\neq 0$ and $j\neq\ell$ to obtain that the $\ell$-th column of $[L]$ is given by

$$[L](:,\ell)=\begin{bmatrix}1\\0\end{bmatrix}\otimes U(:,\ell)=\begin{bmatrix}U(:,\ell)\\ \mathbf{0}_{2^m}\end{bmatrix}$$

and let $\alpha_1 = \beta_{\ell-1} = 1$ for some fixed $\ell$ while $\alpha_i = \beta_j = 0$ if $i \neq 1$ and $j \neq \ell$ to obtain that

$$[L](:, 2^n + \ell) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes V(:, \ell) = \begin{bmatrix} \mathbf{0}_{2^m} \\ V(:, \ell) \end{bmatrix},$$

where $\mathbf{0}_{2^m}$ denotes the zero vectors in $\mathbb{C}^{2^m}$. This shows that $[L] = \text{blkdiag}(U, V)$.

In general, if a multi-controlled $(n+1)$-qubit gate $L$ is defined by

$$L(|x\rangle \otimes |y\rangle) = \begin{cases} |x\rangle \otimes U_0|y\rangle & \text{if } |x\rangle = |0\rangle, \\ |x\rangle \otimes U_1|y\rangle & \text{if } |x\rangle = |1\rangle, \\ \vdots & \vdots \\ |x\rangle \otimes U_{2^k-1}|y\rangle & \text{if } |x\rangle = |2^k - 1\rangle; \end{cases}$$

that is, the controlled qubit $|x\rangle$ is an $m$-qubit state and $L(|x\rangle \otimes |y\rangle) = |x\rangle \otimes U_j|y\rangle$ if $|x\rangle = |j\rangle$. Then the matrix representation of $L$ is given by

$$[L] = \text{blkdiag}(U_0, U_1, \cdots, U_{2^m-1}).$$

since by letting $|x\rangle = |k-1\rangle$ and $|y\rangle = |\ell - 1\rangle$ for some $1 \leqslant k \leqslant 2^m$ and $1 \leqslant \ell \leqslant 2^{n-m+1}$, we have

$$[L](:, (k-1)2^m + \ell) = \mathbf{e}_k \otimes U(:, \ell),$$

where $\{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_{2^m}\}$ is the standard basis of $\mathbb{C}^{2^m}$.

**Example 3.108.** In Example 3.106, we consider the multi-controlled $(n+1)$-qubit gate in which the control qubits are the first $n$ qubits. How about if the control qubits are the last $n$ qubits? Consider multi-controlled $(n+1)$-qubit gate $L$ given by

$$L(|x_0\rangle \otimes |x_1\rangle \otimes \cdots \otimes |x_n\rangle) = \begin{cases} (U_{0\cdots0}|x_0\rangle) \otimes |x_1\rangle \otimes \cdots \otimes |x_n\rangle & \text{if } |x_1\rangle \otimes \cdots \otimes |x_n\rangle = |0\rangle \otimes \cdots \otimes |0\rangle, \\ \vdots & \vdots \\ (U_{1\cdots1}|x_0\rangle) \otimes |x_1\rangle \otimes \cdots \otimes |x_n\rangle & \text{if } |x_1\rangle \otimes \cdots \otimes |x_n\rangle = |1\rangle \otimes \cdots \otimes |1\rangle. \end{cases}$$

where $U_{j_1\cdots j_n}$'s are $2 \times 2$ unitary matrices for all $j_1, \cdots, j_n \in \{0, 1\}^n$, and the control qubits are the last $n$ qubits. By identifying $(j_1 \cdots j_n)_2$ with $j$ or more precisely,

$$j = (j_1 \cdots j_n)_2 = 2^{n-1}j_1 + \cdots + 2j_{n-1} + j_n,$$

we write $U_{j_1\cdots j_n}$ as $U_j$ and $|j_1\rangle \otimes \cdots \otimes |j_n\rangle$ as $|j\rangle$ so that $L$ can be simply written as

$$L(|x_0\rangle \otimes |x\rangle) = (U_j|x_0\rangle) \otimes |x\rangle \qquad \text{if } \ |x\rangle = |j\rangle.$$

Suppose that $U_j = \begin{bmatrix} u_{11}^{(j)} & u_{12}^{(j)} \\ u_{21}^{(j)} & u_{22}^{(j)} \end{bmatrix}$, $|x_0\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ and $|x\rangle = \beta_0|0\rangle + \cdots + \beta_{2^n-1}|2^n-1\rangle$. Then and for $0 \leqslant j \leqslant 2^n - 1$,

$$U_j|x_0\rangle = (u_{11}^{(j)}\alpha_0 + u_{12}^{(j)}\alpha_1)|0\rangle + (u_{21}^{(j)}\alpha_0 + u_{22}^{(j)}\alpha_1)|1\rangle$$

which implies that

$$[L] : \begin{bmatrix} \alpha_0 \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_{2^n-1} \end{bmatrix} \\ \alpha_1 \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_{2^n-1} \end{bmatrix} \end{bmatrix} \mapsto \begin{bmatrix} (u_{11}^{(0)}\alpha_0 + u_{12}^{(0)}\alpha_1)\beta_0 \\ \vdots \\ (u_{11}^{(j)}\alpha_0 + u_{12}^{(j)}\alpha_1)\beta_j \\ \vdots \\ (u_{11}^{(2^n-1)}\alpha_0 + u_{12}^{(2^n-1)}\alpha_1)\beta_{2^n-1} \\ (u_{21}^{(0)}\alpha_0 + u_{22}^{(0)}\alpha_1)\beta_0 \\ \vdots \\ (u_{21}^{(j)}\alpha_0 + u_{22}^{(j)}\alpha_1)\beta_j \\ \vdots \\ (u_{21}^{(2^n-1)}\alpha_0 + u_{22}^{(2^n-1)}\alpha_1)\beta_{2^n-1} \end{bmatrix}.$$

Therefore,

$$\left[L(|x_0\rangle \otimes |x\rangle)\right] = \begin{bmatrix} u_{11}^{(0)} & & & & u_{12}^{(0)} & & & \\ & u_{11}^{(1)} & & & & u_{12}^{(1)} & & \\ & & \ddots & & & & \ddots & \\ & & & u_{11}^{(2^n-1)} & & & & u_{12}^{(2^n-1)} \\ u_{21}^{(0)} & & & & u_{22}^{(0)} & & & \\ & u_{21}^{(1)} & & & & u_{22}^{(1)} & & \\ & & \ddots & & & & \ddots & \\ & & & u_{21}^{(2^n-1)} & & & & u_{22}^{(2^n-1)} \end{bmatrix} \begin{bmatrix} \alpha_0 \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{2^n-1} \end{bmatrix} \\ \alpha_1 \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{2^n-1} \end{bmatrix} \end{bmatrix}.$$

The $2^{n+1} \times 2^{n+1}$ matrix is the matrix representation of $L$. In particular, if $U_j$ is a rotation matrix of the form $U_j = \mathrm{R}(2\theta_{j+1}) = \begin{bmatrix} \cos\theta_{j+1} & -\sin\theta_{j+1} \\ \sin\theta_{j+1} & \cos\theta_{j+1} \end{bmatrix}$ (here we label $U$ from 0 to

$2^n - 1$ but label $\theta$ from 1 to $2^n$), then

$$[L] = \begin{bmatrix} \cos\theta_1 & & & & -\sin\theta_1 & & & \\ & \cos\theta_2 & & & & -\sin\theta_2 & & \\ & & \ddots & & & & \ddots & \\ & & & \cos\theta_{2^n} & & & & -\sin\theta_{2^n} \\ \sin\theta_1 & & & & \cos\theta_1 & & & \\ & \sin\theta_2 & & & & \cos\theta_2 & & \\ & & \ddots & & & & \ddots & \\ & & & \sin\theta_{2^n} & & & & \cos\theta_{2^n} \end{bmatrix}$$

which takes the block structure $\begin{bmatrix} C & -S \\ S & C \end{bmatrix}$. A matrix of this form will play important role in Section 3.8.3.

**Example 3.109.** In this example we consider a special multi-controlled $(n+1)$-qubit gate $A_j$ defined by

$$A_j(|x_0\rangle \otimes \cdots \otimes |x_n\rangle) = |x_0\rangle \otimes \cdots \otimes |x_{j-1}\rangle \otimes (\mathrm{R}_z(\theta_k)|x_j\rangle) \otimes |x_{j+1}\rangle \otimes \cdots \otimes |x_n\rangle$$

if $(x_0 \cdots x_{j-1}x_{j+1} \cdots x_n)_2 = k$, where $\mathrm{R}_z$ is the rotation about $z$-axis given by

$$\mathrm{R}_z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}.$$

This is a multi-controlled gate with $n$ control qubits and the target qubit is the $|x_j\rangle$ qubit, and is sometimes denoted by $F^n_{j+1}(\mathrm{R}_z)$ (since the target qubit $|x_j\rangle$ is the $(j+1)$-th qubit counting from the highest/left-most qubit).

Example 3.106 establishes the case $j = 0$, while Example 3.108 established the case $j = n$. Now we consider the case $1 \leqslant j < n$. We first consider the case $j = 1$. In this case, we note that $A_{n-1} = \mathbf{SWAP}_{n,n+1} \cdot A_n \cdot \mathbf{SWAP}_{n,n+1}$, where the operator $\mathbf{SWAP}_{n,n+1}$ swaps the position of the $n$-th and the $(n+1)$-th qubit, and $A_n$ is the multi-controlled $(n+1)$-qubit gate introduced in Example 3.106 with $U_k = \mathrm{R}_z(\theta_k)$. Example 3.106 shows that the matrix representation of $A_n$ is given by

$$\begin{aligned} [A_n] &= \mathrm{blkdiag}\big(\mathrm{R}_z(\theta_1), \mathrm{R}_z(\theta_2), \cdots, \mathrm{R}_z(\theta_{2^n})\big) \\ &= \mathrm{diag}\big(e^{-i\theta_1/2}, e^{i\theta_1/2}, e^{-i\theta_2/2}, e^{i\theta_2/2}, \cdots, e^{-i\theta_{2^n}/2}, e^{i\theta_{2^n}/2}\big); \end{aligned}$$

thus by the fact that

$$\mathbf{SWAP}_{n,n+1} = \mathrm{I}_{2^{n-1}} \otimes \mathbf{SWAP} = \begin{bmatrix} \mathbf{SWAP} & & & \\ & \mathbf{SWAP} & & \\ & & \ddots & \\ & & & \mathbf{SWAP} \end{bmatrix}$$

and

$$\begin{aligned} &\mathbf{SWAP} \cdot \mathrm{diag}(a,b,c,d) \cdot \mathbf{SWAP} \\ &\quad = \begin{bmatrix} 1 & & & \\ & & 1 & \\ & 1 & & \\ & & & 1 \end{bmatrix} \begin{bmatrix} a & & & \\ & b & & \\ & & c & \\ & & & d \end{bmatrix} \begin{bmatrix} 1 & & & \\ & & 1 & \\ & 1 & & \\ & & & 1 \end{bmatrix} = \begin{bmatrix} a & & & \\ & c & & \\ & & b & \\ & & & d \end{bmatrix} = \mathrm{diag}(a,c,b,d)\,, \end{aligned}$$

we conclude that

$$\begin{aligned} [A_{n-1}] &= \mathbf{SWAP}_{n,n+1} \cdot [A_n] \cdot \mathbf{SWAP}_{n,n+1} \\ &= \begin{bmatrix} e^{-i\theta_1/2} & & & & & & & & \\ & e^{-i\theta_2/2} & & & & & & & \\ & & e^{i\theta_1/2} & & & & & & \\ & & & e^{i\theta_2/2} & & & & & \\ & & & & e^{-i\theta_3/2} & & & & \\ & & & & & e^{-i\theta_4/2} & & & \\ & & & & & & e^{i\theta_3/2} & & \\ & & & & & & & e^{i\theta_4/2} & \\ & & & & & & & & \ddots \end{bmatrix}. \end{aligned}$$

We note that $[A_{n-1}]$ takes the form

$$\mathrm{blkdiag}(Q_1, Q_2, \cdots, Q_{2^{n-1}})\,,$$

where for each $1 \leqslant k \leqslant 2^{n-1}$, $Q_k = \mathrm{diag}\big(e^{-i\theta_{2k-1}/2}, e^{-i\theta_{2k}/2}, e^{i\theta_{2k-1}/2}, e^{i\theta_{2k}/2}\big)$ for some $\theta_1, \cdots, \theta_{2^n} \in \mathbb{R}$.

In the following, for simplicity we will only write the sign and the sub-index of the angle to express the matrix. For example, we will write

$$[A_n] = \mathrm{diag}(-1, +1, -2, +2, \cdots, -2^n, +2^n)$$

and

$$[A_{n-1}] = \mathrm{diag}(-1, -2, +1, +2, -3, -4, +3, +4, \cdots, -(2^n-1), -2^n, +(2^n-1), +2^n)\,.$$

Now we consider $A_{n-2}$. Similar to the previous case, we have

$$A_{n-2} \equiv \mathbf{SWAP}_{n-1,n} \cdot A_{n-1} \cdot \mathbf{SWAP}_{n-1,n} \,.$$

Note that

$$\begin{aligned}
\mathbf{SWAP}_{n-1,n} &= \mathrm{I}_{2^{n-2}} \otimes \mathbf{SWAP} \otimes \mathrm{I}_2 \\
&= \mathrm{blkdiag}(\underbrace{\mathbf{SWAP} \otimes \mathrm{I}_2, \mathbf{SWAP} \otimes \mathrm{I}_2}_{2^{n-2} \text{ copies of } \mathbf{SWAP} \otimes \mathrm{I}_2}, \cdots, \mathbf{SWAP} \otimes \mathrm{I}_2) \\
&= \begin{bmatrix}
\mathrm{I}_2 & & & & & & & \\
& \mathrm{I}_2 & & & & & & \\
& & \mathrm{I}_2 & & & & & \\
& & & \mathrm{I}_2 & & & & \\
& & & & \mathrm{I}_2 & & & \\
& & & & & \mathrm{I}_2 & & \\
& & & & & & \mathrm{I}_2 & \\
& & & & & & & \ddots
\end{bmatrix}
\end{aligned}$$

and

$$\begin{aligned}
&(\mathbf{SWAP} \otimes \mathrm{I}_2) \cdot \mathrm{diag}(a, b, c, d, e, f, g, h) \cdot (\mathbf{SWAP} \otimes \mathrm{I}_2) \\
&\quad = \begin{bmatrix} \mathrm{I}_2 & & & \\ & & \mathrm{I}_2 & \\ & \mathrm{I}_2 & & \\ & & & \mathrm{I}_2 \end{bmatrix} \begin{bmatrix} \mathrm{diag}(a,b) & & & \\ & \mathrm{diag}(c,d) & & \\ & & \mathrm{diag}(e,f) & \\ & & & \mathrm{diag}(g,h) \end{bmatrix} \begin{bmatrix} \mathrm{I}_2 & & & \\ & \mathrm{I}_2 & & \\ & & & \mathrm{I}_2 \\ & & \mathrm{I}_2 & \end{bmatrix} \\
&\quad = \begin{bmatrix} \mathrm{diag}(a,b) & & & \\ & \mathrm{diag}(e,f) & & \\ & & \mathrm{diag}(c,d) & \\ & & & \mathrm{diag}(g,h) \end{bmatrix} = \mathrm{diag}(a, b, e, f, c, d, g, h) \,.
\end{aligned}$$

Therefore, $[A_{n-2}]$ is obtained by

> grouping the diagonal entries of $[A_{n-1}]$ in groups of successive eight entries, and exchanging the pair of the third and the fourth entries with the pair of the fifth and the sixth entries in each group;

thus we conclude that

$$[A_{n-2}] = \mathrm{diag}(-1, -2, -3, -4, +1, +2, +3, +4, -5, -6, -7, -8, +5, +6, +7, +8, \cdots) \,.$$

We note that $[A_{n-2}]$ takes the form

$$\mathrm{blkdiag}(Q_1, Q_2, \cdots, Q_{2^{n-2}}),$$

where for each $1 \leqslant k \leqslant 2^{n-2}$,

$$Q_k = \mathrm{diag}\big(e^{-i\theta_{4k-3}/2}, e^{-i\theta_{4k-2}/2}, e^{-i\theta_{4k-1}/2}, e^{-i\theta_{4k}/2}, e^{i\theta_{4k-3}/2}, e^{i\theta_{4k-2}/2}, e^{i\theta_{4k-1}/2}, e^{i\theta_{4k}/2}\big)$$

for some $\theta_1, \cdots, \theta_{2^n} \in \mathbb{R}$.

In general, for each $j$ we have $A_{j-1} = \mathbf{SWAP}_{j,j+1} \cdot A_j \cdot \mathbf{SWAP}_{j,j+1}$ and the fact that $\mathbf{SWAP}_{j,j+1} = \mathrm{I}_{2^{j-1}} \otimes \mathbf{SWAP} \otimes \mathrm{I}_{2^{n-j}}$ implies that $[A_{n-j-1}]$ is obtained by

> grouping the diagonal entries of $[A_{n-j}]$ in groups of successive $2^{j+2}$ entries, dividing each group into 4 blocks of consecutive $2^j$ entries, and exchanging the two blocks in the middle

so that

$$[A_{n-j}] = \mathrm{diag}(-1, \cdots, -2^j, +1, \cdots, +2^j, -(2^j+1), \cdots, -2^{j+1}, +(2^j+1), \cdots, +2^{j+1}, \cdots).$$

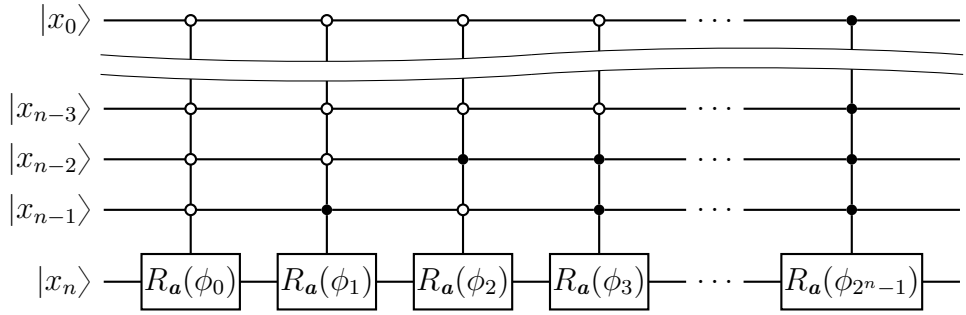
The identity above can be proved rigorously by induction.

**Definition 3.110.** An $(n+1)$-qubit gate $L$ is called a multi-controlled rotation gate of type $F^n_{j+1}(R_{\boldsymbol{a}})$ if there exist a unit vector $\boldsymbol{a} \in \mathbb{R}^3$ and real numbers $\phi_0, \cdots, \phi_{2^n-1}$ such that

$$L(|x_0\rangle \otimes \cdots \otimes |x_n\rangle) = |x_0\rangle \otimes \cdots \otimes |x_{j-1}\rangle \otimes (R_{\boldsymbol{a}}(\phi_k)|x_j\rangle) \otimes |x_{j+1}\rangle \otimes \cdots \otimes |x_n\rangle$$

if $(x_0 \cdots x_{j-1} x_{j+1} \cdots x_n)_2 = k$, where for unit vector $\boldsymbol{a} = (a_x, a_y, a_z)$, $R_{\boldsymbol{a}}$ is a 1-qubit gate given in Definition 2.10.

**Remark 3.111.** One possible quantum circuit for a multi-controlled rotation gate of type $F^n_{n+1}(R_{\boldsymbol{a}})$, in term of 1-qubit quantum gate $R_{\boldsymbol{a}}(\phi)$, is given by




and quantum circuit for a multi-controlled rotation gate of type $F^n_j(R_{\boldsymbol{a}})$ can be constructed using **SWAP** gates and the quantum circuit given above.

## 3.8 Unitary Decomposition

Unitary decomposition is the process of translating an arbitrary unitary gate into a specific (universal) set of single and two-qubit gates. Unitary decomposition is necessary because it is not otherwise possible to execute an arbitrary quantum gate on either a simulator or quantum accelerator. This makes it a required feature for algorithms that use any type of gate that is not supported by the target platform, or just produce an arbitrary unitary gate that will need to be translated.

In order to decompose all possible unitary matrices into quantum gates, a universal gate set is selected. This means the decomposition will result in circuits with (only) the following three gates: rotations around the $Y$ and $Z$ axis by an arbitrary angle, the $\mathrm{R}_z(\theta)$ and $\mathrm{R}_y(\theta)$ gates, and the controlled not, the **CNOT** gate:

$$\mathrm{R}_y(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}, \quad \mathrm{R}_z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}, \quad \textbf{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

### 3.8.1 1-qubit gate decomposition

We first focus on expressing 1 qubit gates (or $2 \times 2$ unitary matrices) in terms of product of qubit gates from the set

$$\left\{ \mathrm{R}_y(\theta), \mathrm{R}_z(\theta), \mathrm{Ph}(\theta) \,\middle|\, \theta \in \mathbb{R} \right\},$$

where Ph is the global phase gate given by $\mathrm{Ph}(\theta) = \mathrm{diag}(e^{i\theta}, e^{i\theta})$.

**Theorem 3.112.** *For every 1-qubit gate $U$, there exist real numbers $\delta$, $\theta$, $\xi$ and $\eta$ such that*

$$U = \mathrm{Ph}(\delta)\mathrm{R}_z(\xi)\mathrm{R}_y(\theta)\mathrm{R}_z(\eta) = \mathrm{R}_z(\xi)\mathrm{R}_y(\theta)\mathrm{R}_z(\eta)\mathrm{Ph}(\delta)\,.$$

*Proof.* Let $U = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ be a $2 \times 2$ unitary matrix. Corollary 3.67 implies that there exists $\delta \in \mathbb{R}$ such that

$$\det(U) = e^{2i\delta}\,. \tag{3.22}$$

Define $V \equiv e^{-i\delta}U$. Then $V$ is also a unitary matrix; thus using the fact that $V^\dagger = V^{-1}$ and $\det(V) = 1$ we find that $V$ takes the form

$$V = \begin{bmatrix} \alpha & -\bar{\beta} \\ \beta & \bar{\alpha} \end{bmatrix}.$$

This further implies that $U$ takes the form

$$U = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = e^{i\delta} \begin{bmatrix} \alpha & -\overline{\beta} \\ \beta & \overline{\alpha} \end{bmatrix},$$

The fact that $|\alpha|^2 + |\beta|^2 = 1$ allows us to set $\alpha = e^{i\mu}\cos\frac{\theta}{2}$ and $\beta = e^{i\nu}\sin\frac{\theta}{2}$ for some $\mu$, $\nu$ and $\theta \in \mathbb{R}$. Let $\xi = \nu - \mu$ and $\eta = -\mu - \nu$. Then

$$\begin{aligned} \mathrm{R}_z(\xi)\mathrm{R}_y(\theta)\mathrm{R}_z(\eta) &= \begin{bmatrix} e^{-i\frac{\xi}{2}} & 0 \\ 0 & e^{i\xi/2} \end{bmatrix} \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \begin{bmatrix} e^{-i\eta/2} & 0 \\ 0 & e^{i\eta/2} \end{bmatrix} \\ &= \begin{bmatrix} e^{-i\xi/2}\cos\frac{\theta}{2} & -e^{-i\xi/2}\sin\frac{\theta}{2} \\ e^{i\xi/2}\sin\frac{\theta}{2} & e^{i\xi/2}\cos\frac{\theta}{2} \end{bmatrix} \begin{bmatrix} e^{-i\eta/2} & 0 \\ 0 & e^{i\eta/2} \end{bmatrix} \\ &= \begin{bmatrix} e^{i\frac{\xi+\eta}{2}}\cos\frac{\theta}{2} & -e^{-i\frac{\xi-\eta}{2}}\sin\frac{\theta}{2} \\ e^{i\frac{\xi-\eta}{2}}\sin\frac{\theta}{2} & e^{-i\frac{\xi+\eta}{2}}\cos\frac{\theta}{2} \end{bmatrix} = \begin{bmatrix} \alpha & -\overline{\beta} \\ \beta & \overline{\alpha} \end{bmatrix} \end{aligned}$$

which concludes the theorem. □

• **Algorithm of 1-qubit gate decomposition**

Let $U$ be a 1-qubit gate (or equivalently, $2 \times 2$ unitary matrix).

**Step 1**: Find $\delta \in \mathbb{R}$ such that $\det(U) = e^{2i\delta}$.

**Step 2**: Find $\mu, \nu, \theta$ such that $e^{i\mu}\cos\frac{\theta}{2} = ae^{-i\delta}$ and $e^{i\nu}\sin\frac{\theta}{2} = ce^{-i\delta}$.

**Step 3**: $U = \mathrm{Ph}(\delta)\mathrm{R}_z(\nu - \mu)\mathrm{R}_y(\theta)\mathrm{R}_z(-\mu - \nu)$.

**Example 3.113.** Consider the decomposition of the X-gate. We follow the procedure give above.

**Step 1**: Since $\det(\mathrm{X}) = -1$, we choose $\delta = -\frac{\pi}{2}$ so that $\det(\mathrm{X}) = e^{2i\delta}$.

**Step 2**: Since $a = 0$ and $c = 1$, we choose $\theta = \pi$ and $\nu = \frac{\pi}{2}$ (and $\mu$ can be given arbitrarily, so we choose $\mu = 0$) so that $e^{i\mu}\cos\frac{\theta}{2} = 0 = ae^{-i\delta}$ and $e^{i\nu}\sin\frac{\theta}{2} = e^{i\frac{\pi}{2}} = ce^{-i\delta}$.

**Step 3**: $\mathrm{X} = \mathrm{Ph}\big(-\frac{\pi}{2}\big)\mathrm{R}_z\big(\frac{\pi}{2}\big)\mathrm{R}_y(\pi)\mathrm{R}_z\big(-\frac{\pi}{2}\big)$ and we verify this identity as follows:

$$\begin{aligned}
&\mathrm{Ph}\big(-\tfrac{\pi}{2}\big)\mathrm{R}_z\big(\tfrac{\pi}{2}\big)\mathrm{R}_y(\pi)\mathrm{R}_z\big(-\tfrac{\pi}{2}\big) \\
&\quad= \begin{bmatrix} e^{-i\pi/2} & 0 \\ 0 & e^{-i\pi/2} \end{bmatrix} \begin{bmatrix} e^{-i\pi/4} & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} e^{i\pi/4} & 0 \\ 0 & e^{-i\pi/4} \end{bmatrix} \\
&\quad= \begin{bmatrix} e^{-i3\pi/4} & 0 \\ 0 & e^{-i\pi/4} \end{bmatrix} \begin{bmatrix} 0 & -e^{-i\pi/4} \\ e^{i\pi/4} & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \mathrm{X}\,.
\end{aligned}$$

### 3.8.2 Singular value decomposition

Recall the spectral theorem from linear algebra given below:

**Theorem 3.114** (Spectral)**.** *Let $A$ be a Hermitian matrix; that is, $A = A^\dagger$. Then there exists unitary matrix $U$ and a real diagonal matrix $D$ such that $A = UDU^\dagger$.*

We note that the columns of $U$ are eigenvectors of $A$ and the diagonal elements of $D$ are eigenvalues of $A$. In fact, if $A = UDU^\dagger$, then $AU = UD$ so that if $\boldsymbol{v}_j$ is the $j$-th column of $U$ and $\lambda_j$ is the $(j,j)$-entry of $D$, then $A\boldsymbol{v}_j = \lambda_j \boldsymbol{v}_j$.

**Remark 3.115.** The spectral theorem extends to a more general class of matrices, the normal matrices. One can show that $A$ is normal (that is, $AA^\dagger = A^\dagger A$) if and only if there exists a unitary matrix $U$ and a diagonal matrix $D$ such that $A = UDU^\dagger$. Here the diagonal matrix $D$ can be complex.

Let $A$ be a complex $m \times n$ square matrix. Then $A^\dagger A \in \mathbb{C}^{n\times n}$ and $AA^\dagger \in \mathbb{C}^{m\times m}$. Moreover,

1. $A^\dagger A$ and $AA^\dagger$ are both hermitian since

$$(A^\dagger A)^\dagger = A^\dagger (A^\dagger)^\dagger = A^\dagger A \qquad \text{and} \qquad (AA^\dagger)^\dagger = (A^\dagger)^\dagger A^\dagger = AA^\dagger\,.$$

2. $A^\dagger A$ and $AA^\dagger$ are both positive semi-definite since

$$\langle \boldsymbol{x}, A^\dagger A\boldsymbol{x}\rangle = \langle A\boldsymbol{x}, A\boldsymbol{x}\rangle = \|A\boldsymbol{x}\|^2 \geqslant 0 \qquad \forall\, \boldsymbol{x} \in \mathbb{C}^n$$

and

$$\langle \boldsymbol{x}, AA^\dagger\boldsymbol{x}\rangle = \langle A^\dagger\boldsymbol{x}, A^\dagger\boldsymbol{x}\rangle = \|A^\dagger\boldsymbol{x}\|^2 \geqslant 0 \qquad \forall\, \boldsymbol{x} \in \mathbb{C}^m\,.$$

Therefore, Theorem 3.114 implies that there exist $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n \geqslant 0$ and an orthonormal basis $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n\}$ of $\mathbb{C}^n$ such that

$$A^\dagger A \boldsymbol{v}_k = \lambda_k \boldsymbol{v}_k \qquad \forall\, 1 \leqslant k \leqslant n\,.$$

Let $\sigma_k = \sqrt{\lambda_k}$, and $r = \#\big\{1 \leqslant k \leqslant n \,\big|\, \lambda_k > 0\big\}$; that is, $A^\dagger A$ has $r$ non-zero eigenvalues. Define

$$\boldsymbol{u}_k = \frac{1}{\sigma_k} A \boldsymbol{v}_k \qquad \text{for } 1 \leqslant k \leqslant r\,.$$

Then

1. $\boldsymbol{u}_k \neq \boldsymbol{0}$ for all $1 \leqslant k \leqslant r$. Moreover,

   $$\|A\boldsymbol{v}_j\|^2 = \langle A\boldsymbol{v}_j, A\boldsymbol{v}_j\rangle = \langle \boldsymbol{v}_j, A^\dagger A \boldsymbol{v}_j\rangle = \langle \boldsymbol{v}_j, \lambda_j \boldsymbol{v}_j\rangle = \lambda_j\,;$$

   thus the fact that $A^\dagger A$ and $A$ have the same null space implies that $\big\{\boldsymbol{v}_{r+1}, \cdots, \boldsymbol{v}_n\big\}$ is an orthonormal basis of the null space of $A$.

2. $\{\boldsymbol{u}_1, \cdots, \boldsymbol{u}_r\}$ is an orthonormal set since

   $$\big\langle \boldsymbol{u}_k, \boldsymbol{u}_\ell\big\rangle = \frac{1}{\sigma_k\sigma_\ell}\big\langle A\boldsymbol{v}_k, A\boldsymbol{v}_\ell\big\rangle = \frac{1}{\sigma_k\sigma_\ell}\big\langle \boldsymbol{v}_k, A^\dagger A\boldsymbol{v}_\ell\big\rangle = \frac{\lambda_\ell}{\sigma_k\sigma_\ell}\big\langle \boldsymbol{v}_k, \boldsymbol{v}_\ell\big\rangle = \frac{\sigma_\ell}{\sigma_k}\delta_{k\ell}\,.$$

3. $\{\boldsymbol{u}_1, \cdots, \boldsymbol{u}_r\}$ are eigenvectors of $AA^\dagger$ withe corresponding eigenvalues $\lambda_1, \cdots, \lambda_r$ since for $1 \leqslant j \leqslant r$,

   $$AA^\dagger \boldsymbol{u}_j = AA^\dagger\Big(\frac{1}{\sigma_j} A\boldsymbol{v}_j\Big) = \frac{1}{\sigma_j} AA^\dagger A\boldsymbol{v}_j = \frac{1}{\sigma_j} A(\lambda_j \boldsymbol{v}_j) = \lambda_j \frac{1}{\sigma_j} A\boldsymbol{v}_j = \lambda_j \boldsymbol{u}_j\,.$$

   By the fact that $r = \operatorname{rank}(A^\dagger A) = \operatorname{rank}(A) = \operatorname{rank}(A^\dagger) = \operatorname{rank}(AA^\dagger)$, the nullity (that is, the dimension of the null space) of $AA^\dagger$ is $m - r$; thus there exist an orthonormal set $\{\boldsymbol{u}_{r+1}, \cdots, \boldsymbol{u}_m\}$ in the null space of $AA^\dagger$. Then

   $$AA^\dagger \boldsymbol{u}_j = \sigma_j^2 \boldsymbol{u}_j \qquad \forall\, 1 \leqslant j \leqslant m\,.$$

   Since $\{\boldsymbol{u}_{r+1}, \cdots, \boldsymbol{u}_m\}$ are eigenvectors of $AA^\dagger$ (corresponding to eigenvalue 0), we find that $\{\boldsymbol{u}_1, \cdots, \boldsymbol{u}_m\}$ is an orthonormal basis of $\mathbb{C}^m$.

Let $U = \big[\boldsymbol{u}_1 \,\vdots\, \boldsymbol{u}_2 \,\vdots\, \cdots \,\vdots\, \boldsymbol{u}_m\big]$ and $V = \big[\boldsymbol{v}_1 \,\vdots\, \boldsymbol{v}_2 \,\vdots\, \cdots \,\vdots\, \boldsymbol{v}_n\big]$, as well as

$$\Sigma = \begin{bmatrix} \sigma_1 & & & & & \\ & \sigma_2 & & & & \\ & & \ddots & & & \\ & & & \sigma_r & & \\ & & & & 0 & \\ & & & & & \ddots \end{bmatrix}.$$

Then

$$\begin{aligned} AV &= A\big[\boldsymbol{v}_1 \,\vdots\, \boldsymbol{v}_2 \,\vdots\, \cdots \,\vdots\, \boldsymbol{v}_n\big] = \big[A\boldsymbol{v}_1 \,\vdots\, A\boldsymbol{v}_2 \,\vdots\, \cdots \,\vdots\, A\boldsymbol{v}_n\big] \\ &= \big[\sigma_1\boldsymbol{u}_1 \,\vdots\, \sigma_2\boldsymbol{u}_2 \,\vdots\, \cdots \,\vdots\, \sigma_r\boldsymbol{u}_r \,\vdots\, \boldsymbol{0} \,\vdots\, \cdots \,\vdots\, \boldsymbol{0}\big] \\ &= \big[\boldsymbol{u}_1 \,\vdots\, \boldsymbol{u}_2 \,\vdots\, \cdots \,\vdots\, \boldsymbol{u}_m\big] \begin{bmatrix} \sigma_1 & & & & & \\ & \sigma_2 & & & & \\ & & \ddots & & & \\ & & & \sigma_r & & \\ & & & & 0 & \\ & & & & & \ddots \end{bmatrix} = U\Sigma\,. \end{aligned}$$

The numbers $\sigma_1, \sigma_2, \cdots, \sigma_n$ are called the ***singular values*** of $A$. The fact that $U$ and $V$ are unitary shows the following

**Theorem 3.116.** *Let $A$ be a complex $m \times n$ matrix. Then there exist unitary matrices $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$ as well as an $m \times n$ matrix $\Sigma$ of the form*

$$\Sigma = \begin{bmatrix} \sigma_1 & & & & & \\ & \sigma_2 & & & & \\ & & \ddots & & & \\ & & & \sigma_r & & \\ & & & & 0 & \\ & & & & & \ddots \end{bmatrix},$$

*where $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r > 0$, such that $A = U\Sigma V^\dagger$.*

**Remark 3.117.** The decomposition $A = U\Sigma V^\dagger$ in Theorem 3.116 is called a ***singular value decomposition*** of $A$. We note that since $\{\boldsymbol{u}_{r+1}, \cdots, \boldsymbol{u}_m\}$ is a chosen orthonormal basis of the null space of $AA^\dagger$, the singular decomposition of $A$ is not unique.

### 3.8.3 The CS decomposition

**Theorem 3.118.** *For any $2 \times 2$ partitioning*

$$Q = \begin{array}{c} \begin{matrix} c_1 & c_2 \end{matrix} \\ \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \end{array} \begin{matrix} r_1 \\ r_2 \end{matrix} \qquad n = c_1 + c_2 = r_1 + r_2\,, \tag{3.23}$$

*of an $n \times n$ unitary matrix $Q$, there exist unitary matrices $U_1$, $U_2$, $V_1$, $V_2$ such that*

$$U^\dagger Q V = \begin{bmatrix} U_1^\dagger & 0 \\ 0 & U_2^\dagger \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix} = \begin{bmatrix} \mathrm{I} & & & \mathrm{O}_s^\dagger & & \\ & \mathrm{C} & & & -\mathrm{S} & \\ & & \mathrm{O}_c & & & -\mathrm{I} \\ \mathrm{O}_s & & & \mathrm{I} & & \\ & \mathrm{S} & & & \mathrm{C} & \\ & & \mathrm{I} & & & \mathrm{O}_c^\dagger \end{bmatrix},$$

*where* C *and* S *are diagonal matrices taking the form*

$$\mathrm{C} = \mathrm{diag}(\gamma_1, \gamma_2, \cdots, \gamma_s)\,, \qquad 1 > \gamma_1 \geqslant \gamma_2 \geqslant \cdots \geqslant \gamma_s > 0\,, \tag{3.24a}$$

$$\mathrm{S} = \mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_s)\,, \qquad 0 < \sigma_1 \leqslant \sigma_2 \leqslant \cdots \leqslant \sigma_s < 1 \tag{3.24b}$$

*and satisfying $\mathrm{C}^2 + \mathrm{S}^2 = \mathrm{I}$, and $\mathrm{O}_s$, $\mathrm{O}_c$ are matrices of zeros, and depending on $Q$ and the partition, may have no row or no columns. Some of the identity matrices may be nonexistent, and no two of them need be equal. The four* C *and* S *matrices are square with the same dimension, and may be nonexistent.*

*Proof.* Choose unitary matrices $U_1$ and $V_1$ to give the usual singular value decomposition of $Q_{11}$, resulting in $D_{11}$. Choose unitary matrices $U_2$ and $V_2$ so that $D_{21} = U_2^\dagger Q_{21} V_1$ is lower triangular with non-negative real entries on the diagonals ending in the bottom right corners and $D_{12} = U_1^\dagger Q_{12} V_2$ is upper triangular with non-positive real entries on the diagonals ending in the bottom right corners. Define

$$D = \begin{bmatrix} U_1^\dagger & 0 \\ 0 & U_2^\dagger \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}. \tag{3.25}$$

Then $D$ is unitary; thus the fact that any column (or row) of $D$ has unit length implies that no singular value of $D_{11}$ can exceed 1. Therefore, $D_{11}$ takes the form

$$D_{11} = \begin{bmatrix} \mathrm{I}_{k\times k} & & \\ & \mathrm{C}_{s\times s} & \\ & & \mathrm{O}_{p\times q} \end{bmatrix}$$

for some C taking the form (3.24a), and the orthogonality of columns of $D$ and the orthogonality of rows of $D$ further show that $D_{21}$ and $D_{12}$ must take the form

$$D_{12} = \begin{bmatrix} \mathrm{O}_{k\times(c_2-s-p)} & & \\ & -\mathrm{S}_{s\times s} & \\ & & -\mathrm{I}_{p\times p} \end{bmatrix}, \qquad D_{21} = \begin{bmatrix} \mathrm{O}_{(c_1-s-q)\times k} & & \\ & \mathrm{S}_{s\times s} & \\ & & \mathrm{I}_{q\times q} \end{bmatrix}, \quad (3.26)$$

where $p = r_1 - k - s$ and $q = c_1 - k - s$. The fact that each column and each row of $D$ has unit length also gives the form of $D_{22}$ so that

$$D = \begin{bmatrix} \mathrm{I} & & & \mathrm{O}_s^\dagger & & \\ & \mathrm{C} & & & -\mathrm{S} & \\ & & \mathrm{O}_c & & & -\mathrm{I} \\ \mathrm{O}_s & & & \mathrm{K} & \mathrm{L} & \\ & \mathrm{S} & & \mathrm{M} & \mathrm{N} & \\ & & \mathrm{I} & & & \mathrm{O}_c^\dagger \end{bmatrix}$$

for some $(r_2-s-q)\times(c_2-s-p)$ matrix K, $(r_2-s-q)\times s$ matrix L, $s\times(c_2-s-p)$ matrix M and $s\times s$ matrix N. The orthogonality of the second and the fourth blocks of columns shows that $\mathrm{SM} = \mathrm{O}_{s\times(c_2-s-p)}$; thus $\mathrm{M} = \mathrm{O}_{s\times(c_2-s-p)}$ since S is non-singular. Similarly, the orthogonality of the second and the fourth blocks of rows shows that $\mathrm{L} = \mathrm{O}_{(r_2-s-q)\times s}$. Next, from the fifth and the second blocks of rows, $\mathrm{SC} - \mathrm{NS} = \mathrm{O}_{s\times s}$, so $\mathrm{N} = \mathrm{C}$ and we obtain that

$$D = \begin{bmatrix} \mathrm{I} & & & \mathrm{O}_s^\dagger & & \\ & \mathrm{C} & & & -\mathrm{S} & \\ & & \mathrm{O}_c & & & -\mathrm{I} \\ \mathrm{O}_s & & & \mathrm{K} & & \\ & \mathrm{S} & & & \mathrm{C} & \\ & & \mathrm{I} & & & \mathrm{O}_c^\dagger \end{bmatrix}$$

Finally, note that $r_2 - s - q = r_2 + k - c_1 = c_2 + k - r_1 = c_2 - s - p$ so that K is a square matrix. Together with the fact that $D^\dagger D = DD^\dagger = \mathrm{I}$, we find that $\mathrm{KK}^\dagger = \mathrm{K}^\dagger\mathrm{K} = \mathrm{I}$ so that K is unitary and can be transformed to I without altering the rest of $D$ by replacing $U_2$ with $U_2\,\mathrm{blkdiag}(K^\dagger, \mathrm{I}_{s\times s}, \mathrm{I}_{q\times q})$ in (3.25). □

**Exercise 3.119.** Write a function named CSD in matlab® in the format

$$[\mathrm{U,D,V}] = \mathrm{CSD(Q)}$$

which outputs the CS decomposition of a unitary matrix $Q$ (so that $D \equiv U^\dagger QV$ takes the form $\begin{bmatrix} \mathrm{C} & -\mathrm{S} \\ \mathrm{S} & \mathrm{C} \end{bmatrix}$ with C and S satisfying (3.24)).

**Remark 3.120.** Suppose that $Q$ is an $n$-qubit quantum gate (that is, $Q$ is an $2^n \times 2^n$ unitary matrix). By partitioning $Q$ into $2 \times 2$ subblocks with equal size (that is, $r_1 = r_2 = c_1 = c_2 = 2^{n-1}$), Theorem 3.118 implies that there exist $2^{n-1} \times 2^{n-1}$ unitary matrices $U_1$, $U_2$, $V_1$, $V_2$ (so that they are $(n-1)$-qubit gates) such that

$$Q = \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} \begin{bmatrix} \mathrm{C} & -\mathrm{S} \\ \mathrm{S} & \mathrm{C} \end{bmatrix} \begin{bmatrix} V_1^\dagger & 0 \\ 0 & V_2^\dagger \end{bmatrix}$$

for some diagonal matrices C and S of the form

$$\mathrm{C} = \begin{bmatrix} \cos\theta_1 & & \\ & \ddots & \\ & & \cos\theta_{2^{n-1}} \end{bmatrix} \quad \text{and} \quad \mathrm{S} = \begin{bmatrix} \sin\theta_1 & & \\ & \ddots & \\ & & \sin\theta_{2^{n-1}} \end{bmatrix},$$

where $0 \leqslant \theta_1 \leqslant \theta_2 \leqslant \cdots \leqslant \theta_{2^{n-1}} \leqslant \dfrac{\pi}{2}$. In terms of quantum circuits, the case $n = 3$ can be illustrated as follows:

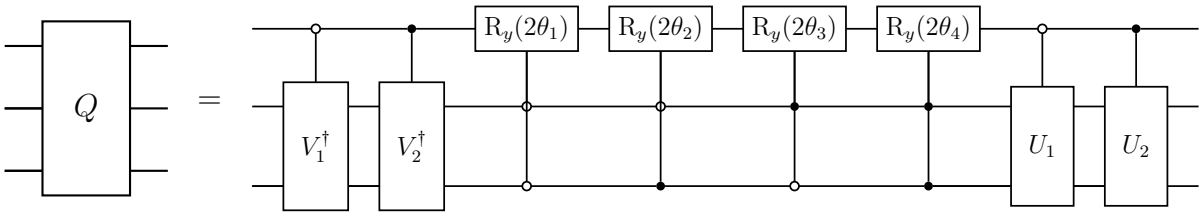
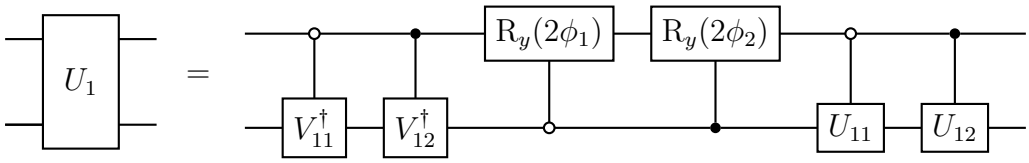



Figure 3.2: The CS decomposition in terms of quantum circuits

The 2-qubit gates $U_1$, $U_2$, $V_1^\dagger$ and $V_2^\dagger$ can be further decomposed. For example,




for some $0 \leqslant \phi_1 \leqslant \phi_2 \leqslant \dfrac{\pi}{2}$ and quantum gates with matrix representations $V_{11}, V_{12}, U_{11}, U_{12}$ so that

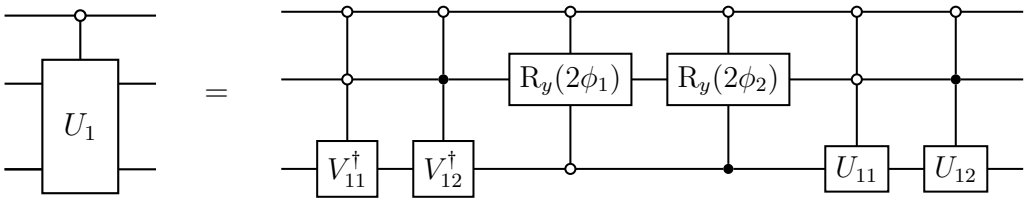



Figure 3.3: The decomposition of the controlled $V_1^\dagger$ gate

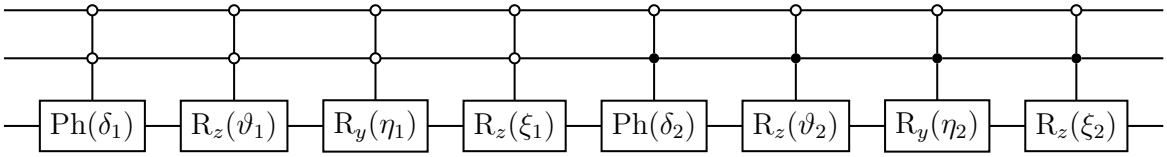
Combining all these quantum gates together, we see that the CS decomposition essentially provides a way to express an $n$-qubit gate as the product of multi-controlled gates.

### 3.8.4 Decomposition of arbitrary quantum gates

By Theorem 3.112, any 1-qubit gate can be decomposed further as the product of rotation gates $\mathrm{R}_y$, $\mathrm{R}_z$ and phase gate Ph. Therefore, if the quantum gates $V_{11}^\dagger$ and $V_{12}^\dagger$ in Figure 3.3 can be expressed as

$$V_{11}^\dagger = \mathrm{R}_z(\xi_1)\mathrm{R}_y(\eta_1)\mathrm{R}_z(\vartheta_1)\mathrm{Ph}(\delta_1)\,, \qquad V_{12}^\dagger = \mathrm{R}_z(\xi_2)\mathrm{R}_y(\eta_2)\mathrm{R}_z(\vartheta_2)\mathrm{Ph}(\delta_2)\,,$$

then the first two controlled $V^\dagger$ gates can be further decomposed into




and so on. Without any further modification, we can express an $n$-qubit gate as the product of multi-controlled ***rotation*** gates, at the expense of some not implementable phase gates. In this section, we talk about how to "cancel out" these phase gates and make an $n$-qubit gate indeed the product of multi-controlled rotation gates.

Before proceeding, we note that if $P$ is a $2^{n-1} \times 2^{n-1}$ diagonal unitary matrix; that is, $P$ takes the form

$$P = \mathrm{diag}\big(e^{i\alpha_1}, \cdots, e^{i\alpha_{2^{n-1}}}\big) \qquad \text{for some } \alpha_1, \cdots, \alpha_{2^{n-1}} \in \mathbb{R}\,,$$

then

$$\begin{bmatrix} P & 0 \\ 0 & P \end{bmatrix} \begin{bmatrix} \mathrm{C} & -\mathrm{S} \\ \mathrm{S} & \mathrm{C} \end{bmatrix} = \begin{bmatrix} \mathrm{C} & -\mathrm{S} \\ \mathrm{S} & \mathrm{C} \end{bmatrix} \begin{bmatrix} P & 0 \\ 0 & P \end{bmatrix}.$$

Therefore, if $P$ is a $2^{n-1} \times 2^{n-1}$ diagonal unitary matrix, then $P^\dagger$ is also a $2^{n-1} \times 2^{n-1}$ diagonal unitary matrix so that the decomposition above implies that

$$\begin{aligned} Q &= \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} \begin{bmatrix} P & 0 \\ 0 & P \end{bmatrix} \begin{bmatrix} P^\dagger & 0 \\ 0 & P^\dagger \end{bmatrix} \begin{bmatrix} \mathrm{C} & -\mathrm{S} \\ \mathrm{S} & \mathrm{C} \end{bmatrix} \begin{bmatrix} V_1^\dagger & 0 \\ 0 & V_2^\dagger \end{bmatrix} \\ &= \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} \begin{bmatrix} P & 0 \\ 0 & P \end{bmatrix} \begin{bmatrix} \mathrm{C} & -\mathrm{S} \\ \mathrm{S} & \mathrm{C} \end{bmatrix} \begin{bmatrix} P^\dagger & 0 \\ 0 & P^\dagger \end{bmatrix} \begin{bmatrix} V_1^\dagger & 0 \\ 0 & V_2^\dagger \end{bmatrix}. \end{aligned}$$

The diagonal unitary matrix $P$ will be chosen to "cancel out the phase gate" so that the matrix $\begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} \begin{bmatrix} P & 0 \\ 0 & P \end{bmatrix}$ is a product of multi-controlled **rotation** gates.

Let $U$ be a $n$-qubit quantum gate. By Remark 3.120 $U$ can be decomposed as

$$U = \begin{bmatrix} U_{11}^1 & 0 \\ 0 & U_{12}^1 \end{bmatrix} \begin{bmatrix} P_{11}^1 & 0 \\ 0 & P_{11}^1 \end{bmatrix} \begin{bmatrix} C_{11}^1 & -S_{11}^1 \\ S_{11}^1 & C_{11}^1 \end{bmatrix} \begin{bmatrix} P_{11}^{1\dagger} & 0 \\ 0 & P_{11}^{1\dagger} \end{bmatrix} \begin{bmatrix} V_{21}^{1\dagger} & 0 \\ 0 & V_{22}^{1\dagger} \end{bmatrix},$$

where $P_{11}^1$ is an arbitrary $2^{n-1} \times 2^{n-1}$ diagonal unitary matrix. Write $P_{11}^{1\dagger} V_{21}^{1\dagger} = U_{21}^1$ and $P_{11}^{1\dagger} V_{22}^{1\dagger} = U_{22}^1$. Then

$$U = \begin{bmatrix} U_{11}^1 & 0 \\ 0 & U_{12}^1 \end{bmatrix} \begin{bmatrix} P_{11}^1 & 0 \\ 0 & P_{11}^1 \end{bmatrix} \begin{bmatrix} C_{11}^1 & -S_{11}^1 \\ S_{11}^1 & C_{11}^1 \end{bmatrix} \begin{bmatrix} U_{21}^1 & 0 \\ 0 & U_{22}^1 \end{bmatrix},$$

and the decomposition can be applied recursively to the sub-matrices $U_{jk}^i$ until a $2 \times 2$ block-diagonal form is encountered. For example, we use the CS decomposition to write

$$U_{11}^1 = U_{111}^2 P_{111}^2 A_{11}^2 P_{111}^{2\dagger} V_{112}^{2\dagger}, \qquad U_{12}^1 = U_{121}^2 P_{121}^2 A_{12}^2 P_{121}^{2\dagger} V_{122}^{2\dagger},$$

so that by defining $U_{112}^2 = P_{111}^{2\dagger} V_{112}^{2\dagger}$ and $U_{122}^2 = P_{121}^{2\dagger} V_{122}^{2\dagger}$,

$$\begin{aligned} \begin{bmatrix} U_{11}^1 & 0 \\ 0 & U_{12}^1 \end{bmatrix} &= \begin{bmatrix} U_{111}^2 P_{111}^2 A_{11}^2 U_{112}^2 & 0 \\ 0 & U_{121}^2 P_{121}^2 A_{12}^2 U_{122}^2 \end{bmatrix} \\ &= \begin{bmatrix} U_{111}^2 & 0 \\ 0 & U_{121}^2 \end{bmatrix} \begin{bmatrix} P_{111}^2 & 0 \\ 0 & P_{121}^2 \end{bmatrix} \begin{bmatrix} A_{11}^2 & 0 \\ 0 & A_{12}^2 \end{bmatrix} \begin{bmatrix} U_{112}^2 & 0 \\ 0 & U_{122}^2 \end{bmatrix}. \end{aligned}$$

We note that in principle we need to specify $P_{11}^1$ first before we can decompose $U_{21}^1$ and $U_{22}^1$ further since $U_{21}^1$ and $U_{22}^1$ depend on $P_{11}^1$.

In general, with $U_1^0$ denoting $U$, for $1 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant 2^{i-1}$, we use the CS decomposition on each block of $U_j^{i-1}$ to write

$$U_j^{i-1} = U_{2j-1}^i P_{2j-1}^i A_{2j-1}^i P_{2j-1}^{i\dagger} V_{2j}^{i\dagger},$$

where $U^i_{2j-1}$ and $U^i_{2j} \equiv P^{i\dagger}_{2j-1}V^{i\dagger}_{2j}$ are block diagonal matrices consisting of $2^i$ blocks of $2^{n-i}\times 2^{n-i}$ unitary matrices, $P^i_{2j-1}$ is a block diagonal matrix of the form

$$P^i_{2j-1} = \mathrm{blkdiag}\big(Q^i_1, Q^i_1, Q^i_2, Q^i_2, \cdots, Q^i_{2^{i-1}}, Q^i_{2^{i-1}}\big)$$

for some $2^{n-i}\times 2^{n-i}$ diagonal unitary matrices $Q^i_1$, $\cdots$, $Q^i_{2^{i-1}}$ to be determined. We also note that $U^i_{2j}$ depends on $P^i_{2j-1}$.

Define $P^i_{2j} = P^{i-1}_j$ and $A^i_{2j} = A^{i-1}_j$. We then have the following sequence of decomposition

$$\begin{aligned}
U &= U^1_1P^1_1A^1_1U^1_2 = U^2_1P^2_1A^2_1U^2_2P^2_2A^2_2U^2_3P^2_3A^2_3U^2_4 = U^2_1P^2_1A^2_1U^2_2P^2_2A^2_2U^2_3P^2_3A^2_3U^2_4\\
&= U^3_1P^3_1A^3_1U^3_2P^3_2A^3_2U^3_3P^3_3A^3_3U^3_4P^3_4A^3_4U^3_5P^3_5A^3_5U^3_6P^3_6A^3_6U^3_7P^3_7A^3_7U^3_8 = \cdots\\
&= U^{n-1}_1P^{n-1}_1A^{n-1}_1U^{n-1}_2P^{n-1}_2A^{n-1}_2U^{n-1}_3P^{n-1}_3A^{n-1}_3\cdots U^{n-1}_{2^{n-1}-1}P^{n-1}_{2^{n-1}-1}A^{n-1}_{2^{n-1}-1}U^{n-1}_{2^{n-1}}\\
&= \Big(\prod_{j=1}^{2^{n-1}-1} U^{n-1}_jP^{n-1}_jA^{n-1}_j\Big)U^{n-1}_{2^{n-1}}\,. \qquad (3.27)
\end{aligned}$$

Here the upper index denotes the level of recursion, whereas the lower index denotes the position of the matrix within the resulting matrix product. In (3.27),

1. $U^{n-1}_j$ takes the form
$$U^{n-1}_j = \mathrm{blkdiag}(U_1, U_2, \cdots, U_{2^{n-1}})$$
for some $2\times 2$ unitary matrices $U_1, \cdots, U_{2^{n-1}}$.

2. For each $j\in\mathbb{N}$, let the number $\gamma(j)$ indicate the position (counting from the lowest bit) of the right-most non-zero bit in the binary presentation of the number $j$. In other words, for each $j\in\mathbb{N}$, $\gamma(j)$ is the unique integer satisfying
$$j = 2^{\gamma(j)-1}(2k-1) \quad \text{for some } k\in\mathbb{N}\,.$$
Then $P^{n-1}_j = P^{n-1}_{2^{\gamma(j)-1}(2k-1)} = P^{n-\gamma(j)}_{2k-1}$ and $A^{n-1}_j = A^{n-1}_{2^{\gamma(j)-1}(2k-1)} = A^{n-\gamma(j)}_{2k-1}$ which imply that $P^{n-1}_j$ and $A^{n-1}_j$ appear first time in the $(n-\gamma(j))$-th recursion of decompositions and do not appear in any previous recursion of decompositions. Therefore, $P^{n-1}_j$ takes the form
$$P^{n-1}_j = \mathrm{blkdiag}\big(Q_1, Q_1, \cdots, Q_{2^{n-\gamma(j)-1}}, Q_{2^{n-\gamma(j)-1}}\big)$$
for some $2^{\gamma(j)}\times 2^{\gamma(j)}$ diagonal unitary matrices $Q_1, \cdots, Q_{2^{n-\gamma(j)-1}}$, and $A^{n-1}_j$ takes the form
$$A^{n-1}_j = \mathrm{blkdiag}\left(\begin{bmatrix} C_1 & -S_1\\ S_1 & C_1\end{bmatrix}, \begin{bmatrix} C_2 & -S_2\\ S_2 & C_2\end{bmatrix}, \cdots, \begin{bmatrix} C_{2^{n-\gamma(j)-1}} & -S_{2^{n-\gamma(j)-1}}\\ S_{2^{n-\gamma(j)-1}} & C_{2^{n-\gamma(j)-1}}\end{bmatrix}\right), \qquad (3.28)$$

where for each $1 \leqslant k \leqslant 2^{n-\gamma(j)-1}$,

$$C_k = \operatorname{diag}(\cos\theta_1^k, \cdots, \cos\theta_{2^{\gamma(j)}}^k) \quad \text{and} \quad S_k = \operatorname{diag}(\sin\theta_1^k, \cdots, \sin\theta_{2^{\gamma(j)}}^k)$$

for some $0 \leqslant \theta_1^k \leqslant \theta_2^k \leqslant \cdots \leqslant \theta_{2^{\gamma(j)}}^k \leqslant \frac{\pi}{2}$. We note that $A_j^{n-1}$ is indeed a multi-controlled gate of type $F^n_{n-\gamma(j)}(\mathrm{R}_y)$.

3. $P_j^{n-1}$ can be chosen according to $U_j^{n-1}$ so that $U_j^{n-1}P_j^{n-1}$ is a product of multi-controlled rotation gates (which will be explained soon). On the other hand, for each $1 \leqslant j \leqslant 2^n - 1$ the block diagonal matrix $U_{j+1}^{n-1}$ depends on $P_k^{n-1}$ for all $1 \leqslant k \leqslant j$; thus we need to specify $P_1^{n-1}$, $P_2^{n-1}$, $\cdots$ successively in order to complete the decomposition.

**Remark 3.121.** In matlab®, $\gamma$ can be implemented by

$$\gamma(j) = \textbf{min}(\textbf{find}(\textbf{de2bi}(j) == 1)).$$

Now we determine $P_1^{n-1}$. Since $U_1^{n-1}$ is a block diagonal matrix consisting of $2^{n-1}$ blocks of $2 \times 2$ unitary matrices $U_{11}^{n-1}, \cdots, U_{12^{n-1}}^{n-1}$; that is,

$$U_1^{n-1} = \begin{bmatrix} U_{11}^{n-1} & & & \\ & U_{12}^{n-1} & & \\ & & \ddots & \\ & & & U_{12^{n-1}}^{n-1} \end{bmatrix},$$

by Theorem 3.112 for each $1 \leqslant j \leqslant 2^{n-1}$ there exist $\delta_j, \xi_j, \theta_j, \eta_j$ such that

$$U_{1j}^{n-1} = \mathrm{R}_z(\xi_j)\mathrm{R}_y(\theta_j)\mathrm{R}_z(\eta_j)\mathrm{Ph}(\delta_j)\,;$$

thus

$$U_1^{n-1} = \begin{bmatrix} \mathrm{R}_z(\xi_1) & & \\ & \ddots & \\ & & \mathrm{R}_z(\xi_{2^{n-1}}) \end{bmatrix} \begin{bmatrix} \mathrm{R}_y(\theta_1) & & \\ & \ddots & \\ & & \mathrm{R}_y(\theta_{2^{n-1}}) \end{bmatrix} \begin{bmatrix} \mathrm{R}_z(\eta_1) & & \\ & \ddots & \\ & & \mathrm{R}_z(\eta_{2^{n-1}}) \end{bmatrix} \begin{bmatrix} \mathrm{Ph}(\delta_1) & & \\ & \ddots & \\ & & \mathrm{Ph}(\delta_{2^{n-1}}) \end{bmatrix}.$$

For each $1 \leqslant j \leqslant 2^{n-2}$, let $\alpha_j = -\dfrac{\delta_{2j-1} + \delta_{2j}}{2}$. Define $Q_j = \operatorname{diag}(e^{i\alpha_j}, e^{i\alpha_j})$ and $\beta_j = \delta_{2j} - \delta_{2j-1}$. Then

$$\mathrm{Ph}(\delta_{2j-1})Q_j = \operatorname{diag}\big(e^{-i\beta_j/2}, e^{-i\beta_j/2}\big) \qquad \text{and} \qquad \mathrm{Ph}(\delta_{2j})Q_j = \operatorname{diag}\big(e^{i\beta_j/2}, e^{i\beta_j/2}\big)$$

so that

$$\mathrm{blkdiag}\big(\mathrm{Ph}(\delta_{2j-1}),\mathrm{Ph}(\delta_{2j})\big)\cdot\mathrm{blkdiag}(Q_j,Q_j)=\mathrm{diag}(e^{-i\beta_j/2},e^{-i\beta_j/2},e^{i\beta_j/2},e^{i\beta_j/2})\,.$$

Therefore, by defining $P_1^{n-1}=\mathrm{blkdiag}\big(Q_1,Q_1,Q_2,Q_2,\cdots,Q_{2^{n-1}},Q_{2^{n-1}}\big)$ we have

$$\begin{aligned}&\mathrm{blkdiag}\big(\mathrm{Ph}(\delta_1),\cdots,\mathrm{Ph}(\delta_{2^{n-1}})\big)P_1^{n-1}\\&\quad=\mathrm{diag}\big(e^{-i\beta_1/2},e^{-i\beta_1/2},e^{i\beta_1/2},e^{i\beta_1/2},e^{-i\beta_2/2},e^{-i\beta_2/2},e^{i\beta_2/2},e^{i\beta_2/2},\cdots\\&\qquad\cdots,e^{-i\beta_{2^{n-2}}/2},e^{-i\beta_{2^{n-2}}/2},e^{i\beta_{2^{n-2}}/2},e^{i\beta_{2^{n-2}}/2}\big)\,.\end{aligned}$$

which, by Example 3.109, is a multi-controlled gate of type $F_n^n(\mathrm{R}_z)$ (with $\theta_{2j-1}=\theta_{2j}$ for all $1\leqslant j\leqslant 2^{n-1}$). In other words, by multiplying $P_1^{n-1}$ on the right-hand side of the block diagonal matrix generated by the phases of each $2\times 2$ unitary matrix, we obtain a multi-controlled gate whose target qubit is the $(n-1)$-th qubit. This shows that $U_1^{n-1}P_1^{n-1}$ is a product of multi-controlled gates in which the rotation gates involved are $\mathrm{R}_y$ and $\mathrm{R}_z$.

Suppose that $P_1^{n-1},\cdots,P_{j-1}^{n-1}$ are specified so that $U_2^{n-1},\cdots,U_j^{n-1}$ are determined accordingly. Since $U_j^{n-1}$ is also a block diagonal matrix consisting of $2^{n-1}$ blocks of $2\times 2$ unitary matrices $U_{j1}^{n-1}$, $\cdots$, $U_{j2^{n-1}}^{n-1}$, by Theorem 3.112 we can decompose $U_j^{n-1}$ as

$$\begin{aligned}U_j^{n-1}=\mathrm{blkdiag}\big(\mathrm{R}_z(\xi_1),\cdots,\mathrm{R}_z(\xi_{2^{n-1}})\big)\cdot\mathrm{blkdiag}(\mathrm{R}_y(\theta_1),\cdots,\mathrm{R}_y(\theta_{2^{n-1}}))\cdot\\\cdot\,\mathrm{blkdiag}\big(\mathrm{R}_z(\eta_1),\cdots,\mathrm{R}_z(\eta_{2^{n-1}})\big)\cdot\mathrm{blkdiag}\big(\mathrm{Ph}(\delta_1),\cdots,\mathrm{Ph}(\delta_{2^{n-1}})\big)\end{aligned}$$

for some $\xi_1,\cdots,\xi_{2^{n-1}}$, $\theta_1,\cdots,\theta_{2^{n-1}}$, $\eta_1,\cdots,\eta_{2^{n-1}}$ and $\delta_1,\cdots,\delta_{2^{n-1}}$. We note that these $\xi_j$'s, $\theta_j$'s, $\eta_j$'s and $\delta_j$'s are in principle different from those values used in the decomposition of $U_1^{n-1}$, $\cdots$, $U_{j-1}^{n-1}$.

For $1\leqslant k\leqslant n-\gamma(j)-1$ and $1\leqslant\ell\leqslant 2^{\gamma(j)}$, define

$$\alpha_{(k-1)2^{\gamma(j)}+\ell}=-\frac{1}{2}\Big(\delta_{(k-1)2^{\gamma(j)}+[\frac{\ell+1}{2}]}+\delta_{(k-1)2^{\gamma(j)}+2^{\gamma(j)-1}+[\frac{\ell+1}{2}]}\Big)\,,$$

where $\big[\frac{\ell+1}{2}\big]$ in the sub-index denotes the largest integer which is not greater than $\frac{\ell+1}{2}$. Let

$$Q_k=\mathrm{diag}\big(e^{i\alpha_{2^{(k-1)\gamma(j)}+1}},\cdots,e^{i\alpha_{2^k\gamma(j)}}\big)$$

and

$$P_j^{n-1}=\mathrm{blkdiag}\big(Q_1,Q_1,Q_2,Q_2,\cdots,Q_{2^{n-\gamma(j)-1}},Q_{2^{n-\gamma(j)-1}}\big)\,,$$

we find that $\mathrm{blkdiag}\big(\mathrm{Ph}(\delta_1),\cdots,\mathrm{Ph}(\delta_{2^{n-1}})\big)P_j^{n-1}$, with $N$ denoting $2^{\gamma(j)}$, takes the form

$$\begin{aligned}&\mathrm{diag}(e^{-i\beta_1/2},e^{-i\beta_2/2},\cdots,e^{-i\beta_N/2},e^{i\beta_1/2},e^{i\beta_2/2},\cdots,e^{i\beta_N/2},\\&\quad e^{-i\beta_{N+1}/2},e^{-\beta_{N+2}/2},\cdots,e^{-i\beta_{2N}/2},e^{i\beta_{N+1}/2},e^{\beta_{N+2}/2},\cdots,e^{i\beta_{2N}/2},\\&\quad\cdots,e^{-i\beta_{2^{n-1}-N+1}/2},e^{-\beta_{2^{n-1}-N+2}/2},\cdots,e^{-i\beta_{2^{n-1}}/2},e^{i\beta_{2^{n-1}-N+1}/2},e^{\beta_{2^{n-1}-N+2}/2},\cdots,e^{i\beta_{2^{n-1}}/2}\big)\end{aligned}$$

for some $\beta_1,\cdots,\beta_{2^{n-1}}\in\mathbb{R}$. By Example 3.109, it is a multi-controlled gate of type $F^n_{n-\gamma(j)}(\mathrm{R}_z)$ whose target qubit is the $(n-\gamma(j))$-th qubit. Therefore, $U_j^{n-1}P_j^{n-1}$ is the product of multi-controlled gates in which the rotation gates involved are $\mathrm{R}_y$ and $\mathrm{R}_z$.

Let $U$ be an $n$-qubit gate (or equivalently, $2^n\times 2^n$ unitary matrix). Using (3.27),

$$U=\Big(\prod_{j=1}^{2^{n-1}-1}U_j^{n-1}P_j^{n-1}A_j^{n-1}\Big)U_{2^{n-1}}^{n-1},$$

where $U_j^{n-1}$ is a block diagonal of $2\times 2$ matrix for all $j$, and $A_j^i$ takes the form (3.28). From the argument above, we know that $U_j^{n-1}P_j^{n-1}$ is the product of multi-controlled gates, while each $A_j^{n-1}$ is a multi-controlled gate of type $F^n_{n-\gamma(j)}(\mathrm{R}_y)$. Therefore, in order to implement the quantum gate with matrix representation $U$ using quantum circuits, it suffices to consider how to implement a multi-controlled gate in which the rotation gate involved is $\mathrm{R}_y$ or $\mathrm{R}_z$.

**Theorem 3.122.** *Each $2^{n+1}\times 2^{n+1}$ unitary matrix can be expressed as the product of multi-controlled rotation gates of type $F_k^n(\mathrm{R}_y)$ and $F_k^n(\mathrm{R}_z)$, $k=1,2,\cdots,n+1$.*

## 3.9 Implementation of Multi-Controlled Rotation Gates

In this section we are concerned with the implementation of multi-controlled rotation gates of type $F^n_{n+1}(R_{\boldsymbol{a}})$ with unit vector $\boldsymbol{a}=(0,a_y,a_z)$ using quantum circuits. The implementation of multi-controlled gate of this type is the building block of the implementation of general quantum gates. We note that multi-controlled rotation gate of type $F_k^n(R_{\boldsymbol{a}})$, where $1\leqslant k\leqslant n$, can be obtained by applying several swap operations on multi-controlled rotation gate of type $F^n_{n+1}(R_{\boldsymbol{a}})$; thus arbitrary multi-controlled rotation gates can also be implemented even though we only focus on the case of $F^n_{n+1}(R_{\boldsymbol{a}})$.

Recall that Example 3.106 shows that the matrix representation of multi-controlled

rotation gates of type $F^n_{n+1}(R_{\boldsymbol{a}})$ takes the form

$$R = \text{blkdiag}\big(R_{\boldsymbol{a}}(\phi_1), \cdots, R_{\boldsymbol{a}}(\phi_{2^n})\big) = \begin{bmatrix} R_{\boldsymbol{a}}(\phi_1) & & & \\ & R_{\boldsymbol{a}}(\phi_2) & & \\ & & \ddots & \\ & & & R_{\boldsymbol{a}}(\phi_{2^n}) \end{bmatrix}, \tag{3.29}$$

where for a given unit vector $\boldsymbol{a} = (a_x, a_y, a_z)$ and angle $\phi$, the rotation matrix $R_{\boldsymbol{a}}(\phi)$ is given by (2.8) or equivalently,

$$R_{\boldsymbol{a}}(\phi) = \mathrm{I}\cos\frac{\phi}{2} + i(a_x\sigma_x + a_y\sigma_y + a_z\sigma_z)\sin\frac{\phi}{2},$$

in which $\sigma_x$, $\sigma_y$ and $\sigma_z$ are the Pauli matrices

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

Such operator $R_{\boldsymbol{a}}(\phi)$ is called the rotation (of a qubit) about the three-dimensional vector $\boldsymbol{a}$ with angle $\phi$ (on the Bloch sphere), and has the following properties:

1. $\mathrm{R}_y(\phi) = R_{(0,1,0)}(-\phi) = R_{(0,-1,0)}(\phi)$ for all $\phi \in \mathbb{R}$.
2. $\mathrm{R}_z(\phi) = R_{(0,0,1)}(-\phi) = R_{(0,0,-1)}(\phi)$ for all $\phi \in \mathbb{R}$.
3. $R_{\boldsymbol{a}}(\phi)^\dagger = R_{\boldsymbol{a}}(-\phi)$ for all unit vectors $\boldsymbol{a} \in \mathbb{R}^3$ and $\phi \in \mathbb{R}$.
4. $R_{\boldsymbol{a}}(\phi)$ is unitary for all unit vectors $\boldsymbol{a} \in \mathbb{R}^3$ and $\phi \in \mathbb{R}$.
5. $R_{\boldsymbol{a}}(\theta)R_{\boldsymbol{a}}(\phi) = R_{\boldsymbol{a}}(\theta + \phi)$ for all unit vectors $\boldsymbol{a} \in \mathbb{R}$ and $\theta, \phi \in \mathbb{R}$.
6. $\mathrm{X}R_{\boldsymbol{a}}(\phi)\mathrm{X} = R_{\boldsymbol{a}}(-\phi)$ for all unit vectorss $\boldsymbol{a} = (0, a_y, a_z) \in \mathbb{R}^3$ and $\phi \in \mathbb{R}$.

We use the following example of the idea of the implementation of a 4-qubit multi-controlled rotation gate of type $F^3_4(R_{\boldsymbol{a}})$ with unit vector $\boldsymbol{a} = (0, a_y, a_z)$.

**Example 3.123.** Let $\boldsymbol{a} = (0, a_y, a_z)$ be a unit vector in $\mathbb{R}^3$. In this example we consider the multi-controlled 4-qubit gate given by

$$|k\rangle \otimes |y\rangle \mapsto |j\rangle \otimes (R_{\boldsymbol{a}}(\alpha_j)|y\rangle) \qquad \text{if } |k\rangle = |j\rangle,$$

where $|k\rangle = |k_3\rangle \otimes |k_2\rangle \otimes |k_1\rangle$ with $k = (k_3k_2k_2)_2$. In Example 3.106, we have shown that the matrix representation of this quantum gate is

$$\text{blkdiag}(R_{\boldsymbol{a}}(\alpha_1), R_{\boldsymbol{a}}(\alpha_2), \cdots, R_{\boldsymbol{a}}(\alpha_8)),$$

and we would like to express this multi-controlled gate as a sequence of implementable quantum circuits (such as **CNOT** and some 1-qubit gate) on a 4-qubit system. In particular, we would like to find $C_1, C_2, \cdots, C_k \in \{\mathbf{CNOT}_{1,4}, \mathbf{CNOT}_{2,4}, \mathbf{CNOT}_{3,4}\}$ and $16 \times 16$ block diagonal matrix $R_1, R_2, \cdots, R_k$ of the form $R_j = \mathrm{blkdiag}\big(\underbrace{R_{\boldsymbol{a}}(\theta_j), R_{\boldsymbol{a}}(\theta_j), \cdots, R_{\boldsymbol{a}}(\theta_j)}_{8 \text{ copies of } R_{\boldsymbol{a}}(\theta_j)}\big)$ (which is the matrix representation of $\mathrm{I}_2 \otimes \mathrm{I}_2 \otimes \mathrm{I}_2 \otimes R_{\boldsymbol{a}}(\theta_j)$) for some $\theta_j \in \mathbb{R}$ so that

$$\mathrm{blkdiag}(R_{\boldsymbol{a}}(\alpha_1), R_{\boldsymbol{a}}(\alpha_2), \cdots, R_{\boldsymbol{a}}(\alpha_8)) = C_8 R_8 C_7 R_7 \cdots C_2 R_2 C_1 R_1 \,. \tag{3.30}$$

Here we recall that $\mathbf{CNOT}_{i,4}$ denotes the controlled-not gate whose control qubit is the $i$-th qubit while the target qubit is the 4-th qubit, and the matrix representation of $\mathbf{CNOT}_{1,4}$, $\mathbf{CNOT}_{2,4}$, $\mathbf{CNOT}_{3,4}$ are given by

$$\begin{aligned}
\mathbf{CNOT}_{1,4} &= \mathrm{blkdiag}(\mathrm{I}_2, \mathrm{I}_2, \mathrm{I}_2, \mathrm{I}_2, \mathrm{X}, \mathrm{X}, \mathrm{X}, \mathrm{X})\,, \\
\mathbf{CNOT}_{2,4} &= \mathrm{blkdiag}(\mathrm{I}_2, \mathrm{I}_2, \mathrm{X}, \mathrm{X}, \mathrm{I}_2, \mathrm{I}_2, \mathrm{X}, \mathrm{X})\,, \\
\mathbf{CNOT}_{3,4} &= \mathrm{blkdiag}(\mathrm{I}_2, \mathrm{X}, \mathrm{I}_2, \mathrm{X}, \mathrm{I}_2, \mathrm{X}, \mathrm{I}_2, \mathrm{X})\,.
\end{aligned}$$

Define $\widetilde{R}_k$ by

$$\widetilde{R}_k = C_8 C_7 \cdots C_k R_k C_k C_{k+1} \cdots C_8 \,.$$

By the fact that $C_j C_j = \mathrm{I}_{16}$ and $C_j C_k = C_k C_j$ for all $1 \leqslant j, k \leqslant 8$, we find that

$$\begin{aligned}
\widetilde{R}_8 \widetilde{R}_7 \cdots \widetilde{R}_1 &= (C_8 R_8 C_8)(C_8 C_7 R_7 C_7 C_8)(C_8 C_7 C_6 R_6 C_6 C_7 C_8) \cdots (C_8 \cdots C_1 R_1 C_1 \cdots C_8) \\
&= (C_8 R_8)(C_7 R_7 C_7)(C_7 C_6 R_6 C_6 C_7) \cdots (C_7 \cdots C_1 R_1 C_1 \cdots C_8) \\
&= (C_8 R_8)(C_7 R_7)(C_6 R_6 C_6) \cdots (C_6 \cdots C_1 R_1 C_1 \cdots C_8) \\
&= \cdots\cdots\cdots \\
&= (C_8 R_8)(C_7 R_7)(C_6 R_6) \cdots (C_1 R_1)(C_1 \cdots C_8)
\end{aligned}$$

so that

$$C_8 R_8 C_7 R_7 \cdots C_1 R_1 = \widetilde{R}_8 \widetilde{R}_7 \cdots \widetilde{R}_1 \cdot (C_1 C_2 \cdots C_8) \,.$$

Since $\mathrm{X}R_{\boldsymbol{a}}(\phi)\mathrm{X} = -R_{\boldsymbol{a}}(\phi)$ for all $\boldsymbol{a} = (0, a_y, a_z)$, we find that for all $\phi_1, \phi_2, \cdots, \phi_8 \in \mathbb{R}$,

$$\begin{aligned}
&\mathbf{CNOT}_{1,4} \cdot \mathrm{blkdiag}\big(R_{\boldsymbol{a}}(\phi_1), R_{\boldsymbol{a}}(\phi_2), \cdots, R_{\boldsymbol{a}}(\phi_8)\big) \cdot \mathbf{CNOT}_{1,4} \\
&\qquad = \mathrm{blkdiag}\big(R_{\boldsymbol{a}}(\phi_1), R_{\boldsymbol{a}}(\phi_2), R_{\boldsymbol{a}}(\phi_3), R_{\boldsymbol{a}}(\phi_4), R_{\boldsymbol{a}}(-\phi_5), R_{\boldsymbol{a}}(-\phi_6), R_{\boldsymbol{a}}(-\phi_7), R_{\boldsymbol{a}}(-\phi_8)\big)\,, \\
&\mathbf{CNOT}_{2,4} \cdot \mathrm{blkdiag}\big(R_{\boldsymbol{a}}(\phi_1), R_{\boldsymbol{a}}(\phi_2), \cdots, R_{\boldsymbol{a}}(\phi_8)\big) \cdot \mathbf{CNOT}_{2,4} \\
&\qquad = \mathrm{blkdiag}\big(R_{\boldsymbol{a}}(\phi_1), R_{\boldsymbol{a}}(\phi_2), R_{\boldsymbol{a}}(-\phi_3), R_{\boldsymbol{a}}(-\phi_4), R_{\boldsymbol{a}}(\phi_5), R_{\boldsymbol{a}}(\phi_6), R_{\boldsymbol{a}}(-\phi_7), R_{\boldsymbol{a}}(-\phi_8)\big)\,, \\
&\mathbf{CNOT}_{3,4} \cdot \mathrm{blkdiag}\big(R_{\boldsymbol{a}}(\phi_1), R_{\boldsymbol{a}}(\phi_2), \cdots, R_{\boldsymbol{a}}(\phi_8)\big) \cdot \mathbf{CNOT}_{3,4} \\
&\qquad = \mathrm{blkdiag}\big(R_{\boldsymbol{a}}(\phi_1), R_{\boldsymbol{a}}(-\phi_2), R_{\boldsymbol{a}}(\phi_3), R_{\boldsymbol{a}}(-\phi_4), R_{\boldsymbol{a}}(\phi_5), R_{\boldsymbol{a}}(-\phi_6), R_{\boldsymbol{a}}(\phi_7), R_{\boldsymbol{a}}(-\phi_8)\big)\,.
\end{aligned}$$

Therefore, $\widetilde{R}_k$ must take the form

$$\mathrm{blkdiag}\big(R_{\boldsymbol{a}}(b_{k1}\theta_k), R_{\boldsymbol{a}}(b_{k2}\theta_k), \cdots, R_{\boldsymbol{a}}(b_{k8}\theta_k)\big)\,,$$

where $b_{kj} = \pm 1$ and $b_{kj}$ is determined by $C_k, \cdots, C_8$. In fact, with $\boldsymbol{r}_j$ denoting the symbol of $C_j$ (see Remark 3.103 for symbols of CNOT gates) and $.*$ denoting the Hadamard product given by

$$\big[\,u_1, u_2, \cdots, u_n\,\big] .* \big[\,v_1, v_2, \cdots, v_n\,\big] = \big[\,u_1v_1, u_2v_2, \cdots, u_nv_n\,\big]\,,$$

we have

$$\boldsymbol{b}_k \equiv \big[\,b_{k1}, b_{k2}, \cdots, b_{k8}\,\big] = \boldsymbol{r}_k .* \cdots .* \boldsymbol{r}_7 .* \boldsymbol{r}_8\,. \tag{3.31}$$

Let M be the $8 \times 8$ Hadamard matrix; that is,

$$\mathrm{M} \equiv \sqrt{2}^{\,3}\mathrm{H}_3 = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1
\end{bmatrix}.$$

and denote the $k$-th row of M by $S_{k-1}$. The symbol for $\mathbf{CNOT}_{3,4}$, $\mathbf{CNOT}_{2,4}$ and $\mathbf{CNOT}_{1,4}$ are then $S_1$, $S_2$ and $S_4$, respectively. Note that the identity $R_{\boldsymbol{a}}(\theta)R_{\boldsymbol{a}}(\phi) = R_{\boldsymbol{a}}(\theta+\phi)$ implies

that

$$\begin{aligned}\widetilde{R}_8\widetilde{R}_7\cdots\widetilde{R}_1 &= \mathrm{blkdiag}\big(R_{\boldsymbol{a}}(b_{81}\theta_8), R_{\boldsymbol{a}}(b_{82}\theta_8), \cdots, R_{\boldsymbol{a}}(b_{88}\theta_8)\big)\\ &\quad\cdot \mathrm{blkdiag}\big(R_{\boldsymbol{a}}(b_{71}\theta_7), R_{\boldsymbol{a}}(b_{72}\theta_7), \cdots, R_{\boldsymbol{a}}(b_{78}\theta_7)\big)\\ &\quad\cdots\cdot \mathrm{blkdiag}\big(R_{\boldsymbol{a}}(b_{11}\theta_1), R_{\boldsymbol{a}}(b_{12}\theta_1), \cdots, R_{\boldsymbol{a}}(b_{18}\theta_1)\big)\\ &= \mathrm{blkdiag}\big(R_{\boldsymbol{a}}(b_{81}\theta_8 + b_{71}\theta_7 + \cdots + b_{11}\theta_1), R_{\boldsymbol{a}}(b_{82}\theta_8 + b_{72}\theta_7 + \cdots + b_{12}\theta_1), \cdots,\\ &\qquad R_{\boldsymbol{a}}(b_{88}\theta_8 + b_{78}\theta_8 + \cdots + b_{18}\theta_1)\big)\,. \end{aligned} \tag{3.32}$$

The computation above motivates us to choose $\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_8 \in \{S_1, S_2, S_4\}$ (which is equivalent of choosing $C_1, \cdots, C_8 \in \{\mathbf{CNOT}_{1,4}, \mathbf{CNOT}_{2,4}, \mathbf{CNOT}_{3,4}\}$) such that $\boldsymbol{b}_k$ given by $\boldsymbol{b}_k = \boldsymbol{r}_k .* \cdots .* \boldsymbol{r}_7 .* \boldsymbol{r}_8$ satisfying

1. $\boldsymbol{b}_1 = S_0$ (if so, then $C_1C_2\cdots C_8 = \mathrm{I}_{16}$ which implies that $C_8R_8C_7R_7\cdots C_1R_1 = \widetilde{R}_8\widetilde{R}_7\cdots\widetilde{R}_1$).

2. The collection $\{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_8\}$ is linearly independent (so that it is a permutation of $\{S_0, S_1, \cdots, S_7\}$).

If we are able to find such $\boldsymbol{r}_1, \cdots, \boldsymbol{r}_8$, then we choose $\theta_1, \cdots, \theta_8$ satisfying

$$\begin{bmatrix} b_{11} & b_{21} & \cdots & b_{81}\\ b_{12} & b_{22} & \cdots & b_{82}\\ \vdots & \vdots & \ddots & \vdots\\ b_{18} & b_{28} & \cdots & b_{88}\end{bmatrix}\begin{bmatrix}\theta_1\\ \theta_2\\ \vdots\\ \theta_8\end{bmatrix} = \begin{bmatrix}\alpha_1\\ \alpha_2\\ \vdots\\ \alpha_8\end{bmatrix} \tag{3.33}$$

whose solvability is guaranteed by property 2 above (since the $B$ matrix has full column rank). Such $\theta_k$'s will then verify (3.30) becuase of (3.32) and (3.33), as well as the fact that $C_1\cdots C_8 = \mathrm{I}_{16}$.

Finally, let us talk about how to find $\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_8 \in \{S_1, S_2, S_4\}$ satisfying the two properties above. First we establish some rules of multiplications of $S_i's$ (since $\boldsymbol{b}$ are Hadamard product of some $\boldsymbol{r}$'s). Note that

$$S_1 .* S_2 = S_3\,, \quad S_1 .* S_4 = S_5\,, \quad S_2 .* S_4 = S_6\,, \quad S_1 .* S_2 .* S_4 = S_7$$

so we have

$$\mathrm{M} = \begin{bmatrix} \mathbf{ones}(1,8) \equiv S_0 \\ S_1 \\ S_2 \\ S_1 \mathbin{.*} S_2 \\ S_4 \\ S_1 \mathbin{.*} S_4 \\ S_2 \mathbin{.*} S_4 \\ S_1 \mathbin{.*} S_2 \mathbin{.*} S_4 \end{bmatrix} .$$

Therefore, all rows of M can be generated by $S_1$, $S_2$ and $S_4$ using the Hadamard product $.*$ and we have

$$S_i \mathbin{.*} S_j = S_{i+j} \quad \forall\, i,j \in \{1,2,4\} \qquad \text{and} \qquad S_1 \mathbin{.*} S_2 \mathbin{.*} S_4 = S_7\,. \tag{3.34}$$

In order to compute $S_i \mathbin{.*} S_j$ for general $0 \leqslant i,j \leqslant 7$, we write

$$S_\ell = S_1^{x_\ell} \mathbin{.*} S_2^{y_\ell} \mathbin{.*} S_4^{z_\ell} \qquad \forall\, 0 \leqslant \ell \leqslant 7$$

and use the formula

$$S_i \mathbin{.*} S_j = S_1^{x_i \oplus x_j} \mathbin{.*} S_2^{y_i \oplus y_j} \mathbin{.*} S_4^{z_i \oplus z_j}\,, \tag{3.35}$$

where $\oplus$ is the addition in $\mathbb{Z}_2$, $S_k^0 \equiv S_0$ for $k = 1,2,4$, and we use the fact that $S_i \mathbin{.*} S_i = S_0$ and $S_i \mathbin{.*} S_j = S_j \mathbin{.*} S_i$ for $S_i, S_j \in \{S_1, S_2, S_4\}$ to conclude the identity. We note that using (3.34), by writing $\ell = (\ell_2\ell_1\ell_0)_2$ we have

$$S_\ell = S_1^{\ell_0} \mathbin{.*} S_2^{\ell_1} \mathbin{.*} S_4^{\ell_2}$$

so that (3.35) becomes

$$S_i \mathbin{.*} S_j = S_1^{i_0 \oplus j_0} \mathbin{.*} S_2^{i_1 \oplus j_1} \mathbin{.*} S_4^{i_2 \oplus j_2} \qquad \forall\, 0 \leqslant i,j \leqslant 7\,, i = (i_2 i_1 i_0)_2, j = (j_2 j_1 j_0)_2\,.$$

The use of (3.34) further shows that

$$S_{(i_2 i_1 i_0)_2} \mathbin{.*} S_{(j_2 j_1 j_0)_2} = S_{(k_2 k_1 k_0)_2} \quad \text{where} \;\; i_\ell, j_\ell \in \{0,1\} \text{ and } k_\ell = i_\ell \oplus j_\ell\,.$$

By identifying $S_{(\ell_2\ell_1\ell_0)_2}$ as $(\ell_2, \ell_1, \ell_0)$, we find that the group $(\{S_0, S_1, \cdots, S_7\},\, .*\,)$ is isomorphic to the group $(\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2, \oplus)$, where $\oplus$ on $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$ is given by

$$(i_2, i_1, i_0) \oplus (j_2, j_2, j_0) \equiv (i_2 \oplus j_2, i_1 \oplus j_1, i_0 \oplus j_0)\,, \qquad i_\ell, j_\ell \in \{0,1\}\,;$$

that is, there exists a bijection $\varphi : \{S_0, \cdots, S_7\} \to \mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$ given by $\varphi(S_{(k_2k_1k_0)_2}) = (k_2, k_1, k_0)$ such that

$$\varphi\big(S_{(i_2i_1i_0)_2} \mathbin{.*} S_{(j_2j_1j_0)_2}\big) = \varphi\big(S_{(i_2i_1i_0)_2}\big) \oplus \varphi\big(S_{(j_2j_1j_0)_2}\big) = (i_2 \oplus j_2, i_1 \oplus j_1, i_0 \oplus j_0)\,.$$

Now, since $\boldsymbol{r}_j$ are symbols of $\mathbf{CNOT}_{1,4}$, $\mathbf{CNOT}_{2,4}$ or $\mathbf{CNOT}_{3,4}$, $\boldsymbol{r}_j = S_{(x_jy_jz_j)_2}$ for some $x_j, y_j, z_j \in \{0, 1\}$ with the property that one and only one of $x_j, y_j, z_j$ is 1. Since

$$(x_k, y_k, z_k) \oplus \cdots \oplus (x_8, y_8, z_8) = (x_k \oplus \cdots \oplus x_8, y_k \oplus \cdots \oplus y_8, z_k \oplus \cdots \oplus z_8)\,,$$

we find that $\varphi(\boldsymbol{b}_k)$ and $\varphi(\boldsymbol{b}_{k+1})$, the correspondence of $\boldsymbol{b}_k$ and $\boldsymbol{b}_{k+1}$ in $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$, differs by only one slot/bit (since every addition of new $\boldsymbol{r}_k$ to $\boldsymbol{b}_{k+1}$ corresponds to the addition of $(0, 0, 0)$, $(0, 1, 0)$ or $(1, 0, 0)$ to $\varphi(\boldsymbol{b}_{k+1})$ in $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$). This motivates the idea of the reflected binary code (also called Gray code) which is a scheme for listing all $n$-bit binary numbers so that successive numbers differ in exactly one bit. A 3-qubit reflected Gray code is given by $[0, 1, 3, 2, 6, 7, 5, 4]$. We list these numbers in terms of binary representation in the following table and one can see that adjacent numbers differ by one bit.

| $j = (j_2j_1j_0)_2$ | 0 | 1 | 3 | 2 | 6 | 7 | 5 | 4 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| $j_2$ | 0 | 0 | 0 | 0 | **1** | 1 | 1 | **1** | **0** |
| $j_1$ | 0 | 0 | **1** | 1 | 1 | 1 | **0** | 0 | 0 |
| $j_0$ | 0 | **1** | 1 | **0** | 0 | **1** | 1 | 0 | 0 |

From the table above, $\boldsymbol{b}_1$, $\boldsymbol{b}_2$, $\cdots$, $\boldsymbol{b}_8$ correspond to $(0, 0, 0)$, $(0, 0, 1)$, $\cdots$, $(1, 0, 0)$ in $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$. How do we find $\boldsymbol{r}_j$? Note that $\boldsymbol{b}_k = \boldsymbol{r}_k \mathbin{.*} \boldsymbol{b}_{k+1}$; thus

$$\boldsymbol{r}_k = \boldsymbol{r}_k \mathbin{.*} \boldsymbol{b}_{k+1} \mathbin{.*} \boldsymbol{b}_{k+1} = \boldsymbol{b}_k \mathbin{.*} \boldsymbol{b}_{k+1} \qquad \forall\, 1 \leqslant j \leqslant 7\,, \quad \boldsymbol{r}_8 = \boldsymbol{b}_8\,. \tag{3.36}$$

Therefore, $\boldsymbol{r}_1$ corresponds to the element $(0, 0, 0) \oplus (0, 0, 1)$ in $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$, $\boldsymbol{r}_2$ corresponds to the element $(0, 0, 1) \oplus (0, 1, 1)$ in $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$, and etc. This implies that $\boldsymbol{r}_1 = S_1$ and $\boldsymbol{r}_2 = S_2$, and so on. Note that the addition in fact indicates the bit where $\boldsymbol{b}_k$ and $\boldsymbol{b}_{k+1}$ differ (which is shown as boldface colored 0 or 1 in the table). Moreover, the position of the different bit is in fact the position of the control qubit in the CNOT gate (for example, the bit where $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ differs locates in the 3rd qubit; thus $\boldsymbol{r}_1 = \mathbf{CNOT}_{3,4}$). Therefore, a choice of $C_1, C_2, \cdots, C_8$ can be

$$\big\{\mathbf{CNOT}_{3,4}, \mathbf{CNOT}_{2,4}, \mathbf{CNOT}_{3,4}, \mathbf{CNOT}_{1,4}, \mathbf{CNOT}_{3,4}, \mathbf{CNOT}_{2,4}, \mathbf{CNOT}_{1,4}, \mathbf{CNOT}_{1,4}\big\}\,.$$

We summarize the discussion in Example 3.123 and state the general procedure of the decomposition of multi-controlled $(n+1)$-qubit gate (with first $n$-qubit as control qubits) as follows. Let $N = 2^n$.

1. Our goal is to write the matrix representation of a multi-controlled gate in the form

$$\text{blkdiag}(R_{\boldsymbol{a}}(\alpha_1), R_{\boldsymbol{a}}(\alpha_2), \cdots, R_{\boldsymbol{a}}(\alpha_N)) = C_N R_N C_{N-1} R_{N-1} \cdots C_2 R_2 C_1 R_1 \,, \tag{3.37}$$

   where $\boldsymbol{a} = (0, a_y, a_z)$ is a unit vector, $C_k \in \{\mathbf{CNOT}_{1,n+1}, \cdots, \mathbf{CNOT}_{n,n+1}\}$ and $R_k = \text{blkdiag}\big(R_{\boldsymbol{a}}(\theta_k), \cdots, R_a(\theta_k)\big)$ for all $1 \leqslant k \leqslant N$.

2. Using the property that $C_i = C_i^{-1}$ and $C_i$, $C_j$ commute, the right-hand side of (3.37) can be rewritten as

$$C_N R_N C_{N-1} R_{N-1} \cdots C_2 R_2 C_1 R_1 = \widetilde{R}_N \widetilde{R}_{N-1} \cdots \widetilde{R}_1 \cdot (C_1 C_2 \cdots C_N) \,,$$

   where $\widetilde{R}_k = (C_k C_{k+1} \cdots C_N) R_k (C_N C_{N-1} \cdots C_k)$.

3. The effect of $C_k \cdots C_N$ on $R_k$ leads to the result

$$\widetilde{R}_k = \text{blkdiag}\big(R_{\boldsymbol{a}}(b_{k1}\theta_k), R_{\boldsymbol{a}}(b_{k2}\theta_k), \cdots, R_{\boldsymbol{a}}(b_{kN}\theta_k)\big) \,,$$

   where $\boldsymbol{b}_k = [b_{k1}, b_{k2}, \cdots, b_{kN}]$ are Hadamard product of the symbols of $C_k$, $\cdots$, $C_N$, respectively, or to be more precise,

$$\boldsymbol{b}_k \equiv \big[\, b_{k1}, b_{k2}, \cdots, b_{kN} \,\big] = \boldsymbol{r}_k \mathbin{.*} \cdots \mathbin{.*} \boldsymbol{r}_{N-1} \mathbin{.*} \boldsymbol{r}_N \,, \tag{3.38}$$

   where $\boldsymbol{r}_j$ are symbol of $C_j$ introduced in Remark 3.103.

4. We choose $\boldsymbol{r}_1, \cdots, \boldsymbol{r}_N$ properly from $\big\{ S_{2^k} \,\big|\, 0 \leqslant k \leqslant n-1 \big\}$, where $S_{2^k}$ is the $(2^k+1)$-th row of $\mathrm{M} = \sqrt{2}^n \mathrm{H}_n$, so that the corresponding $\boldsymbol{b}_k$ satisfies

   (a) $\boldsymbol{b}_1 = \mathbf{ones}(1, N)$;

   (b) the collection $\{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_N\}$ is linearly independent so that it is a permutation of the rows of M.

   Once we have these $\boldsymbol{b}_k$'s, we then solve

$$\begin{bmatrix} b_{11} & b_{21} & \cdots & b_{N1} \\ b_{12} & b_{22} & \cdots & b_{N2} \\ \vdots & & & \vdots \\ b_{1N} & b_{2N} & \cdots & b_{NN} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_8 \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_8 \end{bmatrix} \tag{3.33}$$

   to obtain $\theta_1$, $\cdots$, $\theta_N$.

5. Let $\{x_1, x_2, \cdots, x_N\}$ be a reflected binary code (with $x_1 = 0$) for the list of numbers $\{0, 1, \cdots, N-1\}$, and $f : \{1, \cdots, N\} \to \{1, \cdots n\}$ be defined by

   $f(j)$ is the location where the bit expression of $x_j$ and $x_{j+1}$ differ ($x_{N+1} \equiv 0$).

   Then a choice of $C_1, C_2, \cdots, C_N$ and $\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_N$ are given by

$$C_j = \mathbf{CNOT}_{f(j),n+1}\,, \qquad \boldsymbol{b}_j = \text{ the binary expression of } x_j.$$

**Remark 3.124.** A way to obtain a reflected binary code for the numbers $\{0, 1, 2, \cdots, 2^n - 1\}$ is given as follows: 假設原始的值從 0 開始，格雷碼產生的規律是:

1. 第一步, 改變最右邊的位元值；
2. 第二步，改變右邊起第一個為 1 的位元的左邊的位元；
3. 重複第一步和第二步，直到所有的格雷碼產生完畢。

**Example 3.125.** A Gray code for the case $n = 3$ is given by

$$000 \to 001 \to 011 \to 010 \to 110 \to 111 \to 101 \to 100\,.$$

• **Algorithm of the decomposition of multi-controlled** $(n+1)$**-qubit gates**

Suppose that we are given matrix

$$R = \text{blkdiag}\big(R_{\boldsymbol{a}}(\alpha_1), \cdots, R_{\boldsymbol{a}}(\alpha_N)\big)$$

for some unit vector $\boldsymbol{a} = (0, a_y, a_z)$, where $N = 2^n$.

1. Let $\{x_1, x_2, \cdots, x_N\}$, where $x_1 = 0$, be a reflected binary code (Gray code) for the list of numbers $\{0, 1, \cdots, N-1\}$. Define $x_{N+1} = 0$ and $f : \{1, \cdots, N\} \to \{1, \cdots n\}$ by

   $f(j)$ is the location where the bit expression of $x_j$ and $x_{j+1}$ differ ($x_{N+1} \equiv 0$)

   which can be implemented in matlab® by

$$f(j) = \mathbf{find}(\mathbf{double}(\mathbf{xor}(\mathbf{flip}(\mathbf{de2bi}(x_j, n)), \mathbf{flip}(\mathbf{de2bi}(x_{j+1}, n))))) == 1)\,.$$

   Set $C_j = \mathbf{CNOT}_{f(j),n+1}$ for $1 \leqslant j \leqslant N$.

2. Define a $2^n \times 2^n$ matrix $M = [m_{ij}]$ by

$$m_{ij} = (-1)^{(i-1)\bullet x_j}$$

where the exponent $(i-1)\bullet(x_j)$ is the bitwise dot product (defined in (3.8)) of $(i-1)$ and $x_j$ (which can be implemented in matlab® by **de2bi**$(i-1,n) * $ **de2bi**$(x_j,n)'$). Solve

$$M \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_N \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix} .$$

3. Define $R_k = \text{blkdiag}\big(\underbrace{R_{\boldsymbol{a}}(\theta_k), \cdots, R_{\boldsymbol{a}}(\theta_k)}_{N \text{ copies of } R_{\boldsymbol{a}}(\theta_k)}\big)$. Then $R = C_N R_N C_{N-1} R_{N-1} \cdots C_1 R_1$.

We note that in matlab® $R_k$ can be formed by

$$R_k = \textbf{kron}(\textbf{eye}(N), R_{\boldsymbol{a}}(\theta_k)) .$$

# Chapter 4

# Simon's Algorithm

Simon's algorithm was the first quantum algorithm to show an exponential speed-up versus the best classical algorithm in solving a specific problem. This inspired the quantum algorithms based on the quantum Fourier transform, which is used in the most famous quantum algorithm: Shor's factoring algorithm.

## 4.1 Simon's Problem

Let $N = 2^n$, and identify the set $\{0, \cdots, N-1\}$ with $\{0,1\}^n$. Let $j \oplus s$ be the $n$-bit string obtained by bitwise adding the $n$-bit strings $j$ and $s$ mod 2; that is,

$$j \oplus s = \big((j_1 \oplus s_1)(j_2 \oplus s_2) \cdots (j_n \oplus s_n)\big)_2 \qquad \text{if } \; j = (j_1 j_2 \cdots j_n)_2 \text{ and } s = (s_1 s_2 \cdots s_n)_2 \,.$$

**Simon's problem**:

- **Formulation 1**: For $N = 2^n$, we are given $x = (x_0, \cdots, x_{N-1})$, with $x_i \in \{0,1\}^n$, with the property that there is some unknown nonzero $s \in \{0,1\}^n$ such that $x_i = x_j$ if and only if $(i = j$ or $i = j \oplus s)$. Find $s$.

- **Formulation 2**: If $f : \{0,1\}^n \to \{0,1\}^n$ is either an one-to-one or a two-to-one function satisfying the property that there exists $s \in \{0,1\}^n$ such that $f(i) = f(j)$ if and only if $i = j$ or $i = j \oplus s$. Determine the class to which $f$ belongs to.

Note that the input here are slightly different from before: the input $x = \{x_0, \cdots, x_{N-1}\}$ now has variables $x_i$ that themselves are $n$-bit strings, and one query gives such a string completely $|i0^n\rangle \mapsto |ix_i\rangle$.

## 4.2 The Quantum Algorithm

Simon's algorithm starts out very similar to Deutsch-Jozsa: start in a state of $2^n$ zero qubits $|0^n\rangle|0^n\rangle$ and apply Hadamard transforms to the first $n$ qubits, giving

$$\frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} |i\rangle|0^n\rangle .$$

At this point, the second $n$-qubit register still holds only zeroes. A query turns this into

$$\frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} |i\rangle|x_i\rangle .$$

Now the algorithm measures the second $n$-qubit register in the computational basis; this measurement is actually not necessary, but it facilitates analysis. The measurement outcome will be some value $x_i$ and the first register will collapse to the superposition of the two indices having that $x_i$-value:

$$\frac{1}{\sqrt{2}}(|i\rangle + |i \oplus s\rangle)|x_i\rangle .$$

We will now ignore the second register and apply Hadamard transforms to the first $n$ qubits. Using Equation (2.9) and the fact that $(i \oplus s) \bullet j = (i \bullet j) \oplus (s \bullet j)$ (which is a direct consequence of $(i_k \oplus s_k) \cdot j_k = (i_k \cdot j_k) \oplus s_k \cdot j_k)$ for all $i_k, s_k, j_k \in \{0,1\}$), we can write the resulting state as

$$\begin{aligned} &\frac{1}{\sqrt{2^{n+1}}} \left( \sum_{j \in \{0,1\}^n} (-1)^{i \bullet j} |j\rangle + \sum_{j \in \{0,1\}^n} (-1)^{(i \oplus s) \bullet j} |j\rangle \right) \\ &\quad = \frac{1}{\sqrt{2^{n+1}}} \sum_{j \in \{0,1\}^n} (-1)^{i \bullet j} \big(1 + (-1)^{s \bullet j}\big) |j\rangle . \end{aligned}$$

Note that $|j\rangle$ has nonzero amplitude if $s \bullet j = 0 \bmod 2$. Measuring the state gives a uniformly random element from the set $\{j \mid s \bullet j = 0 \bmod 2\}$. Accordingly, we get a linear equation that gives information about $s$. We repeat this algorithm until we have obtained $n-1$ independent linear equations involving $s$. The solutions to these equations will be $0^n$ and the correct $s$, which we can compute efficiently by a classical algorithm (Gaussian elimination modulo 2). This can be done by means of a classical circuit of size roughly $\mathcal{O}(n^3)$.

Note that if the $j$'s you have generated at some point span a space of size $2^k$, for some $k < n-1$, then the probability that your next run of the algorithm produces a $j$ that is

linearly independent of the earlier ones, is $(2^n - 2^k)/2^n \geqslant 1/2$. Hence an expected number of $\mathcal{O}(n)$ runs of the algorithm suffices to find $n-1$ linearly independent $j$'s. Simon's algorithm thus finds $s$ using an expected number of $\mathcal{O}(n)$ $x_i$-queries and polynomially many other operations.

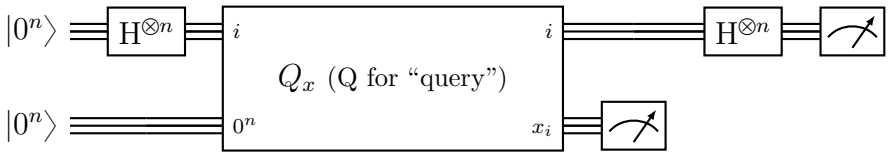



Figure 4.1: Quantum circuit for Simon's algorithm

**Example 4.1.** Let us see the example of Simon's algorithm for periodic function of 2 qubits given by

$$f(x_1, x_2) = (x_1 \oplus x_2, x_1 \oplus x_2) \qquad \forall\, x_1, x_2 \in \{0, 1\}\,.$$

The period $s = (11)_2$, and the quantum circuit to solve the problem is:

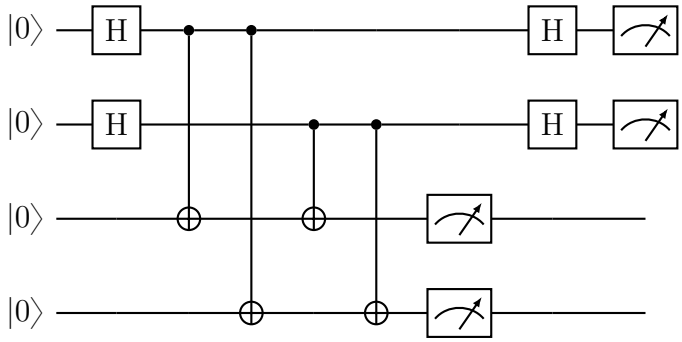



Figure 4.2: Quantum circuit for Simon's algorithm in this example

To check the four CNOT operations indeed provide the oracle $Q_f$, we note that by writing $|x\rangle = |x_1 x_2\rangle$ and $|y\rangle = |y_1\rangle|y_2\rangle$, we have

$$\begin{aligned}
&\mathbf{CNOT}_{2,4}\mathbf{CNOT}_{2,3}\mathbf{CNOT}_{1,4}\mathbf{CNOT}_{1,3}|x\rangle|y\rangle \\
&\quad = \mathbf{CNOT}_{2,4}\mathbf{CNOT}_{2,3}\mathbf{CNOT}_{1,4}\mathbf{CNOT}_{1,3}|x_1\rangle|x_2\rangle|y_1\rangle|y_2\rangle \\
&\quad = \mathbf{CNOT}_{2,4}\mathbf{CNOT}_{2,3}\mathbf{CNOT}_{1,4}|x_1\rangle|x_2\rangle|x_1 \oplus y_1\rangle|y_2\rangle \\
&\quad = \mathbf{CNOT}_{2,4}\mathbf{CNOT}_{2,3}|x_1\rangle|x_2\rangle|x_1 \oplus y_1\rangle|x_1 \oplus y_2\rangle \\
&\quad = \mathbf{CNOT}_{2,4}|x_1\rangle|x_2\rangle|x_1 \oplus x_2 \oplus y_1\rangle|x_1 \oplus y_2\rangle \\
&\quad = |x_1\rangle|x_2\rangle|x_1 \oplus x_2 \oplus y_1\rangle|x_1 \oplus x_2 \oplus y_2\rangle = |x\rangle|y \oplus f(x)\rangle = Q_f|x\rangle|y\rangle\,.
\end{aligned}$$

## 4.3 Classical Algorithms for Simon's Problem

### 4.3.1 Upper bound

Let us first sketch a classical randomized algorithm that solves Simon's problem using $\mathcal{O}(\sqrt{2^n})$ queries, based on the so-called "birthday paradox". Our algorithm will make $T$ randomly chosen distinct queries $i_1, \cdots, i_T$, for some T to be determined later. If there is a collision among those queries (that is, $x_{i_k} = x_{i_\ell}$ for some $k \neq \ell$), then we are done, because then we know $i_k = i_\ell \bmod s$, equivalently $s = i_k \oplus i_\ell$. How large should $T$ be such that we are likely to see a collision in case $s \neq 0^n$? (there will not be any collisions if $s = 0^n$.) There are $C_2^T = \dfrac{T(T-1)}{2} \approx T^2/2$ pairs in our sequence that could be a collision, and since the indices are chosen randomly, the probability for a fixed pair to form a collision is $1/(2^n - 1)$. Hence by linearity of expectation, the expected number of collisions in our sequence will be roughly $T^2/2^{n+1}$. If we choose $T = \sqrt{2^{n+1}}$, we expect to have roughly 1 collision in our sequence, which is good enough to find $s$. Of course, an expected value of 1 collision does not mean that we will have at least one collision with high probability, but a slightly more involved calculation shows the latter statement as well.

### 4.3.2 Lower bound

Simon proved that any classical randomized algorithm that finds $s$ with high probability needs to make $\Omega(\sqrt{2^n})$ queries, so the above classical algorithm is essentially optimal. This was the first proven exponential separation between quantum algorithms and classical bounded-error algorithms (let us stress again that this does not prove an exponential separation in the usual circuit model, because we are counting queries rather than ordinary operations here). Simon's algorithm inspired Shor to his factoring algorithm.

We will prove the classical lower bound for a decision version of Simon's problem:

**Given**: input $x = (x_0, \cdots, x_{N-1})$, where $N = 2^n$ and $x_i \in \{0,1\}^n$.

**Promise**: there exists $s \in \{0,1\}^n$ such that $x_i = x_j$ if and only if ($i = j$ or $i = j \oplus s$).

**Task**: decide whether $s = 0^n$.

Consider the input distribution $\mu$ that is defined as follows. With probability 1/2, $x$ is a uniformly random permutation of $\{0,1\}^n$; this corresponds to the case $s = 0^n$. With probability 1/2, we pick a nonzero string $s$ at random, and for each pair $(i, i \oplus s)$, we pick

a unique value for $x_i = x_{i \oplus s}$ at random. If there exists a randomized $T$-query algorithm that achieves success probability $\geqslant 2/3$ under this input distribution $\mu$, then there also is deterministic $T$-query algorithm that achieves success probability $\geqslant 2/3$ under $\mu$ (because the behavior of the randomized algorithm is an average over a number of deterministic algorithms). Now consider a deterministic algorithm with error $\leqslant 1/3$ under $\mu$, that makes $T$ queries to $x$. We want to show that $T = \Omega(\sqrt{2^n})$.

First consider the case $s = 0^n$. We can assume the algorithm never queries the same point twice. Then the $T$ outcomes of the queries are $T$ distinct $n$-bit strings, and each sequence of $T$ strings is equally likely. Now consider the case $s \neq 0^n$. Suppose the algorithm queries the indices $i_1, \cdots, i_T$ (this sequence depends on $x$) and gets outputs $x_{i_1}, \cdots, x_{i_T}$. Call a sequence of queries $i_1, \cdots, i_T$ good if it shows a collision (that is, $x_{i_k} = x_{i_\ell}$ for some $k \neq \ell$), and bad otherwise. If the sequence of queries of the algorithm is good, then we can find $s$, since $i_k \oplus i_\ell = s$. On the other hand, if the sequence is bad, then each sequence of $T$ distinct outcomes is equally likely - just as in the $s = 0^n$ case! We will now show that the probability of the bad case is very close to 1 for small $T$.

If $i_1, \cdots, i_{k-1}$ is bad, then we have excluded at most $C_2^{k-1}$ possible values of $s$ (namely all values $i_j \oplus i_{j'}$ for all distinct $j, j' \in [k-1]$), and all other values of $s$ are equally likely. The probability that the next query $i_k$ makes the sequence good, is the probability that $x_{i_k} = x_{i_j}$ for some $j < k$, equivalently, that the set $S = \{i_k \oplus i_j \,|\, j < k\}$ happens to contain the string $s$. However, $S$ has only $k-1$ members, while there are $2^n - 1 - C_2^{k-1}$ equally likely remaining possibilities for $s$. This means that the probability that the sequence is still bad after query $i_k$ is made, is very close to 1. In formulas:

$$\begin{aligned}\Pr[i_1, \cdots, i_T \text{ is bad}] &= \prod_{k=2}^{T} \Pr[i_1, \cdots, i_k \text{ is bad} | i_1, \cdots i_{k-1} \text{ is bad}] \\ &= \prod_{k=2}^{T} \left(1 - \frac{k-1}{2^n - 1 - C_2^{k-1}}\right) \geqslant 1 - \sum_{k=2}^{T} \frac{k-1}{2^n - 1 - C_2^{k-1}}\,.\end{aligned}$$

Here we used the fact that $(1-a)(1-b) \geqslant 1-(a+b)$ if $a, b \geqslant 0$.

Note that $2^n - 1 - C_2^{k-1} \approx 2^n$ as long as $k \ll \sqrt{2^n}$, and $\sum_{k=2}^{T}(k-1) = \frac{T(T-1)}{2} \approx T^2/2$. Hence we can approximate the last term in the formula by $1 - T^2/2^{n+1}$ if $k \ll \sqrt{2^n}$. Accordingly, if $T \ll \sqrt{2^n}$ then with probability nearly 1 (probability taken over the distribution $\mu$) the algorithm's sequence of queries is bad. If it gets a bad sequence, it cannot "see" the difference between the $s = 0^n$ case and the $s \neq 0^n$ case, since both cases result in

a uniformly random sequence of $T$ distinct $n$-bit strings as answers to the $T$ queries. This shows that $T$ has to be $\sqrt{2^n}$ in order to enable the algorithm to get a good sequence of queries with high probability.

# Chapter 5

# The Fourier Transform

## 5.1 The Classical Discrete Fourier Transform

The Fourier transform occurs in many different versions throughout classical computing, in areas ranging from signal-processing to data compression to complexity theory. For our purposes, the Fourier transform is going to be an $N \times N$ unitary matrix, all of whose entries have the same magnitude. For $N = 2$, it's just our familiar Hadamard transform:

$$F_2 = \mathrm{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Doing something similar in 3 dimensions is impossible with real numbers: we cannot give three orthogonal vectors in $\{1, -1\}^3$. However, using complex numbers allows us to define the Fourier transform for any $N$. Let $\omega_N = \exp\left(\frac{2\pi i}{N}\right)$ be an $N$-th root of unity. The rows of the matrix will be indexed by $j \in \{0, \cdots, N-1\}$ and the columns by $k \in \{0, \cdots, N-1\}$. Define the $(j, k)$-entry (so we use the $(0, 0)$-entry to denote the usual $(1, 1)$-entry) of the matrix $F_N$ by $\frac{1}{\sqrt{N}}\omega_N^{jk}$:

$$F_N = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_N & \omega_N^2 & \cdots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \cdots & \omega_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \cdots & \omega_N^{(N-1)(N-1)} \end{bmatrix}.$$

Note that $F_N$ is a unitary matrix, since each column has norm 1, and any pair of columns (say those indexed by $k$ and $k'$) is orthogonal:

$$\sum_{j=0}^{N-1} \frac{1}{\sqrt{N}} \overline{\omega_N^{jk}} \cdot \frac{1}{\sqrt{N}} \omega_N^{jk'} = \frac{1}{N} \sum_{j=0}^{N-1} \omega_N^{j(k'-k)} = \begin{cases} 1 & \text{if } k' = k\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

Since $F_N$ is unitary and symmetric, the inverse $F_N^{-1} = F_N^*$ only differs from $F_N$ by having minus signs in the exponent of the entries. For a vector $v \in \mathbb{R}^N$, the vector $\widehat{v} = F_N v$ is called the discrete Fourier transform (DFT) of $v$. Doing the matrix-vector multiplication, its entries are given by $\widehat{v}_j = \dfrac{1}{\sqrt{N}} \sum\limits_{k=1}^{N} \omega_N^{jk} v_k$.

## 5.2 The Fast Fourier Transform

The naive way of computing the Fourier transform $\widehat{v} = F_N v$ of $v \in \mathbb{R}^N$ just does the matrix-vector multiplication to compute all the entries of $\widehat{v}$. This would take $\mathcal{O}(N)$ steps (additions and multiplications) per entry, and $\mathcal{O}(N^2)$ steps to compute the whole vector $\widehat{v}$. However, there is a more efficient way of computing $\widehat{v}$. This algorithm is called the Fast Fourier Transform (FFT, due to Cooley and Tukey in 1965), and takes only $\mathcal{O}(N \log_2 N)$ steps. This difference between the quadratic $N^2$ steps and the near-linear $N \log_2 N$ is tremendously important in practice when $N$ is large, and is the main reason that Fourier transforms are so widely used.

We will assume $N = 2^n$, which is usually fine because we can add zeroes to our vector to make its dimension a power of 2 (but similar FFTs can be given also directly for most $N$ that are not a power of 2). The key to the FFT is to rewrite the entries of $\widehat{v}$ as follows:

$$\begin{aligned} \widehat{v}_j &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{jk} v_k = \frac{1}{\sqrt{N}} \Big( \sum_{k \text{ even}} \omega_N^{jk} v_k + \sum_{k \text{ odd}} \omega_N^{jk} v_k \Big) \\ &= \frac{1}{\sqrt{2}} \Big( \frac{1}{\sqrt{N/2}} \sum_{k \text{ even}} \omega_{N/2}^{jk/2} v_k + \frac{\omega_N^j}{\sqrt{N/2}} \sum_{k \text{ odd}} \omega_{N/2}^{j(k-1)/2} v_k \Big)\,. \end{aligned}$$

Note that we have rewritten the entries of the $N$-dimensional discrete Fourier transform $\widehat{v}$ in terms of two $\dfrac{N}{2}$-dimensional discrete Fourier transforms, one of the even-numbered entries of $v$, and one of the odd-numbered entries of $v$. This suggests a recursive procedure for computing $\widehat{v}$: first separately compute the Fourier transform $\widehat{v_{\text{even}}}$ of the $\dfrac{N}{2}$-dimensional

vector of even-numbered entries of $v$ and the discrete Fourier transform $\widehat{v_{\text{odd}}}$ of the $\frac{N}{2}$-dimensional vector of odd-numbered entries of $v$, and then compute the $N$ entries using

$$\widehat{v}_j = \frac{1}{\sqrt{2}}\big[(\widehat{v_{\text{even}}})_j + \omega_N^j (\widehat{v_{\text{odd}}})_j\big] \qquad \forall\, 0 \leqslant j \leqslant \frac{N}{2} - 1\,,$$
$$\widehat{v}_{j+\frac{N}{2}} = \frac{1}{\sqrt{2}}\big[(\widehat{v_{\text{even}}})_j - \omega_N^j (\widehat{v_{\text{odd}}})_j\big] \qquad \forall\, 0 \leqslant j \leqslant \frac{N}{2} - 1\,.$$

The computation time $T(N)$ it takes to implement $F_N$ this way can be written recursively as $T(N) = 2T\big(\frac{N}{2}\big) + 2N$, because we need to compute two $\frac{N}{2}$-dimensional Fourier transforms and do $2N$ additional operations (additions and multiplications) to compute $\widehat{v}$. This works out to time $T(N) = \mathcal{O}(N \log_2 N)$, as promised. Similarly, we have an equally efficient algorithm for the inverse discrete Fourier transform $F_N^{-1} = F_N^*$, whose entries are $\frac{1}{\sqrt{N}}\omega_N^{-jk}$.

## 5.3 Application: Multiplying Two Polynomials

Suppose we are given two real-valued polynomials $p$ and $q$, each of degree at most $d$:

$$p(x) = \sum_{j=0}^{d} a_j x^j \qquad \text{and} \qquad q(x) = \sum_{k=0}^{d} b_k x^k\,.$$

We would like to compute the product of these two polynomials

$$p(x)q(x) = \Big(\sum_{j=0}^{d} a_j x^j\Big)\Big(\sum_{k=0}^{d} b_k x^k\Big) = \sum_{\ell=0}^{2d} \underbrace{\Big(\sum_{j=0}^{\ell} a_j b_{\ell-j}\Big)}_{c_\ell} x^\ell\,.$$

Clearly, each coefficient $c_\ell$ by itself takes $(2\ell + 1)$ steps (additions and multiplications) to compute, which suggests an algorithm for computing the coefficients of $p \cdot q$ that takes $\mathcal{O}(d^2)$ steps. However, using the fast Fourier transform we can do this in $\mathcal{O}(d \log_2 d)$ steps, as follows.

The convolution of two vectors $a, b \in \mathbb{R}^N$ is a vector $a * b \in \mathbb{R}^N$ whose $\ell$-th entry is defined by

$$(a * b)_\ell = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} a_j b_{(\ell-j) \bmod N}\,.$$

Let us set $N = 2d + 1$ (the number of nonzero coefficients of $p \cdot q$) and make the $(d + 1)$-dimensional vectors of coefficients $a$ and $b$ N-dimensional by adding $d$ zeroes. Then the

coefficients of the polynomial $p \cdot q$ are proportional to the entries of the convolution: $c_\ell = \sqrt{N}(a * b)_\ell$. It is easy to show that the Fourier coefficients of the convolution of $a$ and $b$ are the products of the Fourier coefficients of $a$ and $b$: for every $\ell \in \{0, ..., N-1\}$ we have $(\widehat{a * b})_\ell = (\widehat{a} .* \widehat{b})_\ell$:

$$\begin{aligned}
(\widehat{a * b})_\ell &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{\ell k} (a * b)_k = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} \omega_N^{\ell k} a_j b_{(k-j) \bmod N} \\
&= \frac{1}{N} \sum_{j=0}^{N-1} \omega_N^{\ell j} a_j \sum_{k=0}^{N-1} \omega_N^{\ell (k-j)} a_j b_{(k-j) \bmod N} \\
&= \frac{1}{N} \sum_{j=0}^{N-1} \omega_N^{\ell j} a_j \sum_{k=0}^{N-1} \omega_N^{\ell k} a_j b_{k \bmod N} \\
&= \Big( \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega_N^{\ell j} a_j \Big) \Big( \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{\ell k} a_j b_{k \bmod N} \Big) = (\widehat{a} .* \widehat{b})_\ell \,,
\end{aligned}$$

where we have used the periodicity to conclude that

$$\sum_{k=0}^{N-1} \omega_N^{\ell (k-j)} a_j b_{(k-j) \bmod N} = \sum_{k=0}^{N-1} \omega_N^{\ell k} a_j b_{k \bmod N} \,.$$

This immediately suggests an algorithm for computing the vector of coefficients $c_\ell$: apply the FFT to $a$ and $b$ to get $\widehat{a}$ and $\widehat{b}$, multiply those two vectors entrywise to get $\widehat{a} .* \widehat{b}$, apply the inverse FFT to get $a * b$, and finally multiply $a * b$ with $\sqrt{N}$ to get the vector $c$ of the coefficients of $p \cdot q$. Since the FFTs and their inverse take $\mathcal{O}(N \log_2 N)$ steps, and pointwise multiplication of two $N$-dimensional vectors takes $\mathcal{O}(N)$ steps, this whole algorithm takes $\mathcal{O}(N \log_2 N) = \mathcal{O}(d \log_2 d)$ steps.

Note that if two numbers $a_d \cdots a_1 a_0$ and $b_d \cdots b_1 b_0$ are given in decimal notation, then we can interpret the digits as coefficients of polynomials $p$ and $q$, respectively, and the two numbers will be $p(10)$ and $q(10)$. Their product is the evaluation of the product-polynomial $p \cdot q$ at the point $x = 10$. This suggests that we can use the above procedure (for fast multiplication of polynomials) to multiply two numbers in $\mathcal{O}(d \log_2 d)$ steps, which would be a lot faster than the standard $\mathcal{O}(d^2)$ algorithm for multiplication that one learns in primary school. However, in this case we have to be careful since the steps of the above algorithm are themselves multiplications between numbers, which we cannot count at unit cost anymore if our goal is to implement a multiplication between numbers! Still, it turns out that implementing this idea carefully allows one to multiply two $d$-digit numbers

in $\mathcal{O}(d \log_2 d \log_2 \log_2 d)$ elementary operations. This is known as the Schönhage-Strassen algorithm. We will skip the details.

## 5.4 The Quantum Fourier Transform

Since $F_N$ is an $N \times N$ unitary matrix, we can interpret it as a quantum operation, mapping an $N$-dimensional vector of amplitudes to another $N$-dimensional vector of amplitudes. This is called the quantum Fourier transform (QFT). In case $N = 2^n$ (which is the only case we will care about), this will be an $n$-qubit unitary. Notice carefully that this quantum operation does something different from the classical Fourier transform: in the classical case we are given a vector $v$, written on a piece of paper so to say, and we compute the vector $\widehat{v} = F_N v$, and also write the result on a piece of paper. In the quantum case, we are working on quantum states; these are vectors of amplitudes, but we do **NOT** have those written down anywhere – they only exist as the amplitudes in a superposition. We will see below that the QFT can be implemented by a quantum circuit using $\mathcal{O}(n^2)$ elementary gates. This is exponentially faster than even the FFT (which takes $\mathcal{O}(N \log_2 N) = \mathcal{O}(2^n n)$ steps), but it achieves something different: computing the QFT will **NOT** give us the entries of the Fourier transform written down on a piece of paper, but only as the amplitudes of the resulting state.

**Definition 5.1.** The $N$-dimnesional quantum Fourier transform $F_N$, where $N = 2^n$, is a linear map on the $n$-qubit space $\{|0\rangle, |1\rangle, \cdots, |N-1\rangle\}$ satisfying that

$$F_N|k\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega_N^{jk} |j\rangle \qquad \forall\, |k\rangle = |k_1 k_2 \cdots k_n\rangle = |k_1\rangle \otimes \cdots \otimes |k_n\rangle,$$

where again $\omega_N = \exp\big(\frac{2\pi i}{N}\big)$.

Since $\exp\big(\frac{2\pi i j k}{2^n}\big) = \exp\Big(i \sum_{\ell=1}^{n} \frac{2\pi k j_\ell}{2^\ell}\Big)$, using (3.17) we find that

$$F_N|k\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{\frac{2\pi i j k}{2^n}} |j\rangle = \bigotimes_{\ell=1}^{n} \frac{1}{\sqrt{2}} \Big(|0\rangle + e^{\frac{2\pi i k}{2^\ell}} |1\rangle\Big) \tag{5.1}$$

Using the convection $0.b_1 b_2 \cdots b_m = \sum_{\ell=1}^{m} b_\ell 2^{-\ell}$ for $b = b_1 b_2 \cdots b_m \in \{0,1\}^m$ (for example,

$0.101 = 1 \cdot \frac{1}{2} + 0 \cdot \frac{1}{4} + 1 \cdot \frac{1}{8} = \frac{5}{8}$), by the fact that $e^{2\pi i} = 1$ we have

$$\exp\Big(\frac{2\pi i k}{2^\ell}\Big) = \exp\Big(2\pi i \sum_{j=1}^{n} k_j 2^{n-j-\ell}\Big) = \exp\Big(2\pi i \sum_{j=n-\ell+1}^{n} k_j 2^{n-j-\ell}\Big)$$
$$= \exp\Big(2\pi i \sum_{m=1}^{\ell} k_{n-\ell+m} 2^{-m}\Big) = \exp\big(2\pi i 0.k_{n-\ell+1}k_{n-\ell+2}\cdots k_n\big)$$

so that (5.1) implies that

$$F_N|k\rangle = \bigotimes_{\ell=1}^{n} \frac{1}{\sqrt{2}}\Big(|0\rangle + e^{2\pi i 0.k_{n-\ell+1}\cdots k_n}|1\rangle\Big). \tag{5.2}$$

In the following, we will describe the efficient circuit for the $n$-qubit QFT. The elementary gates we will allow ourselves are Hadamards and controlled-$\mathrm{R}_s$ gates, where

$$\mathrm{R}_s = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^s} \end{bmatrix}.$$

Note that $\mathrm{R}_1 = \mathrm{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$, $\mathrm{R}_2 = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$, and

$$\mathrm{R}_s|k\rangle = e^{2\pi i \frac{k}{2^s}}|k_\ell\rangle \qquad \forall\, k \in \{0,1\}.$$

For large $s$, $e^{2\pi i/2^s}$ is close to 1 and hence the $\mathrm{R}_s$-gate is close to the identity-gate I. We could implement $\mathrm{R}_s$-gates using Hadamards and controlled-$\mathrm{R}_s$ gates for $s = 1, 2, 3$, but for simplicity we will just treat each $\mathrm{R}_s$ as an elementary gate.

**Example 5.2.** In this example illustrate how to construct the quantum circuit of $F_8$. Using (5.2),

$$F_8|k_1k_2k_3\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle + e^{2\pi i 0.k_3}|1\rangle\big) \otimes \frac{1}{\sqrt{2}}\big(|0\rangle + e^{2\pi i 0.k_2k_3}|1\rangle\big) \otimes \frac{1}{\sqrt{2}}\big(|0\rangle + e^{2\pi i 0.k_1k_2k_3}|1\rangle\big).$$

1. To prepare the first qubit of the desired state $F_8|k_1k_2k_3\rangle$, just apply a Hadamard to $|k_3\rangle$ since
$$\mathrm{H}|k_3\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle + (-1)^{k_3}|1\rangle\big) = \frac{1}{\sqrt{2}}\big(|0\rangle + e^{2\pi i 0.k_3}|1\rangle\big).$$
2. To prepare the second qubit of the desired state, we first apply a Hadamard to $|k_2\rangle$ to obtain $\frac{1}{\sqrt{2}}\big(|0\rangle + e^{2\pi i 0.k_2}|1\rangle\big)$, and then conditioned on $k_3$ (before we apply the Hadamard to $|k_3\rangle$) apply $\mathrm{R}_2$: by applying $\mathrm{R}_2$ it multiplies $|1\rangle$ with a phase $e^{2\pi i 0.0k_3}$, producing the correct qubit $\frac{1}{\sqrt{2}}\big(|0\rangle + e^{2\pi i 0.k_2k_3}|1\rangle\big)$.

3. To prepare the third qubit of the desired state, we apply a Hadamard to $|k_1\rangle$, apply $R_2$ conditioned on $k_2$ and $R_3$ conditioned $k_3$. This produces the correct qubit $\frac{1}{\sqrt{2}}\big(|0\rangle + e^{2\pi i 0.k_1k_2k_3}|1\rangle\big)$.

Note that the order of the output is wrong: the first qubit should be the third and vice versa. So the final step is just to swap qubits 1 and 3. Therefore, $F_8$ can be achieved by the following quantum circuit:

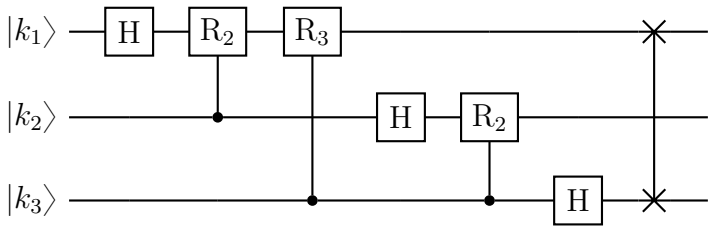



Figure 5.1: QFT for 3-qubits

The general case works analogously: starting with $\ell = 1$, we apply a Hadamard to $|k_\ell\rangle$ and then "rotate in" the additional phases required, conditioned on the values of the later bits $k_{\ell+1}, \cdots, k_n$.

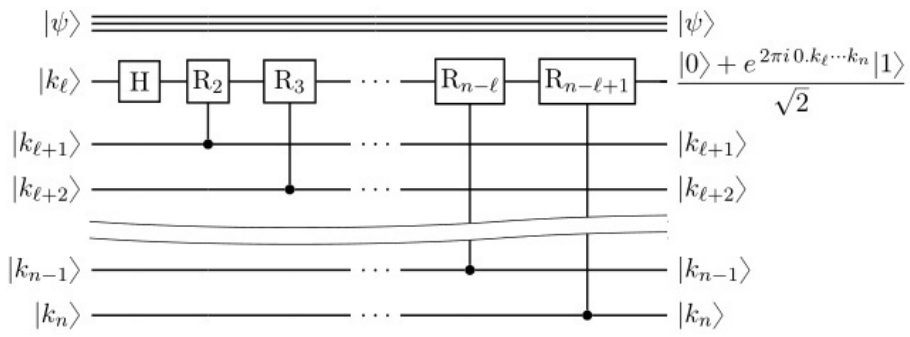



Figure 5.2: The $\ell$-th block of QFT for $n$-qubits, where $|\psi\rangle$ is a $(\ell-1)$ qubit quantum state

Some swap gates at the end then put the qubits in the right order, and we have the full quantum circuit of QFT for $n$-qubits below:

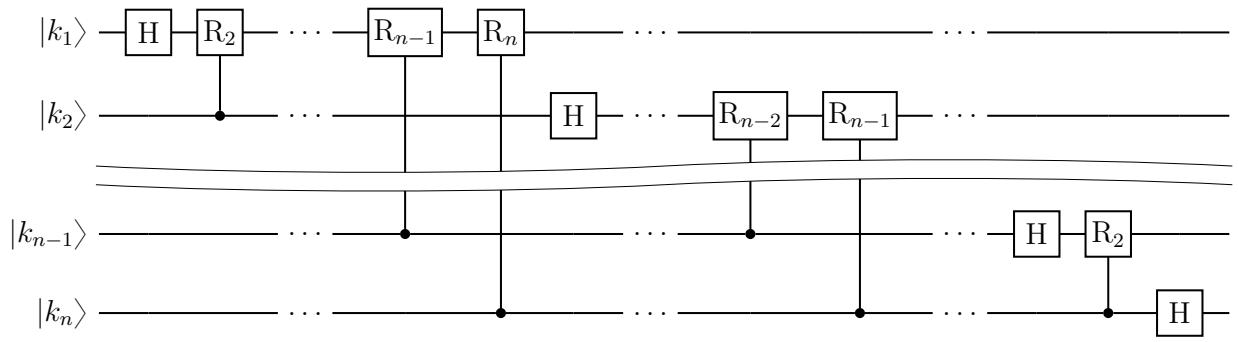



Figure 5.3: The quantum circuit of QFT for $n$-qubits (finally one should apply an order reverse operator)

Since the circuit involves $n$ qubits, and at most $n$ gates are applied to each qubit, the overall circuit uses at most $n^2$ gates. In fact, many of those gates are phase gates $R_s$ with $s \gg \log n$, which are very close to the identity and hence do not do much anyway. We can actually omit those from the circuit, keeping only $\mathcal{O}(\log n)$ gates per qubit and $\mathcal{O}(n \log n)$ gates overall. Intuitively, the overall error caused by these omissions will be small (a homework exercise asks you to make this precise). Finally, note that by inverting the circuit (that is, reversing the order of the gates and taking the adjoint $\mathrm{U}^*$ of each gate U) we obtain an equally efficient circuit for the inverse quantum Fourier transform $F_N^{-1} = F_N^*$.

## 5.5 Application: phase estimation

Suppose we can apply a unitary $U$ and we are given an eigenvector $|\psi\rangle$ of $U$ corresponding to an unknown eigenvalue $\lambda$ (that is, $U|\psi\rangle = \lambda|\psi\rangle$ for some unknown $\lambda \in \mathbb{C}$), and we would like to compute or at least approximate the $\lambda$. Since $U$ is unitary, $\lambda$ must have magnitude 1, so we can write it as $\lambda = e^{2\pi i\phi}$ for some real number $\phi \in [0, 1)$; the only thing that matters is this phase $\phi$. Suppose for simplicity that we know that $\phi = 0.\phi_1\phi_2\cdots\phi_n$ can be written exactly with $n$ bits of precision. Then here's the algorithm for phase estimation:

1. Start with $|0^n\rangle|\psi\rangle$.

2. For $N = 2^n$, apply $F_N$ to the first $n$ qubits to get $\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle|\psi\rangle$ (in fact, $\mathrm{H}^n \otimes \mathrm{I}$ would have the same effect).

3. Apply the map $|j\rangle|\psi\rangle \mapsto |j\rangle U^j|\psi\rangle$. In other words, apply $U$ to the second register for a number of times given by the first register.

4. Apply the inverse Fourier transform $F_N^{-1}$ to the first $n$ qubits and measure the result.

Note that after step 3, the first $n$ qubits are in state

$$\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i \phi j} |j\rangle \,,$$

hence the inverse quantum Fourier transform is going to give us $|2^n\phi\rangle = |\phi_1 \cdots \phi_n\rangle$ with probability 1. In case $\phi$ cannot be written exactly with $n$ bits of precision, one can show that this procedure still (with high probability) spits out a good $n$-bit approximation to $\phi$. We'll omit the calculation.

**Definition 5.3.** Let $U \in \mathrm{U}(2^m)$ be an $2^m \times 2^m$ unitary matrix and let $|\psi\rangle$ be one of the eigenvector of $U$ with corresponding eigenvalue $e^{2\pi i\theta}$. The Quantum Phase Estimation algorithm, abbreviated **QPE**, takes the inputs the $m$-qubit quantum gate for $U$ and the state $|0^n\rangle|\psi\rangle$ and returns the state $|\widetilde{\theta}\rangle|\psi\rangle$, where $\widetilde{\theta}$ denotes a binary approximation to $2^n\theta$ and the $n$ subscript denotes it has been truncated to $n$ digits. In notation, with $[\cdot]$ denoting the Gauss/floor function,

$$\mathbf{QPE}(U, |0^n\rangle|\psi\rangle) = |\widetilde{\theta}\rangle|\psi\rangle \,, \qquad \widetilde{\theta} = \left[2^n\theta\right] .$$

We will use $|\theta\rangle_n$ to denote $|\widetilde{\theta}\rangle$ if $\widetilde{\theta} = \left[2^n\theta\right]$.

# Chapter 6

# Shor's Factoring Algorithm

Suppose that $N$ is the product of two unknown prime numbers $p, q$. Then a classical way of factoring $N$ is to run a routine check to see which natural number not greater than $\sqrt{N}$ is a factor of $N$. The worse case scenario is to try this division $\sqrt{N}$ times in order to find the correct factors. The current encryption system is designed based on the fact that "it is much easier to compute the product of two prime numbers than to factor a number which is the product of two prime numbers is difficult". In the following, we quickly review the current encryption system and the mathematics behind it, and study the most famous quantum algorithm to factor large numbers, the Shor algorithm.

## 6.1 RSA Encryption

RSA is an asymmetric encryption (非對稱式加密) technique that uses two different keys as public and private keys to perform the encryption and decryption. The public key is represented by the integers $n$ and $e$, and the private key by the integer $d$. A basic principle behind RSA is to find three very large positive integers $e$, $d$, and $n$, such that with modular exponentiation all messages $m \in \mathbb{N}$ with $0 \leqslant m < n$ satisfies

$$(m^e)^d \equiv m \pmod{n}$$

and that knowing $e$ and $n$, or even $m$, it can be extremely difficult to find $d$.

### 6.1.1 Mathematical foundation

**Definition 6.1** (Greatest common divisor)**.** Let $a$ and $b$ be non-zero integers. We say the integer $d$ is the **greatest common divisor (gcd)** of $a$ and $b$, and write $d = \gcd(a, b)$, if

1. $d$ is a common divisor of $a$ and $b$.

2. every common divisor $c$ of $a$ and $b$ is not greater than $d$.

**Theorem 6.2.** *Let $a$ and $b$ be positive integers with $a \leqslant b$. Suppose that $b = aq_0 + r_1$, $a = r_1q_1 + r_2$, $r_{j-1} = r_jq_j + r_{j+1}$ for $2 \leqslant j \leqslant k$, where $0 = r_{k+1} < r_k < \cdots < r_2 < r_1 < a$ and $q_j \in \mathbb{N}$ for all $0 \leqslant j \leqslant k$.*

1. $\gcd(a, b) = r_k$, *the last non-zero remainder in the list.*

2. *If $\{s_j\}_{j=}$ and $\{t_j\}_{j=}$ are defined by*

$$s_j = \begin{cases} 1 & \text{if } j = -1\,, \\ 0 & \text{if } j = 0\,, \\ s_{j-2} - q_{j-1}s_{j-1} & \text{if } j \geqslant 1\,, \end{cases} \quad \text{and} \quad t_j = \begin{cases} 0 & \text{if } j = -1\,, \\ 1 & \text{if } j = 0\,, \\ t_{j-2} - q_{j-1}t_{j-1} & \text{if } j \geqslant 1\,, \end{cases}$$

*then*

$$at_j + bs_j = r_j \qquad \forall\, 1 \leqslant j \leqslant k\,.$$

*Proof.* Let $a$ and $b$ be positive integers with $a \leqslant b$. By the Division Algorithm, there exists positive integer $q_1$ and non-negative integer $r_1$ such that $b = aq_0 + r_1$ and $0 \leqslant r_1 < a$. If $r_1 = 0$, the lists terminate; otherwise, for $0 < r_1 < a$, there exists positive integer $q_1$ and non-negative integer $r_2$ such that $a = r_1q_1 + r_2$ and $0 \leqslant r_2 < r_1$. If $r_2 = 0$, the lists terminate; otherwise, for $0 < r_2 < r_1$, there exists positive integer $q_2$ and non-negative integer $r_3$ such that $r_1 = r_2q_2 + r_3$ and $0 \leqslant r_3 < r_2$.

Continuing in this fashion, we obtain a strictly decreasing sequence of non-negative integers $r_1, r_2, r_3, \cdots$. This lists must end, so there is an integer $k$ such that $r_{k+1} = 0$. Therefore, with $r_{-1}$ and $r_0$ denoting $b$ and $a$ respectively, we have

$$\begin{aligned} &r_{-1} \geqslant r_0 > r_1 > r_2 > \cdots > r_k > r_{k+1} = 0\,, \\ &r_{j-1} = r_jq_j + r_{j+1} \quad \text{for all } 0 \leqslant j \leqslant k\,. \end{aligned}$$

1. We now show that $r_k = d \equiv \gcd(a, b)$.

   (a) The remainder $r_k$ divides $r_{k-1}$ since $r_{k-1} = r_kq_k$. Therefore, by the fact that $r_{j-1} = r_jq_j + r_{j+1}$ for all $0 \leqslant j \leqslant k$, we find that $r_k$ divides $r_{j-1}$ for all $0 \leqslant j \leqslant k$; thus $r_k$ divides $a$ and $b$.

   (b) On the other hand, $d$ divides $r_1$ since $r_1 = b - aq_0$. Therefore, by the fact that $r_{j+1} = r_{j-1} - r_jq_j$ for all $0 \leqslant j \leqslant k$, we find that $d$ divides $r_k$ for all $0 \leqslant j \leqslant k$.

By (a), $r_k$ is a common divisor of $a$ and $b$. By (b), the greatest common divisor of $a$ and $b$ must divide $r_k$; thus we conclude that $r_k = \gcd(a, b)$.

2. To see that for all $1 \leqslant j \leqslant k$,
$$at_j + bs_j = r_j\,, \tag{6.1}$$
we note that

   (a) (6.1) holds for the case $k = 1$ since $(s_1, t_1) = (1, -q_0)$ and $b = aq_0 + r_1$.

   (b) (6.1) holds for the case $k = 2$ since $(s_2, t_2) = (-q_1, 1 + q_0q_1)$ and
$$at_2 + bs_2 = a(1 + q_0q_1) - bq_1 = a - q_1(b - aq_0) = r_0 - q_1r_1 = r_2\,.$$

   (c) Suppose that (6.1) holds for $k = \ell, \ell - 1$, $\ell \geqslant 2$. Then
$$\begin{aligned} at_{\ell+1} + bs_{\ell+1} &= a(t_{\ell-1} - q_\ell t_\ell) + b(s_{\ell-1} - q_\ell s_\ell) = at_{\ell-1} + bs_{\ell-1} - q_\ell(at_\ell + bs_\ell) \\ &= r_{\ell-1} - q_\ell r_\ell = r_{\ell+1}\,. \end{aligned}$$

   By induction, we conclude that (6.1) holds for $1 \leqslant j \leqslant k$. □

**Remark 6.3.** Let $a, b \in \mathbb{N}$ with $a \leqslant b$. The algorithm to compute $\gcd(a, b)$ given in part 1 of Theorem 6.2 is caleed **Euclid's Algorithm** (輾轉相除法), and the algorithm to compute $x, y \in \mathbb{Z}$ so that $ax + by = \gcd(a, b)$ given in part 2 of Theorem 6.2 is called **Extended Euclid's Algorithm**.

**Example 6.4.** We compute $\gcd(32, 12)$ using Euclid's algorithm as follows:
$$32 = 12 \times 2 + 8\,, \quad 12 = 8 \times 1 + 4\,, \quad 8 = 4 \times 2 + 0\,.$$
Therefore, $4 = \gcd(12, 32)$. Moreover, by working backward,
$$4 = 12 - 8 \times 1 = 12 - (32 - 12 \times 2) \times 1 = 12 \times 3 + 32 \times (-1)\,.$$
One can also obtain the "coefficients" 3 and $-1$ using Extended Euclid's Algorithm:

| $j$ | $r_j$ | $q_j$ | $s_j$ | $t_j$ |
|---|---|---|---|---|
| -1 | 32 | | 1 | 0 |
| 0 | 12 | 2 | 0 | 1 |
| 1 | 8 | 1 | 1 | $-2$ |
| 2 | 4 | 2 | $-1$ | 3 |

**Theorem 6.5.** *Let $a$ and $b$ be non-zero integers. The gcd of $a$ and $b$ is the smallest positive linear combination of $a$ and $b$; that is,*

$$\gcd(a,b) = \min\{am+bn \,|\, am+bn > 0\,, m,n \in \mathbb{Z}\}\,.$$

*Proof.* Let $d = am + bn$ be the smallest positive linear combination of $a$ and $b$. We show that $d$ satisfies (1) and (2) in the definition of the greatest common divisor.

1. **First we show that $d$ divides $a$.** By the Division Algorithm, there exist integers $q$ and $r$ such that $a = dq + r$, where $0 \leqslant r < d$. Then

   $$r = a - dq = a - (am+bn)q = a(1-m) + b(-nq)\,;$$

   thus $r$ is a linear combination of $a$ and $b$. Since $0 \leqslant r < d$ and $d$ is the smallest positive linear combination, we must have $r = 0$. Therefore, $a = dq$; thus $d$ divides $a$. Similarly, $d$ divides $b$ (replacing $a$ by $b$ in the argument above); thus $d$ is a common divisor of $a$ and $b$.

2. **Next we show that all common divisors of $a$ and $b$ is not greater than $d$.** Let $c$ be a common divisor of $a$ and $b$. Then $c$ divides $d$ since $d = am + bn$. Therefore, $c \leqslant d$.

By (1) and (2), we find that $d = \gcd(a,b)$. □

**Definition 6.6** (Euler function)**.** Let $n \in \mathbb{N}$. The function $\varphi : \mathbb{N} \to \mathbb{N}$ defined by

$$\varphi(n) = \#\{k \in \mathbb{N} \,|\, 1 \leqslant k \leqslant n \text{ and } \gcd(k,n) = 1\}$$

is called the Euler (phi) function. In other words, the Euler function counts the positive integers up to a given integer $n$ that are coprime to $n$.

**Proposition 6.7.** *For each $n \in \mathbb{N}$,*

$$\varphi(n) = n \prod_{p|n} \left(1 - \frac{1}{p}\right).$$

*In particular, by writing $n = \prod_{j=1}^{r} p_j^{k_j} = p_1^{k_1} p_2^{k_2} \cdots p_r^{k_r}$, where $p_1, \cdots, p_r$ are distinct prime numbers and $k_1, \cdots, k_r \in \mathbb{N}$,*

$$\varphi(n) = \prod_{j=1}^{r} p_j^{k_j - 1}(p_j - 1)\,.$$

**Corollary 6.8.** *Let $m, n \in \mathbb{N}$ be such that* $\gcd(m, n) = 1$. *Then* $\varphi(mn) = \varphi(m)\varphi(n)$.

**Definition 6.9.** Given $a \in \mathbb{Z}$ and $n \in \mathbb{N}$, $a$ modulo $n$ (abbreviated as $a \bmod n$) is the remainder of the Euclidean division of $a$ by $n$, where $a$ is the dividend and $n$ is the divisor. In other words, $a \bmod n$ outputs $r$ if $a = qn + r$ for some $q \in \mathbb{Z}$ and $r \in \{0, 1, \cdots, n-1\}$. For $a, b \in \mathbb{Z}$, the notation $a \equiv b \pmod{n}$ denotes the fact that $n|(a-b)$; that is, there exists $m \in \mathbb{Z}$ such that $a - b = mn$.

We note that by the definition, $a \equiv b \pmod{n}$ if and only if $a - b \equiv 0 \pmod{n}$.

**Definition 6.10.** The addition $\oplus$ on $\mathbb{Z}_n$ is defined by

$$c = a \oplus b \qquad \text{if and only if} \qquad (a + b) \bmod n \text{ outputs } c\,,$$

and the multiplication $\odot$ on $\mathbb{Z}_n$ is defined by

$$c = a \odot b \qquad \text{if and only if} \qquad (a \cdot b) \bmod n \text{ outputs } c\,,$$

where $+$ and $\cdot$ are the usual addition and multiplication on $\mathbb{Z}$.

**Proposition 6.11.** $(\mathbb{Z}_n, \oplus)$ *is a group; that is,*

1. $\mathbb{Z}_n$ *is closed under addition* $\oplus$*;*
2. *there exists an additive identity* $0$ *(that is,* $a \oplus 0 = a$ *for all* $a \in \mathbb{Z}_n$*), and*
3. *every element in* $\mathbb{Z}_n$ *has an additive inverse (that is, for each* $a \in \mathbb{Z}_n$ *there exists* $b \in \mathbb{Z}_n$ *such that* $a \oplus b = 0$*).*

**Proposition 6.12.** *Let* $n \geqslant 2$ *be an integer, and* $a, b \in \mathbb{Z}$ *satisfy* $a \equiv b \pmod{n}$. *Then* $\gcd(a, n) = 1$ *if and only if* $\gcd(b, n) = 1$.

*Proof.* It suffices to shows that if $\gcd(a, n) \neq 1$, then $\gcd(b, n) \neq 1$.

Suppose that $\gcd(a, n) = p > 1$. Then $a = pq_1$ and $n = pq_2$ for some $q_1, q_2 \in \mathbb{Z}$. Since $a \equiv b \pmod{n}$, there exists $m \in \mathbb{Z}$ such that $a - b = mn$. Therefore, $b = a - mn = pq_1 - pq_2m = p(q_1 - q_2m)$ which shows that $\gcd(b, n) \geqslant p$. ▫

Proposition 6.12 shows that if $a \in \mathbb{Z}$ satisfies $\gcd(a, n) = 1$, then $(a \bmod n)$ is coprime to $n$.

**Proposition 6.13.** *Let $a, b, c, d \in \mathbb{Z}$ and $n \in \mathbb{N}$ be such that $a \equiv c \ (mod\ n)$ and $b \equiv d \ (mod\ n)$. Then $a \cdot b \equiv c \cdot d \ (mod\ n)$.*

**Proposition 6.14** (Cancellation law in $\mathbb{Z}_n$)**.** *Let $a, n \in \mathbb{N}$ be such that $\gcd(a, n) = 1$. If $a \cdot b \equiv a \cdot c \ (mod\ n)$, then $b \equiv c \ (mod\ n)$.*

**Theorem 6.15.** *The integers coprime to $n$ from the set $\{0, 1, \cdots, n-1\}$ of $n$ non-negative integers form a group under multiplication modulo $n$. In other words, let $S$ be a subset of $\mathbb{Z}_n$ consisting of numbers coprime to $n$; that is, $S = \left\{k \in \mathbb{N} \,\middle|\, 1 \leqslant k \leqslant n \text{ and } \gcd(k, n) = 1\right\}$. Then $(S, \odot)$ is a group; that is,*

1. *$S$ is closed under multiplication $\odot$;*
2. *there exists an multiplicative identity 1 (that is, $a \odot 1 = a$ for all $a \in S$), and*
3. *every element in $S$ has an multiplicative inverse element (that is, for each $a \in S$ there exists $b \in S$ such that $a \odot b = 1$).*

*Proof.* It suffices to prove 1 and 3.

1. Let $a, b \in S$. Then $a \cdot b$ is coprime to $n$; thus Proposition 6.12 implies that $a \cdot b \bmod n$ is coprime to $n$ as well. Therefore, $a \odot b \in S$.
3. Let $a \in S$. Then the set $a \odot S \equiv \left\{a \odot s \,\middle|\, s \in S\right\}$ is a subset of $S$. Moreover, if $s_1, s_2 \in S$ satisfying that $a \odot s_1 = a \odot s_2$; that is, $a \cdot s_1 \equiv a \cdot s_2 \pmod{n}$, then $s_1 = s_2$; thus $\#(a \odot S) = \varphi(n)$. This fact shows that there exists $s \in S$ such that $a \odot s = 1$. ▫

**Definition 6.16.** The multiplicative group of integers modulo $n$ (given in Theorem 6.15) is denoted by $(\mathbb{Z}_n^*, \odot)$.

**Theorem 6.17.** *Let $n \in \mathbb{N}$ and $a \in \mathbb{Z}_n^*$. If $a \cdot x + n \cdot y = 1$ for some $x, y \in \mathbb{Z}$, then*

$$a^{-1} \equiv x \ (mod\ n)\,,$$

*where $a^{-1}$ denotes the number in $\mathbb{Z}_n^*$ satisfying $a \odot a^{-1} = a^{-1} \odot a = 1$.*

**Theorem 6.18** (Euler)**.** *Let $a, n \in \mathbb{N}$ be such that $\gcd(a, n) = 1$. Then $a^{\varphi(n)} \equiv 1 \pmod{n}$.*

*Proof.* Let $a\mathbb{Z}_n^*$ be the set $a\mathbb{Z}_n^* \equiv \{a \cdot s \,|\, s \in \mathbb{Z}_n^*\}$. Then the set $(a\mathbb{Z}_n^* \bmod n) \equiv \{(a \cdot s) \bmod n \,|\, s \in \mathbb{Z}_n^*\}$ is identical to $\mathbb{Z}_n^*$. Therefore, Proposition 6.13 implies that

$$\prod_{k \in \mathbb{Z}_n^*} k \equiv \prod_{k \in a\mathbb{Z}_n^*} k \pmod{n}.$$

Since $\prod_{k \in a\mathbb{Z}_n^*} k = a^{\varphi(n)} \prod_{k \in \mathbb{Z}_n^*} k$ and $\prod_{k \in \mathbb{Z}_n^*} k$ is coprime to $n$, by the cancellation law for $\mathbb{Z}_n$ (Proposition 6.14) we find that $a^{\varphi(n)} \equiv 1 \pmod{n}$. □

**Corollary 6.19** (Fermat little theorem)**.** *Let $p$ be a prime number, and $a \in \mathbb{N}$ satisfy* $\gcd(a, p) = 1$. *Then* $a^{p-1} \equiv 1 \pmod{p}$.

### 6.1.2 Encryption based on factoring large numbers

The RSA algorithm involves four steps: key generation, key distribution, encryption, and decryption.

#### Key generation

The keys for the RSA algorithm are generated in the following way:

1. Choose two distinct prime numbers $p$ and $q$.
   (a) For security purposes, the integers $p$ and $q$ should be chosen at random and should be similar in magnitude but differ in length by a few digits to make factoring harder. Prime integers can be efficiently found using a primality test.
   (b) $p$ and $q$ are kept secret.
2. Compute $n = pq$.
   (a) $n$ is used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the key length.
   (b) $n$ is released as part of the public key.
3. Compute $\varphi(n)$, where $\varphi$ is the Euler function. By Proposition 6.7, $\varphi(n) = (p-1)(q-1)$. $\varphi(n)$ is kept secret.
4. Choose an integer $e$ such that $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$; that is, $e$ and $\varphi(n)$ are coprime.

(a) $e$ having a short bit-length and small Hamming weight results in more efficient encryption - the most commonly chosen value for $e$ is $2^{16} + 1 = 65537$. The smallest (and fastest) possible value for $e$ is 3, but such a small value for e has been shown to be less secure in some settings.

(b) $e$ is released as part of the public key.

5. Determine $d$ as $d \equiv e^{-1} \pmod{\varphi(n)}$; that is, $d$ is the modular multiplicative inverse of $e$ modulo $\varphi(n)$.

   (a) This means: solve for $d$ the equation $d \cdot e \equiv 1 \pmod{\varphi(n)}$; $d$ can be computed efficiently by using the extended Euclidean algorithm, since, thanks to $e$ and $\varphi(n)$ being coprime, said equation is a form of Bézout's identity, where $d$ is one of the coefficients.

   (b) $d$ is kept secret as the private key exponent.

The public key consists of the modulus $n$ and the public (or encryption) exponent $e$. The private key consists of the private (or decryption) exponent $d$, which must be kept secret. $p$, $q$, and $\varphi(n)$ must also be kept secret because they can be used to calculate $d$. In fact, they can all be discarded after $d$ has been computed.

Note: The authors of the original RSA paper carry out the key generation by choosing $d$ and then computing $e$ as the modular multiplicative inverse of $d$ modulo $\varphi(n)$, whereas most current implementations of RSA, such as those following PKCS#1, do the reverse (choose $e$ and compute $d$). Since the chosen key can be small, whereas the computed key normally is not, the RSA paper's algorithm optimizes decryption compared to encryption, while the modern algorithm optimizes encryption instead.

**Remark 6.20.** In modern RSA implementation the use of Euler function $\varphi$ is replaced by Carmichael's totient function $\lambda$ defined by

$$\lambda(n) = \min\left\{k \in \mathbb{N} \,\middle|\, a^k \equiv 1 \pmod{n} \text{ for all } a \in \mathbb{Z}_n^*\right\}.$$

If $n = pq$ with prime numbers $p$ and $q$, then $\lambda(n) = \operatorname{lcm}(p-1, q-1)$, the least common multiple of $p-1$ and $q-1$.

**Remark 6.21.** If both $n$ and $\varphi(n)$ are known, then two primes $p$ and $q$ satisfying

$$n = pq\,, \varphi(n) = (p-1)(q-1)$$

can be solved easily since $p$ and $q$ are zeros of $x^2 + \big[\varphi(n) - (n+1)\big]x + n = 0$.

### Key distribution

Suppose that Bob wants to send information to Alice. If they decide to use RSA, Bob must know Alice's public key to encrypt the message, and Alice must use her private key to decrypt the message. To enable Bob to send his encrypted messages, Alice transmits her public key $(n, e)$ to Bob via a reliable, but not necessarily secret, route. Alice's private key $(d)$ is never distributed.

### Encryption

After Bob obtains Alice's public key, he can send a message $M$ to Alice.

To do it, he first turns $M$ (strictly speaking, the un-padded plaintext) into an integer $m$ (strictly speaking, the padded plaintext), such that $0 \leqslant m < n$ by using an agreed-upon reversible protocol known as a padding scheme. He then computes the ciphertext $c$, using Alice's public key $e$, corresponding to

$$c \equiv m^e \pmod{n}.$$

This can be done reasonably quickly, even for very large numbers, using modular exponentiation. Bob then transmits $c$ to Alice. Note that at least nine values of $m$ will yield a ciphertext $c$ equal to $m$, but this is very unlikely to occur in practice.

### Decryption

Alice can recover $m$ from $c$ by using her private key exponent $d$ by computing

$$c^d \equiv (m^e)^d \equiv m \pmod{n}.$$

Given $m$, she can recover the original message $M$ by reversing the padding scheme.

**Example 6.22.** Here is an toy example of RSA encryption and decryption.

1. Choose two prime numbers $p = 11$ and $q = 31$.
2. Compute $n = pq = 341$.
3. Compute $\varphi(n) = (p-1)(q-1) = 300$ / $(\lambda(n) = \operatorname{lcm}(10, 30) = 30)$.
4. Choose the encryption key $e = 17$ so that $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$ / $(1 < e < \lambda(n)$ and $\gcd(e, \lambda(n)) = 1)$.

5. Compute the decryption key $d$ by Euclid's algorithm (and Theorem 6.17):

$$\begin{aligned} 300 &= 17 \times 17 + 11 & 30 &= 17 \times 1 + 13 \\ 17 &= 11 \times 1 + 6 & 17 &= 13 \times 1 + 4 \\ 11 &= 6 \times 1 + 5 & 13 &= 4 \times 3 + 1 \\ 6 &= 5 \times 1 + 1 & 4 &= 1 \times 4 \\ 5 &= 1 \times 5 + 0 \end{aligned}$$

which implies that $300 \times (-3) + 17 \times 53 = 1$ ($30 \times 4 + 17 \times (-7) = 1$); thus $d = 53$ ($d \equiv -7 \pmod{30}$ or $d = 23$).

| $j$ | $r_j$ | $q_j$ | $s_j$ | $t_j$ |
|---|---|---|---|---|
| -1 | 300 | | 1 | 0 |
| 0 | 17 | 17 | 0 | 1 |
| 1 | 11 | 1 | 1 | $-17$ |
| 2 | 6 | 1 | $-1$ | 18 |
| 3 | 5 | 1 | 2 | $-35$ |
| 4 | 1 | 5 | $-3$ | 53 |

| $j$ | $r_j$ | $q_j$ | $s_j$ | $t_j$ |
|---|---|---|---|---|
| -1 | 30 | | 1 | 0 |
| 0 | 17 | 1 | 0 | 1 |
| 1 | 13 | 1 | 1 | $-1$ |
| 2 | 4 | 3 | $-1$ | 2 |
| 3 | 1 | 4 | 4 | $-7$ |

Therefore, to encrypt $m = 30$, we raise to the power of 17 and obtain the encrypted message:

$$30^{17} \equiv 123 \pmod{341}.$$

To decrypt the encrypted message, we raise it to the power of 53 (23) and obtain that

$$123^{53} \equiv (123^3)^{17} \cdot 123^2 \equiv 30^{17} \cdot 125 \equiv 123 \cdot 125 \equiv 30 \pmod{341}$$

$$(123^{23} \equiv (123^3)^7 \cdot 123^2 \equiv 30^7 \cdot 125 \equiv 123 \cdot 125 \equiv 30 \pmod{341}).$$

## 6.2 Reduction from Factoring to Period-finding

The crucial observation of Shor was that there is an efficient quantum algorithm for the problem of period-finding and that factoring can be reduced to this, in the sense that an efficient algorithm for period-finding implies an efficient algorithm for factoring. We first explain the reduction. Suppose we want to find factors of the composite number $N > 1$. We may assume $N$ is odd and not a prime power, since those cases can easily be filtered out by a classical algorithm. Now randomly choose some integer $x \in \{2, \cdots, N-1\}$ which

is coprime to $N$. If $x$ is not coprime to $N$, then the greatest common divisor of $x$ and $N$ is a nontrivial factor of $N$, so then we are already done. From now on consider $x$ and $N$ are coprime, so $x$ is an element of the multiplicative group $\mathbb{Z}_N^*$. Consider the sequence

$$1 = x^0 \bmod N\,, \quad x^1 \bmod N\,, \quad x^2 \bmod N\,, \cdots$$

This sequence will cycle after a while: there is a least $0 < r \leqslant N$ such that $x^r \equiv 1 \pmod{N}$. The smallest such number $r$ is called the period of the sequence (a.k.a. the order of the element $x$ in the group $(\mathbb{Z}_N^*, \odot)$). If $r$ is even, then

$$\begin{aligned} x^r \equiv 1 \pmod{N} &\Leftrightarrow (x^{r/2})^2 \equiv 1 \pmod{N} \Leftrightarrow (x^{r/2}+1)(x^{r/2}-1) \equiv 0 \pmod{N} \\ &\Leftrightarrow (x^{r/2}+1)(x^{r/2}-1) = kN \text{ for some } k \in \mathbb{N}. \end{aligned}$$

Because both $x^{r/2}+1 > 0$ and $x^{r/2}-1 > 0$ (due to the fact that $x > 1$), we must have $k \neq 0$. Hence $x^{r/2}+1$ or $x^{r/2}-1$ will share a factor with $N$. Note that $x^{r/2} \neq 1 \bmod N$ for otherwise $r/2$ is a period of $f$, a contradict to the assumption that $r$ is the smallest period. In other words, $\gcd(x^{r/2}-1, N) \neq N$. It is still possible that $\gcd(x^{r/2}-1, N) = 1$ and this is equivalent to that $\gcd(x^{r/2}+1, N) = N$. Therefore, we are able to factor $N$ if $\gcd(x^{r/2}+1, N) < N$.

Assuming that $N$ is odd and not a prime power, it can be shown (in Theorem 6.37) that with probability not less than $1/2$, the period $r$ is even and $x^{r/2}+1$ and $x^{r/2}-1$ are not multiples of $N$.

Accordingly, with high probability we can obtain an even period $r$ so that $\gcd(x^{r/2}+1, N)$ is a non-trivial factor of $N$. If we are unlucky we might have chosen an $x$ that does not give a factor (which we can detect efficiently), but trying a few different random $x$ gives a high probability of finding a factor.

The factorization algorithm above is summarized as follows. Let $N$ be an odd natural number N that has at least two distinct prime factors.

**Step 1**: Choose $x \in \{2, \cdots, N-1\}$ and compute $\gcd(x, N)$.

(a) If $\gcd(x, N) > 1$, then $\gcd(x, N)$ is a non-trivial factor of $N$ and we are done.

(b) If $\gcd(x, N) = 1$, then goto **Step 2**.

**Step 2**: Determine the period $r$ of the function $f(a) = x^a \bmod N$.

(a) If $r$ is odd, goto **Step 1**.

(b) If $r$ is even, goto **Step 3**.

**Step 3**: Determine $\gcd(x^{r/2}+1, N)$.

(a) If $\gcd(x^{r/2}+1, N) = N$, then goto **Step 1**.

(b) If $\gcd(x^{r/2}+1, N) < N$, then $\gcd(x^{r/2}+1, N)$ is a non-trivial factor of $N$ and we are done.

Thus the problem of factoring reduces to finding the period $r$ of the function given by modular exponentiation $f(a) = x^a \bmod N$. In general, the period-finding problem can be stated as follows:

**The period-finding problem**: We are given some function $f : \mathbb{N} \to \{0, 1, \cdots, N-1\}$ with the property that there is some unknown $r \in \{0, 1, \cdots, N-1\}$ such that $f(a) = f(b)$ if and only if $a \equiv b \bmod r$. The goal is to find $r$.

A naive algorithm to find the period is to compute $f(0)$, $f(1)$, $f(2)$, $\cdots$ until we encounter the value $f(0)$ for the second time. The input at which this happens is the period $r$ that we are trying to find. The problem with this approach is that $r$ could be huge, polynomial in $N$. To be efficient, we would like a runtime that is polynomial in $\log N$, since that is the bitsize of the inputs to $f$. It is generally believed that classical computers cannot solve period-finding problems efficiently. We will show below how we can solve this problem efficiently on a quantum computer, using only $\mathcal{O}(\log \log N)$ evaluations of $f$ and $\mathcal{O}(\log \log N)$ quantum Fourier transforms. An evaluation of $f$ can be viewed as analogous to the application of a query in the previous algorithms. Even a somewhat more general kind of period-finding problems can be solved by Shor's algorithm with very few $f$-evaluations, whereas any classical bounded-error algorithm would need to evaluate the function $\Omega(N^{1/3}/\sqrt{\log N})$ times in order to find the period.

## 6.3 Shor's Period-finding Algorithm

Before proceeding to the discussion of Shor's algorithm, let us point out that the period-finding problem in the previous section can be related to the phase estimation problem in the following sense: given $x \in \mathbb{Z}_N^*$, the (unitary) map

$$U|y\rangle = |x \odot y\rangle \equiv |x \cdot y \bmod N\rangle$$

has an eigenvector

$$|\psi_s\rangle \equiv \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left(-\frac{2\pi i s k}{r}\right) |x^k \bmod N\rangle$$

for each $0 \leqslant s < r$ since

$$\begin{aligned} U|\psi_s\rangle &= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left(-\frac{2\pi i s k}{r}\right) U|x^k \bmod N\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left(-\frac{2\pi i s k}{r}\right) |x^{k+1} \bmod N\rangle \\ &= \frac{1}{\sqrt{r}} \exp\left(\frac{2\pi i s}{r}\right) \sum_{k=0}^{r-1} \exp\left(-\frac{2\pi i s (k+1)}{r}\right) |x^{k+1} \bmod N\rangle \\ &= \frac{1}{\sqrt{r}} \exp\left(\frac{2\pi i s}{r}\right) \sum_{\ell=1}^{r} \exp\left(-\frac{2\pi i s \ell}{r}\right) |x^{\ell} \bmod N\rangle \\ &= \frac{1}{\sqrt{r}} \exp\left(\frac{2\pi i s}{r}\right) \sum_{\ell=0}^{r-1} \exp\left(-\frac{2\pi i s \ell}{r}\right) |x^{\ell} \bmod N\rangle = \exp\left(\frac{2\pi i s}{r}\right) |\psi_s\rangle . \end{aligned}$$

Therefore, the phase estimation algorithm introduced in Section 5.5 can be applied to find $r$ as long as the eigenvector $|\psi_s\rangle$ is known (even though we do not know $|\psi_s\rangle$ for $s \neq 0$).

Now we will show how Shor's algorithm finds the period $r$ of the function $f$, given a "black-box" that maps $|a\rangle|0^K\rangle \mapsto |a\rangle|f(a)\rangle$. We can always efficiently pick some $q = 2^L$ such that $N^2 < q \leqslant 2N^2$. Then we can implement the Fourier transform $\mathrm{QFT}_q$ using $\mathcal{O}((\log N)^2)$ gates. Let $\mathrm{O}_f$ denote the unitary that maps $|a\rangle|0^K\rangle \mapsto |a\rangle|f(a)\rangle$, where the first register consists of $L$ qubits, and the second of $K = [\log_2 N] + 1$ qubits.

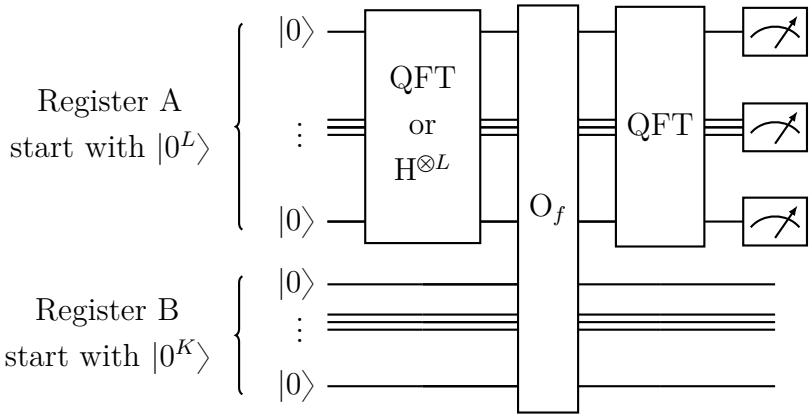



Figure 6.1: Shor's period-finding algorithm

Shor's period-finding algorithm is illustrated in Figure 6.1. Start with $|\psi_0\rangle = |0^L\rangle|0^K\rangle$.

Apply the QFT (or just $L$ Hadamard gates) to the first register to build the uniform superposition

$$|\psi_1\rangle = (\mathrm{H}^{\otimes L} \otimes \mathrm{I}_K)|\psi_0\rangle = \frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle|0^K\rangle\,,$$

where $\mathrm{I}_K$ denotes the identity map on the second register. The second register still consists of zeroes. Now use the "black-box" to compute $f(a)$ in quantum parallel:

$$|\psi_2\rangle = \mathrm{O}_f|\psi_1\rangle = \frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle|f(a)\rangle\,.$$

Next we apply the quantum Fourier transform **QFT** to the first register to obtain the quantum state $|\psi_3\rangle = (F_q \otimes \mathrm{I}_K)|\psi_2\rangle$. Finally, we measure the first register and obtain a number $b$ and wish to find the period of $f$ based on this observation.

Some of the measurement $b$ obtained by Shor's algorithm above are useless. The measurement $b$ becomes useful for us to determine the period $r$ if $b$ belongs to the set

$$E = \left\{ b \in \mathbb{N} \cup \{0\} \,\middle|\, 0 \leqslant b \leqslant q-1 \text{ and } \left|\frac{b}{q} - \frac{c}{r}\right| < \frac{1}{2r^2} \text{ for some } c \in \mathbb{Z}_r^* \right\},$$

where we recall that $\mathbb{Z}_r^*$ is the collection of numbers from $\{1, \cdots, r-1\}$ that are coprime to $r$ so that $\#\mathbb{Z}_r^* = \varphi(r)$. We note that $E$ is indeed unknown to us (since $r$ is unknown to us) but it exists and is a non-empty set. We will show in Section 6.5 that the probability of obtaining $b \in E$ by Shor's algorithm is not less than $\dfrac{1}{10 \ln L}$. This implies that if we apply Shor's algorithm $k$ times, the probability of obtaining **no** $b \in E$ is less than $\left(1 - \dfrac{1}{10 \ln L}\right)^k$ which is quite small when $k$ is large.

Suppose that we apply Shor's algorithm and obtain one such $b \in E$. Then there exists $c \in \mathbb{Z}_r^*$ such that $\left|\dfrac{b}{q} - \dfrac{c}{r}\right| < \dfrac{1}{2r^2}$. We note that in this inequality we only know $b$ and $q$ (so is the number $x = b/q$), but both $c$ and $r$ are unknown to us. Even though $c$ and $r$ are unknown to us, the fact that $c \in \mathbb{Z}_r^*$ implies that $\dfrac{c}{r}$ is an irreducible fraction（最簡分數）. Therefore, if there is a fast algorithm to find all irreducible fractions $\dfrac{n}{m}$ satisfying

$$\left| x - \frac{n}{m} \right| \leqslant \frac{1}{2m^2} \qquad \text{and} \qquad m < N\,, \tag{6.2}$$

we can check whether the denominators $m$ of all such irreducible fractions satisfying $m < N$ is the period of $f$. In Section 6.4 an efficient algorithm based on continuous fractions is proposed to find all irreducible fractions $\dfrac{n}{m}$ satisfying (6.2).

**Shor's period-finding algorithm**: Let $f : \mathbb{N} \cup \{0\} \to \mathbb{N} \cup \{0\}$ be a periodic function with period $r$ satisfying $19 \leqslant r < 2^{L/2}$ for some $L \in \mathbb{N}$ such that $f$ is injective within one period and is bounded by $2^K - 1$.

**Step 1**: Prepare the oracle $U_f$ (or $\mathrm{O}_f$) satisfying

$$U_f|a\rangle|b\rangle = |a\rangle|b \oplus f(a)\rangle \qquad \forall\, a \in \{0,1\}^L, b \in \{0,1\}^K\,.$$

**Step 2**: Measure the first register of the quantum state

$$(F_{2^L} \otimes \mathrm{I}_K) U_f (\mathrm{H}^{\otimes L} \otimes \mathrm{I}_K)(|0^L\rangle \otimes |0^K\rangle)\,.$$

and obtain $b$.

**Step 3**: Use continuous fraction algorithm to find all irreducible fractions $\dfrac{n}{m}$ satisfying

$$\Big|\frac{b}{2^L} - \frac{n}{m}\Big| < \frac{1}{2m^2} \qquad \text{and} \qquad m < 2^{L/2}\,.$$

(a) If one of such denominator $m$ is the period of $f$, we are done.

(b) If none of these denominators $m$ is the period of $f$, then $b \notin E$ and goto **Step 1**.

## 6.4 Continued fractions

A continued fraction, or simply CF, is a real number of the form

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\cdots}}}\,.$$

The continued fraction above is denote by $[a_0; a_1, a_2, \cdots]$ (here the number of $a_i$'s can be finite or infinite), and the $a_i$'s are called the partial quotients. We assume these to be positive natural numbers. $[a_0; \cdots, a_n]$ is the $n$-th convergent of the continued fraction $[a_0; a_1, a_2, \cdots]$, and can be simply computed by the following iterative scheme: if

$$\begin{aligned} &p_0 = a_0, \quad p_1 = a_1 a_0 + 1, \quad p_n = a_n p_{n-1} + p_{n-2}\,, \\ &q_0 = 1, \quad\ q_1 = a_1, \qquad\quad\ q_n = a_n q_{n-1} + q_{n-2}\,. \end{aligned} \tag{6.3}$$

then $[a_0; \cdots, a_n]$, in its lowest terms, is $\dfrac{p_n}{q_n}$; that is,

$$[a_0; \cdots, a_n] = \frac{p_n}{q_n}\,, \qquad \gcd(p_n, q_n) = 1\,.$$

Note that $q_n$ increases at least exponentially with $n$ since $q_n \geqslant 2q_{n-2}$. Given a real number $x$, the following "algorithm" gives a continued fraction expansion of $x$:

$$\begin{aligned} a_0 &\equiv [x]\,, & x_1 &\equiv 1/(x-a_0)\,,\\ a_1 &\equiv [x_1]\,, & x_2 &\equiv 1/(x_1-a_1)\,,\\ a_2 &\equiv [x_2]\,, & x_3 &\equiv 1/(x_2-a_2)\,,\\ &\vdots \end{aligned}$$

Informally, we just take the integer part of the number as the partial quotient and continue with the inverse of the decimal part of the number.

**Example 6.23.** Let $x = \sqrt{2}$. Then $a_0 = 1$ and $a_k = 2$ for all $k \in \mathbb{N}$. To see this, we note that $x_1 = \dfrac{1}{\sqrt{2}-1} = \sqrt{2}+1$ so we have $a_1 = 2$. This then shows that

$$x_2 = \frac{1}{x_1 - a_1} = \frac{1}{\sqrt{2}+1-2} = \sqrt{2}+1$$

and as a consequence $a_2 = 2$. Repeating this process, we find that $x_k = \sqrt{2}+1$ and $a_k = 2$ for all $k \in \mathbb{N}$.

Using (6.3), we obtain that

| $n$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $p_n$ | 3 | 7 | 17 | 41 | 99 | 239 | 577 | 1393 | 3363 |
| $q_n$ | 2 | 5 | 12 | 29 | 70 | 169 | 408 | 985 | 2378 |
| $\left\lvert x - \frac{p_n}{q_n}\right\rvert$ | 0.0858 | 0.0142 | 0.0025 | 4.2e-4 | 7.2e-5 | 1.2e-5 | 2.1e-6 | 3.6e-7 | 6.3e-8 |

**Theorem 6.24.** *For an $x \in \mathbb{R}$, the sequence $\{a_j\}$ constructed from the algorithm above terminates if and only if $x$ is rational.*

The convergence of the CF approximate $x$ follows from the fact that

$$\text{if } x = [a_0; \cdots, a_n], \text{ then } \left| x - \frac{p_n}{q_n}\right| \leqslant \frac{1}{q_n^2}\,.$$

Recall that $q_n$ increases exponentially with $n$, so this convergence is quite fast. Moreover, $p_n/q_n$ provides the best approximation of $x$ among all fractions with denominator not greater than $q_n$:

$$\text{if } n \geqslant 1,\ q \leqslant q_n,\ \frac{p}{q} \neq \frac{p_n}{q_n}, \text{ then } \left| x - \frac{p_n}{q_n}\right| < \left| x - \frac{p}{q}\right|.$$

The following theorem shows how one accomplish **Step 3** in Shor's period-finding algorithm.

**Theorem 6.25.** *Let $b, q \in \mathbb{N}$ be given and let $[a_0; a_1, \cdots, a_n]$ be the continued fraction of their quotient; that is*

$$\frac{b}{q} = [a_0; a_1, \cdots, a_n] .$$

*If $c, r \in \mathbb{N}$ are such that*

$$\left|\frac{b}{q} - \frac{c}{r}\right| < \frac{1}{2r^2} ,$$

*then $\frac{c}{r}$ is a convergent of the continued fraction of $\frac{b}{q}$; that is, there exists a $j \in \{0, 1, \cdots, n\}$ such that*

$$\frac{c}{r} = [a_0; a_1, \cdots, a_j] = \frac{p_j}{q_j}$$

*where $p_j$ and $q_j$ are as constructed by* (6.3).

## 6.5 Efficiency of Shor's Algorithm

### 6.5.1 Shor's period-finding algorithm

Shor's algorithm can be applied to find the period of a more general class of periodic functions.

**Theorem 6.26.** *Let $f : \mathbb{N} \cup \{0\} \to \mathbb{N} \cup \{0\}$ be a periodic function with period $r$ satisfying $19 \leqslant r < 2^{L/2}$ for some $L \in \mathbb{N}$ such that $f$ is injective within one period and is bounded by $2^K - 1$, and $U_f$ be an $(L + K)$-qubit quantum gate satisfying*

$$U_f|a\rangle|b\rangle = |a\rangle|b \oplus f(a)\rangle , \qquad \forall\, a \in \{0,1\}^L, b \in \{0,1\}^K .$$

*Then each application of Shor's algorithm provides the period $r$ with a probability of at least $\frac{1}{10 \ln L}$.*

*Proof.* Let $M = \max\left\{ f(a) \,\middle|\, 0 \leqslant a \leqslant 2^L - 1 \right\}$ and $K \in \mathbb{N}$ with $M < 2^K$, and $\mathbb{H}$ be the usual qubit Hilbert space with basis $\{|0\rangle, |1\rangle\}$. Set $|\psi_0\rangle = |0^L\rangle \otimes |0^K\rangle$. Then with $\mathrm{I}_K$ denoting the identity map on $\mathbb{H}^{\otimes K}$,

$$|\psi_1\rangle \equiv (\mathrm{H}^{\otimes L} \otimes \mathrm{I}_K)|\psi_0\rangle = \frac{1}{\sqrt{2^L}} \sum_{a=0}^{2^L-1} |a\rangle \otimes |0^K\rangle .$$

Applying $U_f$ to $|\psi_1\rangle$, we find that

$$|\psi_2\rangle = U_f|\psi_1\rangle = \frac{1}{\sqrt{2^L}} \sum_{a=0}^{2^L-1} |a\rangle \otimes |f(a)\rangle .$$

Define $m \equiv \big[\frac{2^L-1}{r}\big]$, the largest integer smaller than $\frac{2^L-1}{r}$, and $R \equiv (2^L-1) \bmod r$. Then

$$|\psi_2\rangle = \frac{1}{\sqrt{2^L}} \sum_{j=0}^{m-1} \sum_{s=0}^{r-1} |jr+s\rangle \otimes |f(s)\rangle + \sum_{s=0}^{R} |mr+s\rangle \otimes |f(s)\rangle \,.$$

Define $m_s = m - \mathbf{1}_{(R,\infty)}(s)$; that is, $m_s = m$ if $s \leqslant R$ and $m_s = m-1$ if $s > R$. Then

$$|\psi_2\rangle = \frac{1}{\sqrt{2^L}} \sum_{s=0}^{r-1} \sum_{j=0}^{m_s} |jr+s\rangle \otimes |f(s)\rangle \,.$$

Next we apply the quantum Fourier transform to the first $L$ qubits of $|\psi_2\rangle$ and obtain that

$$\begin{aligned} |\psi_3\rangle \equiv (F_{2^L} \otimes \mathrm{I}_B)|\psi_2\rangle &= \frac{1}{\sqrt{2^L}} \sum_{s=0}^{r-1} \sum_{j=0}^{m_s} (F_{2^L}|jr+s\rangle) \otimes |f(s)\rangle \\ &= \frac{1}{2^L} \sum_{s=0}^{r-1} \sum_{j=0}^{m_s} \sum_{b=0}^{2^L-1} \exp\Big(2\pi i \frac{(jr+s)b}{2^L}\Big) |b\rangle \otimes |f(s)\rangle \end{aligned}$$

Now we measure the input register, and let $P(b)$ denote the probability of observing $|b\rangle$ upon measurement. Let $E$ be the collection of $b \in \{0,1,\cdots,2^L-1\}$ such that there exists $c \in \{0,\cdots,r-1\}$ satisfying $\big|\frac{b}{2^L} - \frac{c}{r}\big| < \frac{1}{2r^2}$ and $\gcd(c,r) = 1$; that is,

$$E = \Big\{ b \in \mathbb{N} \cup \{0\} \,\Big|\, 0 \leqslant b \leqslant 2^L - 1 \text{ and } \big|\frac{b}{2^L} - \frac{c}{r}\big| < \frac{1}{2r^2} \text{ for some } c \in \mathbb{Z}_r^* \Big\} \,.$$

We note for each $b \in \{0,\cdots,2^L-1\}$,

$$\begin{aligned} P(b) &= \frac{1}{2^{2L}} \sum_{s=0}^{r-1} \Big| \sum_{j=0}^{m_s} \exp\Big(2\pi i \frac{(jr+s)b}{2^L}\Big) \Big|^2 \\ &= \frac{1}{2^{2L}} \sum_{s=0}^{r-1} \Big[ \sum_{j_1,j_2=0}^{m_s} \exp\Big(2\pi i \frac{(j_1 r+s)b}{2^L}\Big) \exp\Big(-2\pi i \frac{(j_2 r+s)b}{2^L}\Big) \Big] \\ &= \frac{1}{2^{2L}} \sum_{s=0}^{r-1} \Big[ \sum_{j_1,j_2=0}^{m_s} \exp\Big(2\pi i \frac{j_1 rb}{2^L}\Big) \exp\Big(-2\pi i \frac{j_2 rb}{2^L}\Big) \Big] \\ &= \frac{1}{2^{2L}} \sum_{s=0}^{r-1} \Big| \Big[ \sum_{j=0}^{m_s} \exp\Big(2\pi i \frac{jrb}{2^L}\Big) \Big|^2 \,. \end{aligned}$$

Since

$$\sum_{j=0}^{d} a^j = \begin{cases} d+1 & \text{if } a = 1 \,, \\ \dfrac{1-a^{d+1}}{1-a} & \text{if } a \neq 1 \,, \end{cases}$$

we obtain that

$$\sum_{j=0}^{m_s} \exp\Big(2\pi i \frac{jrb}{2^L}\Big) = \begin{cases} m_s + 1 & \text{if } \dfrac{rb}{2^L} \in \mathbb{N} \cup \{0\}\,, \\ \dfrac{1 - e^{2\pi i \frac{(m_s+1)rb}{2^L}}}{1 - e^{2\pi i \frac{rb}{2^L}}} & \text{if } \dfrac{rb}{2^L} \notin \mathbb{N} \cup \{0\}\,; \end{cases}$$

thus

$$P(b) = \begin{cases} \dfrac{1}{2^{2L}} \displaystyle\sum_{s=0}^{r-1} (m_s+1)^2 & \text{if } \dfrac{rb}{2^L} \in \mathbb{N} \cup \{0\}\,, \\ \dfrac{1}{2^{2L}} \displaystyle\sum_{s=0}^{r-1} \Big| \dfrac{1 - e^{2\pi i \frac{(m_s+1)rb}{2^L}}}{1 - e^{2\pi i \frac{rb}{2^L}}} \Big|^2 & \text{if } \dfrac{rb}{2^L} \notin \mathbb{N} \cup \{0\}\,. \end{cases}$$

Define

$$B = \Big\{ b \in \mathbb{N} \cup \{0\} \,\Big|\, 0 \leqslant b \leqslant 2^L - 1 \text{ and } \big|rb - c2^L\big| \leqslant \frac{r}{2} \text{ for some (unique) } c \in \mathbb{Z}_r^* \Big\}\,.$$

We note that the fact that $r < 2^{L/2}$ implies that if $b \in B$,

$$\Big|\frac{b}{2^L} - \frac{c_b}{r}\Big| = \frac{1}{r2^L}\big|rb - c_b 2^L\big| \leqslant \frac{r}{2} \cdot \frac{1}{r2^L} = \frac{1}{2 \cdot 2^L} < \frac{1}{2r^2}\,.$$

In other words, $B \subseteq E$. Let $b \in B$.

1. The case $\dfrac{rb}{2^L} \in \mathbb{N} \cup \{0\}$: In this case

$$\begin{aligned} P(b) = \frac{1}{2^{2L}} \sum_{s=0}^{r-1} (m_s+1)^2 &= \frac{1}{2^{2L}} \Big[ \sum_{s=0}^{R} (m+1)^2 + \sum_{s=R+1}^{r-1} m^2 \Big] \\ &= \frac{1}{2^{2L}} \Big[ (R+1)(m+1)^2 + (r-1-R)m^2 \Big] \\ &\geqslant \frac{1}{2^{2L}} \Big[ (R+1)m^2 + (r-1-R)m^2 \Big] = \frac{1}{r} \Big( \frac{rm}{2^L} \Big)^2\,. \end{aligned}$$

   Recall that $m = \Big[\dfrac{2^L - 1}{r}\Big]$. By the fact that $r < 2^{\frac{L}{2}}$ and

$$r - 1 \geqslant (2^L - 1) \bmod r \geqslant 2^L - 1 - mr\,,$$

   we find that $\dfrac{mr}{2^L} \geqslant 1 - \dfrac{r}{2^L} > 1 - \dfrac{1}{\sqrt{2^L}}$. Therefore,

$$P(b) \geqslant \frac{1}{r}\Big(1 - \frac{1}{\sqrt{2^L}}\Big)^2 > \frac{1}{r}\Big(1 - \frac{1}{2^{L/2-1}}\Big)\,.$$

2. The case $\frac{rb}{2^L} \notin \mathbb{N} \cup \{0\}$: Suppose that $c \in \{0, 1, \cdots, r-1\}$ satisfies

$$\left|rb - c2^L\right| \leqslant \frac{r}{2}\,. \tag{6.4}$$

Then

$$\begin{aligned} P(b) &= \frac{1}{2^{2L}} \sum_{s=0}^{r-1} \left| \frac{1 - e^{2\pi i \frac{(m_s+1)rb}{2^L}}}{1 - e^{2\pi i \frac{rb}{2^L}}} \right|^2 = \frac{1}{2^{2L}} \sum_{s=0}^{r-1} \left| \frac{1 - e^{2\pi i \frac{(m_s+1)(rb-c2^L)}{2^L}}}{1 - e^{2\pi i \frac{rb-c2^L}{2^L}}} \right|^2 \\ &= \frac{1}{2^{2L}} \sum_{s=0}^{r-1} \frac{\sin^2 \pi \frac{rb-c2^L}{2^L}(m_s+1)}{\sin^2 \pi \frac{rb-c2^L}{2^L}}\,, \end{aligned}$$

where we have used the identity $|1 - e^{i\theta}| = 2\left|\sin \frac{\theta}{2}\right|$ to conclude the last equality. Let $\alpha = \pi \frac{rb - c2^L}{2^L}$. Then

$$|\alpha| \leqslant \frac{\pi}{2^L} \cdot \frac{r}{2} < \frac{\pi}{2^{\frac{L}{2}+1}} \ll \frac{\pi}{2}\,.$$

Within this range, the function $\beta \mapsto \frac{\sin^2 \beta(m_s+1)}{\sin^2 \beta}$ cannot attain its minimum in the interior of the interval and we have

$$\frac{\sin^2 \pi \frac{rb-c2^L}{2^L}(m_s+1)}{\sin^2 \pi \frac{rb-c2^L}{2^L}} = \frac{\sin^2 \alpha(m_s+1)}{\sin^2 \alpha} \geqslant \frac{\sin^2 \frac{\pi r}{2^{L+1}}(m_s+1)}{\sin^2 \frac{\pi r}{2^{L+1}}} \qquad \forall\, |\alpha| < \frac{\pi}{2}\,.$$

Note that the fact that $m \leqslant m_s + 1 \leqslant m + 1$ and $R = (2^L - 1) \bmod r$ imply that

$$\frac{r(m_s+1)}{2^L} \geqslant \frac{mr}{2^L} = \frac{mr + R + 1}{2^L} - \frac{R+1}{2^L} = 1 - \frac{R+1}{2^L} \geqslant 1 - \frac{r}{2^L}$$

and

$$\frac{r(m_s+1)}{2^L} \leqslant \frac{r(m+1)}{2^L} = \frac{mr + R + 1}{2^L} + \frac{r - R - 1}{2^L} \leqslant 1 + \frac{r}{2^L}\,.$$

Therefore, the inequalities $\sin^2 x \leqslant x^2$ and $\cos x \geqslant 1 - \frac{x^2}{2}$ for all for all $x \in \mathbb{R}$ show that

$$\begin{aligned} \frac{\sin^2 \pi \frac{rb-c2^L}{2^L}(m_s+1)}{\sin^2 \pi \frac{rb-c2^L}{2^L}} &\geqslant \frac{\sin^2 \frac{\pi r(m_s+1)}{2^{L+1}}}{\sin^2 \frac{\pi r}{2^{L+1}}} \geqslant \left(\frac{2^{L+1}}{\pi r}\right)^2 \sin^2 \frac{\pi r(m_s+1)}{2^{L+1}} \\ &\geqslant \left(\frac{2^{L+1}}{\pi r}\right)^2 \sin^2 \left[\frac{\pi}{2}\left(1 - \frac{r}{2^L}\right)\right] \geqslant \left(\frac{2^{L+1}}{\pi r}\right)^2 \left[1 - \frac{1}{2}\left(\frac{\pi}{2}\frac{r}{2^L}\right)^2\right]^2 \\ &\geqslant \frac{2^{2L+2}}{\pi^2 r^2}\left[1 - \left(\frac{\pi}{2}\frac{1}{\sqrt{2^L}}\right)^2\right] = \frac{2^{2L+2}}{\pi^2 r^2}\left(1 - \frac{\pi^2}{2^{L+2}}\right); \end{aligned}$$

thus

$$P(b) \geqslant \frac{1}{2^{2L}} \sum_{s=0}^{r-1} \frac{2^{2L+2}}{\pi^2 r^2} \Big(1 - \frac{\pi^2}{2^{L+2}}\Big) = \frac{r}{2^{2L}} \frac{2^{2L+2}}{\pi^2 r^2} \Big(1 - \frac{\pi^2}{2^{L+2}}\Big) = \frac{4}{\pi^2 r} \Big(1 - \frac{\pi^2}{2^{L+2}}\Big) .$$

For $L \geqslant 4$, we have

$$\frac{4}{\pi^2 r} \Big(1 - \frac{\pi^2}{2^{L+2}}\Big) \leqslant \frac{1}{r} \Big(1 - \frac{1}{2^{L/2-1}}\Big) ;$$

thus

$$P_{\min} \equiv \frac{4}{\pi^2 r} \Big(1 - \frac{\pi^2}{2^{L+2}}\Big) \leqslant P(b) \qquad \text{if } b \in B \text{ and } L \geqslant 4 .$$

Now we find a lower bound for $\mathbf{P}(E)$, the probability of measuring an element of $E$. By the definition of $B$ for any $b \in B$ there exists $c \in \mathbb{Z}_r^*$ satisfying (6.4). Moreover, if $c_1, c_2 \in \mathbb{Z}_r^*$ satisfy $\big|rb - c_1 2^L\big| \leqslant \frac{r}{2}$ and $\big|rb - c_2 2^L\big| \leqslant \frac{r}{2}$, then

$$\big|c_1 - c_2\big| \leqslant \Big|c_1 - \frac{rb}{2^L}\Big| + \Big|c_2 - \frac{rb}{2^L}\Big| \leqslant \frac{1}{2^L} \Big(\big|rb - c_1 2^L\big| + \big|rb - c_2 2^L\big|\Big) \leqslant \frac{r}{2^L} < 1 .$$

Therefore, for any $b \in B$ there exists a unique $c = c_b \in \mathbb{Z}_r^*$ satisfying (6.4). On the other hand, by the fact that $r < 2^{L/2}$, every $c \in \mathbb{Z}_r^*$ corresponds to a unique $b = b_c \in \{0, 1, \cdots, 2^L - 1\}$ such that (6.4) holds: if $b_1$ and $b_2$ both satisfy (6.4), then $|b_1 - b_2| = 1$ and

$$\frac{b_1 + b_2}{2} r = c 2^L$$

which, by the fact that $\gcd(b_1 + b_2, 2^{L+1}) = 1$, implies that $2^{L+1} | r$, a contradiction.

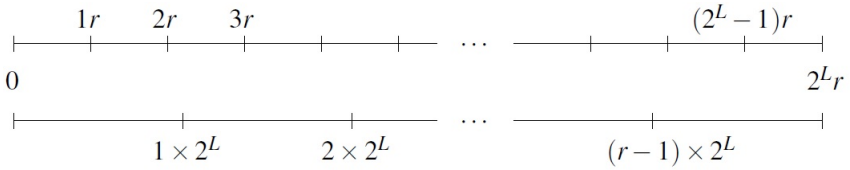



Figure 6.2: The distribution of $br$ and $c2^L$ for various $b$ and $c$.

Therefore, there is an one-to-one correspondence between $\mathbb{Z}_r^*$ and $B$; thus if $L \geqslant 4$,

$$\begin{aligned} \mathbf{P}(E) &= \sum_{b \in E} P(b) \geqslant \sum_{b \in B} P(b) \geqslant \sum_{b \in B} \frac{4}{\pi^2 r} \Big(1 - \frac{\pi^2}{2^{L+2}}\Big) \\ &= \frac{\#B}{r} \frac{4}{\pi^2} \Big(1 - \frac{\pi^2}{2^{L+2}}\Big) = \frac{\#\mathbb{Z}_r^*}{r} \frac{4}{\pi^2} \Big(1 - \frac{\pi^2}{2^{L+2}}\Big) \\ &= \frac{\varphi(r)}{r} \frac{4}{\pi^2} \Big(1 - \frac{\pi^2}{2^{L+2}}\Big) . \end{aligned}$$

A famous result in number theory implies that

$$\frac{r}{\varphi(r)} < 4 \ln \ln r \qquad \forall\, r \geqslant 19\,;$$

thus if $r \geqslant 19$ (so that $L \geqslant 9$),

$$\mathbf{P}(E) \geqslant \frac{4}{\pi^2}\Big(1 - \frac{\pi^2}{2^{11}}\Big)\frac{1}{4 \ln \ln r} > \frac{1}{10 \ln L}\,.$$

Once we measure a $b \in E$, we make use of continuous fractions to find the period $r$. □

### 6.5.2 The period of $f(a) = x^a \bmod N$ is most likely even

In this sub-section we focus on proving the following

**Theorem 6.27.** *Let $N \in \mathbb{N}$ be odd with prime factorization $N = \prod_{j=1}^{J} p_j^{\nu_j}$, where $p_1, \cdots, p_J$ are distinct prime numbers. For a randomly chosen $b \in \mathbb{Z}_N^*$, the probability of that $r \equiv \min\big\{r \in \mathbb{N}\,\big|\, b^r = 1 \bmod N\big\}$ is even and $b^{r/2} + 1 \bmod N \neq 0$ is at least $1 - \frac{1}{2^{J-1}}$.*

In the application of the factoring algorithm proposed in the previous sections, $J = 2$ so that the probability of that for a randomly chosen $b \in \mathbb{Z}_N^*$ the number $\gcd(b^{r/2} + 1, N)$ is a prime factor of $N$ is at least $1/2$.

Let $N \in \mathbb{N}$. Recall that $\mathbb{Z}_N^*$ consists of numbers from $\{1, 2, \cdots, N-1\}$ that is coprime to $N$; that is,

$$\mathbb{Z}_N^* = \big\{n \in \mathbb{N}\,\big|\, 1 \leqslant n \leqslant N-1 \text{ and } \gcd(n, N) = 1\big\}\,.$$

The number of elements in $\mathbb{Z}_N^* = \varphi(N)$, where $\varphi$ is the Euler function (given in Definition 6.6).

**Definition 6.28.** Let $b, N \in \mathbb{N}$ with $\gcd(b, N) = 1$. The order of $b$ in $\mathbb{Z}_N^*$, denoted by $\mathrm{ord}_N(b)$, is the period of the function $f(x) = b^x - 1 \bmod N$. In other words,

$$\mathrm{ord}_N(b) = \min\big\{r \in \mathbb{N}\,\big|\, b^r = 1 \bmod N\big\}\,.$$

If $\mathrm{ord}_N(b) = \varphi(N)$, then $b$ is called a primitive root modulo $N$.

**Theorem 6.29.** *Let $a, b, N \in \mathbb{N}$ with $\gcd(a, N) = 1 = \gcd(b, N)$. Then the following statements hold.*

1. *For all $k \in \mathbb{N}$, $a^k = 1$ mod $N$ if and only if $\mathrm{ord}_N(a) | k$.*

2. $\mathrm{ord}_N(a)|\varphi(N)$*; that is,* $\mathrm{ord}_N(a)$ *is a factor of* $\varphi(N)$*.*

3. *If* $\mathrm{ord}_N(a)$ *and* $\mathrm{ord}_N(b)$ *are coprime, then* $\mathrm{ord}_N(ab) = \mathrm{ord}_N(a)\mathrm{ord}_N(b)$*.*

4. *If* $a$ *is a primitive root modulo* $N$*; that is,* $\mathrm{ord}_N(a) = \varphi(N)$*, then we also have*

   (a) $\mathbb{Z}_N^* = \left\{a^j \ mod\ N \,\middle|\, 1 \leqslant j \leqslant \varphi(N)\right\}$.

   (b) *If* $b = a^j \ mod\ N$ *for some* $j \in \mathbb{N}$*, then*

$$\mathrm{ord}_N(b) = \mathrm{ord}_N(a^j) = \frac{\varphi(N)}{\gcd(j, \varphi(N))}\,. \tag{6.5}$$

*Proof.* Let $a, b, N \in \mathbb{N}$ with $\gcd(a, N) = 1 = \gcd(b, N)$.

1. ("⇒") Let $k \in \mathbb{N}$ satisfying $a^k = 1 \bmod N$. Then $k \geqslant \mathrm{ord}_N(a)$. Let $c = k \bmod \mathrm{ord}_N(a)$; that is, there exists $q \in \mathbb{N}$ such that $k = q \cdot \mathrm{ord}_N(a) + c$ for some $c \in \{0, 1, \cdots, \mathrm{ord}_N(a) - 1\}$. Then

$$1 = a^k \bmod N = a^{q\cdot\mathrm{ord}_N(a)+c} \bmod N = a^c \bmod N\,;$$

   thus by the definition of the order we must have $c = 0$. Therefore, $\mathrm{ord}_N(a)|k$.

   ("⇐") Suppose that $\mathrm{ord}_N(a)|k$. Then $k = q \cdot \mathrm{ord}_N(a)$ for some $q \in \mathbb{N}$. Therefore,

$$a^k \bmod N = a^{q\cdot\mathrm{ord}_N(a)} \bmod N = 1\,.$$

2. By Theorem 6.18, we know that $a^{\varphi(N)} = 1 \bmod N$; thus part 2 follows from part 1.

3. By Proposition 6.13, the rule of multiplication in $\mathbb{Z}_N^*$, we find that

$$(ab)^{\mathrm{ord}_N(a)\mathrm{ord}_N(b)} \bmod N = a^{\mathrm{ord}_N(a)} b^{\mathrm{ord}_N(b)} \bmod N = 1\,;$$

   thus part 1 implies that

$$\mathrm{ord}_N(ab)|\mathrm{ord}_N(a)\mathrm{ord}_N(b)\,. \tag{6.6}$$

   On the other hand, by the fact that $b^{\mathrm{ord}_N(b)\mathrm{ord}_N(ab)} = 1 \bmod N$,

$$\begin{aligned} a^{\mathrm{ord}_N(b)\mathrm{ord}_N(ab)} \bmod N &= a^{\mathrm{ord}_N(b)\mathrm{ord}_N(ab)} b^{\mathrm{ord}_N(b)\mathrm{ord}_N(ab)} \bmod N \\ &= (ab)^{\mathrm{ord}_N(b)\mathrm{ord}_N(ab)} \bmod N = 1\,. \end{aligned}$$

Therefore, part 1 shows that

$$\mathrm{ord}_N(a)|\mathrm{ord}_N(b)\mathrm{ord}_N(ab)\,.$$

By the assumption that $\mathrm{ord}_N(a)$ and $\mathrm{ord}_N(b)$ are coprime, we must have $\mathrm{ord}_N(a)|\mathrm{ord}_N(ab)$. Similarly, we also have $\mathrm{ord}_N(b)|\mathrm{ord}_N(ab)$. Therefore,

$$\mathrm{ord}_N(a)\mathrm{ord}_N(b)|\mathrm{ord}_N(ab)$$

which, together with (6.6), shows the desired result.

4. Suppose that $\mathrm{ord}_N(a) = \varphi(N)$.

   (a) First we note that Theorem 6.15 implies that $\big\{a^j \bmod N \,\big|\, 1 \leqslant j \leqslant \varphi(N)\big\} \subseteq \mathbb{Z}_N^*$. It then suffices to show that

   $$\#\big\{a^j \bmod N \,\big|\, 1 \leqslant j \leqslant \varphi(N)\big\} = \varphi(N)\,. \tag{6.7}$$

   Let $i, j \in \mathbb{N}$ with $1 \leqslant i \leqslant j \leqslant \varphi(N)$, and suppose that $a^i = a^j \bmod N$. Then $a^{j-i} = 1 \bmod N$. Therefore, part 1 shows that $\mathrm{ord}_N(a)|(j-i)$. Since $\mathrm{ord}_N(a) = \varphi(N)$ and $1 \leqslant i \leqslant j \leqslant \varphi(N)$, we must have $i = j$; thus (6.7) holds.

   (b) We first establish the first "=" of (6.5); that is, if $b = a^j \bmod N$, then $\mathrm{ord}_N(b) = \mathrm{ord}_N(a^j)$. To see that, we note that the identity

   $$1 = b^{\mathrm{ord}_N(b)} \bmod N = (a^j \bmod N)^{\mathrm{ord}_N(b)} \bmod N = (a^j)^{\mathrm{ord}_N(b)} \bmod N$$

   shows that $\mathrm{ord}_N(a^j) \leqslant \mathrm{ord}_N(b)$, while the identity

   $$1 = (a^j)^{\mathrm{ord}_N(a^j)} \bmod N = (a^j \bmod N)^{\mathrm{ord}_N(a^j)} \bmod N = b^{\mathrm{ord}_N(a^j)} \bmod N$$

   shows that $\mathrm{ord}_N(b) \leqslant \mathrm{ord}_N(a^j)$.

   Next we focus on the second "=" of (6.5). We note that part 2 implies that there exists $m_1 \in \mathbb{N}$ such that $m_1 \cdot \mathrm{ord}_N(a^j) = \varphi(N)$; thus it suffices to show that $m_1 = \gcd(j, \varphi(N))$.

   We remark that $m_1$ must satisfy $m_1|\varphi(N)$. Moreover, since

   $$1 = (a^j)^{\mathrm{ord}_N(a^j)} \bmod N = a^{j\cdot\mathrm{ord}_N(a^j)} \bmod N\,,$$

we have $\mathrm{ord}_N(a)|j \cdot \mathrm{ord}_N(a^j)$. By the assumption that $\mathrm{ord}_N(a) = \varphi(N)$, there exists $m_2 \in \mathbb{N}$ such that $m_2 \cdot \varphi(N) = j \cdot \mathrm{ord}_N(a^j)$. Therefore, $j = m_1 m_2$. In particular, $m_1|j$; thus the fact that $m_1|\varphi(N)$ further shows that

$$m_1|\gcd(j, \varphi(N))\,.$$

Suppose the contrary that $m_1 < \widehat{m} \equiv \gcd(j, \varphi(N))$. Then

$$\widehat{r} \equiv \frac{\varphi(N)}{\widehat{m}} < \frac{\varphi(N)}{m_1} = \mathrm{ord}_N(a^j)\,. \tag{6.8}$$

On the other hand, the fact that $\widehat{m}|j$ shows that

$$\begin{aligned}(a^j)^{\widehat{r}} \bmod N &= (a^j)^{\frac{\varphi(N)}{\widehat{m}}} \bmod N = \left(a^{\varphi(N)}\right)^{\frac{j}{\widehat{m}}} \bmod N \\ &= \left(a^{\varphi(N)} \bmod N\right)^{\frac{j}{\widehat{m}}} \bmod N = 1\,.\end{aligned}$$

Thus, we conclude from part 1 that $\mathrm{ord}_N(a^j)|\widehat{r}$, a contradiction to (6.8). ▫

**Lemma 6.30.** *Let $p$ be a prime, $k \in \mathbb{N} \cup \{0\}$, and $f_0$, $f_1$, $\cdots$, $f_k$ be integers such that $p \nmid f_k$. If $f$ is a polynomial given by $f(x) = \sum_{j=0}^{k} f_j x^j$, then either*

1. *$f$ has at most $k$ distinct zeros modulo $p$ in $\mathbb{Z}_p^*$; that is,*

$$\#\left\{x \in \mathbb{Z}_p^* \,\middle|\, f(x) = 0 \ mod\ p\right\} \leqslant k$$

*or*

2. *$f$ is the zero-polynomial modulo $p$; that is, $f(x) = 0$ mod $p$ for all $x \in \mathbb{Z}$ (or $\mathbb{Z}_p^*$).*

*Proof.* We show this by induction in the degree of the polynomial, which we start at $k = 0$: if $f(x) = f_0 \neq 0$ such that $p \nmid f_0$, then it follows that $f_0 \neq 0 \bmod p$, and there is no $x \in \mathbb{Z}$ with $f(x) = 0 \bmod p$. If $f_0 = 0$, then $f$ is the zero-polynomial.

Suppose then the claim holds for all polynomials of degree up to $k-1$ and $f$ is a polynomial of degree $k$. If $f$ has fewer than $k$ zeros modulo $p$ in $\mathbb{Z}_p^*$, the claim holds already. Suppose that $f$ has at least $k$ zeros modulo $p$, and $n_1$, $n_2$, $\cdots$, $n_k \in \mathbb{Z}_p^*$ are distinct zeros of $f$ modulo $p$ (there may be more zeros of $f$ modulo $p$, but we randomly pick $k$ distinct zeros). Then

$$g(x) \equiv f(x) - f_k \prod_{j=1}^{k}(x - n_j) = \sum_{\ell=0}^{k-1} g_\ell x^\ell$$

is a polynomial of degree not exceeding $k-1$. Set $m = \max\left\{\ell \in \{0,1,\cdots,k-1\} \middle| p \nmid g_\ell\right\}$, and define $\widetilde{g}(x) = \sum_{\ell=0}^{m} g_\ell x^\ell$. Then for $x \in \mathbb{Z}$,

$$\widetilde{g}(x) \bmod p = \sum_{\ell=0}^{m} g_\ell x^\ell \bmod p = \sum_{\ell=0}^{k-1} g_\ell x^\ell \bmod p = g(x) \bmod p.$$

Moreover, for $1 \leqslant j \leqslant k$ we have $g(n_j) = f(n_j) = 0 \bmod p$. Therefore, $\widetilde{g}$ has at least $k$ zeros modulo $p$; thus by the induction assumption $\widetilde{g}$ must be the zero polynomial. This shows that $g$ is also the zero polynomial. By the definition of $g$,

$$f(x) = f_k \prod_{j=1}^{k}(x - n_j) \bmod p \qquad \forall\, x \in \mathbb{Z}.$$

Suppose that $z$ is a zero of $f$ modulo $p$. Then by the fact that $p \nmid f_k$, the cancellation law for $\mathbb{Z}_p$ implies that $z - n_j = 0 \bmod p$ for some $1 \leqslant j \leqslant k$. □

**Lemma 6.31.** *Let $p$ be prime, $d$ a natural number satisfying $d|(p-1)$ and let $h$ be the polynomial $h(x) = x^d - 1$. Then there exist exactly $d$ distinct numbers $n_1, n_2, \cdots, n_d$ in $\mathbb{Z}_p^*$ satisfying $h(n_j) = 0$ mod $p$.*

*Proof.* Let $k \in \mathbb{N}$ be such that $p-1 = dk$. Define $f(x) = \sum_{\ell=0}^{k-1} x^{d\ell}$ and $g = hf$. Then

$$g(x) = (x^d - 1)\sum_{\ell=0}^{k-1} x^{d\ell} = x^{kd} - 1 = x^{p-1} - 1.$$

Therefore, $g(x) = 0 \bmod p$ for all $x \in \mathbb{Z}_p^*$. The cancellation law in $\mathbb{Z}_p$ further implies that

$$\text{for all } x \in \mathbb{Z}_p^*, \text{ either } h(x) = 0 \bmod p \text{ or } f(x) = 0 \bmod p.$$

Since $h(p-1) = p - 2 \bmod p$ and $f(1) = k$, $h$ and $f$ are not zero polynomials. By the fact that the leading coefficient of $f$ and $h$ are both 1 (and $p \nmid 1$), Lemma 6.30 implies that the polynomial $h$ has at most $d$ and the polynomial $f$ has at most $d(k-1)$ zeros modulo $p$ in $\mathbb{Z}_p^*$. Denoting the number of zeros modulo $p$ in $\{1, \cdots p-1\}$ of the polynomials $g$, $h$ and $f$ by $N_g$, $N_h$ and $N_f$, we have

$$dk = N_g \leqslant N_h + N_f \leqslant d + d(k-1) = dk.$$

Therefore, exactly $d(k-1)$ elements in $\mathbb{Z}_p^*$ are zeros of $f$ modulo $p$, and exactly $d$ elements in $\mathbb{Z}_p^*$ are zeros of $h$ modulo $p$. □

**Theorem 6.32.** *For every odd prime $p$ there exists at least one primitive root $a$ modulo $p$; that is, a natural number $a$ such that*

$$\mathrm{ord}_p(a) = \min\left\{r \in \mathbb{N} \,\middle|\, a^r = 1 \ mod\ p\right\} = \varphi(p) = p-1\,.$$

*Proof.* For a prime factor $q$ of $p-1$, let $k_q$ be the unique number satisfying $q^{k_q}|(p-1)$ but $q^{k_q+1} \nmid (p-1)$. We first prove that for each prime factor $q$ of $p-1$ there exists $a = a_q \in \mathbb{Z}_p^*$ such that $\mathrm{ord}_p(a_q) = q^{k_q}$.

Let $q$ be a prime factor of $p-1$. By Lemma 6.31 we know that the polynomial $h(x) \equiv x^{q^{k_q}}-1$ has exactly $q^{k_q}$ zeros modulo $p$ in $\mathbb{Z}_p^*$. Let $a_q$ be one of these zeros, then $a_q^{q^{k_q}} = 1 \bmod p$ so it follows that $\mathrm{ord}_p(a_q)|q^{k_q}$. If this zero $a_q$ of $h$ has the additional property $\mathrm{ord}_p(a_q)|q^j$ for some $j \in \mathbb{N}$ with $j < k_q$, then $\mathrm{ord}_p(a_q)|q^{k_q-1}$ holds. Then

$$a_q^{q^{k_q-1}} = 1 \bmod p\,.$$

Hence, $a_q \in \mathbb{Z}_p^*$ is a zero modulo $p$ of the polynomial $f(x) \equiv x^{q^{k_q-1}}-1$. By Lemma 6.31, there are exactly $q^{k_q-1}$ of these. Of the $q^{k_q}$ zeros modulo $p$ in $\mathbb{Z}_p^*$ of $h$ at most $q^{k_q-1}$ can be zeros of $f$ as well. This means that of the $q^{k_q}$ zeros $a_q$ of $h$ at most $q^{k_q-1}$ such $a_q$ satisfy in addition $\mathrm{ord}_p(a_q)|q^j$ with $j < k_q$. Therefore, there remain $q^{k_q}-q^{k_q-1}$ zeros $a_q \in \{1, \cdots, p-1\}$ that satisfy

$$\mathrm{ord}_p(a_q)|q^{k_q} \qquad \text{and} \qquad \mathrm{ord}_p(a_q) \nmid q^j \quad \forall\, j < k_q\,. \tag{6.9}$$

Since $q$ is assumed prime, we conclude that there are $q^{k_q}-q^{k_q-1}$ numbers $a_q \in \{1, 2, \cdots, p-1\}$ satisfying $q^{k_q} = \mathrm{ord}_p(a_q)$. This establishes the first statement.

Let $q_1, q_2, \cdots, q_\ell$ be distinct prime factors of $p-1$. Rewrite $k_{q_j}$ as $k_j$ and let $a_1, a_2, \cdots, a_\ell$ be one particular number in $\{1, 2, \cdots, p-1\}$ satisfying $\mathrm{ord}_p(a_j) = q_j^{k_j}$ for $1 \leqslant j \leqslant \ell$. Define $a = \prod_{j=1}^{\ell} a_j$. Then $a$ is a primitive root modulo $p$ since

$$\mathrm{ord}_p(a) = \prod_{j=1}^{\ell} \mathrm{ord}_p(a_j)$$

which can be shown inductively using part 3 of Theorem 6.29. ▫

**Lemma 6.33.** *Let $p$ be an odd prime and $a$ be a natural number satisfying*

$$\gcd(a, p) = 1 \qquad \textit{and} \qquad a^{\varphi(p)} \bmod p^2 \neq 1\,.$$

*Then for all $k \in \mathbb{N}$, $a^{\varphi(p^k)} \bmod p^{k+1} \neq 1$.*

*Proof.* We first note that if $k \in \mathbb{N}$, by the fact that $\gcd(a, p^k) = 1$ the Euler Theorem (Theorem 6.18) implies that

$$a^{\varphi(p^k)} \bmod p^k = 1\,;$$

thus there exists $n_k \in \mathbb{N}$ such that

$$a^{\varphi(p^k)} = 1 + n_k p^k\,.$$

Let $D = \left\{k \in \mathbb{N} \,\middle|\, a^{\varphi(p^k)} \bmod p^{k+1} \neq 1\right\}$. By assumption, $1 \in D$. Suppose that $k \in D$. Then $a^{\varphi(p^k)} \neq 1 + mp^{k+1}$ for all $m \in \mathbb{N}$. Therefore, $p \nmid n_k$. By Proposition 6.7,

$$\varphi(p^{k+1}) = p^k(p-1) = p\varphi(p^k)\,;$$

thus

$$a^{\varphi(p^{k+1})} = a^{p\varphi(p^k)} = (a^{\varphi(p^k)})^p = (1 + n_k p^k)^p = 1 + n_k p^{k+1} + \sum_{\ell=2}^{p} C_\ell^p n_k^\ell p^{k\ell}\,.$$

Therefore, by the fact that $p \nmid n_k$, we find that

$$a^{\varphi(p^{k+1})} \bmod p^{k+2} = (1 + n_k p^{k+1}) \bmod p^{k+2} \neq 1\,.$$

This shows that $k + 1 \in D$. The lemma is then concluded by induction. □

**Theorem 6.34.** *Let $p$ be an odd prime and $a$ be a primitive root modulo $p$. Then for all $k \in \mathbb{N}$ either $\mathrm{ord}_{p^k}(a) = \varphi(p^k)$ or $\mathrm{ord}_{p^k}(a+p) = \varphi(p^k)$; that is, either $a$ or $a+p$ is a primitive root modulo $p^k$.*

*Proof.* Let $a$ be a primitive root modulo $p$.

**Case 1** - $a^{\varphi(p)} \bmod p^2 \neq 1$: Let $D = \left\{k \in \mathbb{N} \,\middle|\, \mathrm{ord}_{p^k}(a) = \varphi(p^k)\right\}$. Since $a$ is a primitive root modulo $p$, we find that $1 \in D$. Suppose that $k \in D$. By the definition of the order there exists $n \in \mathbb{N}$ such that

$$a^{\mathrm{ord}_{p^{k+1}}(a)} = 1 + np^{k+1} = 1 + npp^k\,.$$

Therefore, $a^{\mathrm{ord}_{p^{k+1}}(a)} \equiv 1 \bmod p^k$ and Theorem 6.29 implies that $\mathrm{ord}_{p^k}(a) | \mathrm{ord}_{p^{k+1}}(a)$. By the inductive assumption, $\mathrm{ord}_{p^k}(a) = \varphi(p^k) = p^{k-1}(p-1)$; thus

$$p^{k-1}(p-1) | \mathrm{ord}_{p^{k+1}}(a)\,.$$

This implies that there exists $n_1 \in \mathbb{N}$ such that $\text{ord}_{p^{k+1}}(a) = n_1 p^{k-1}(p-1)$. On the other hand, Theorem 6.29 also implies that

$$\text{ord}_{p^{k+1}}(a) | \varphi(p^{k+1}) = p^k(p-1)\,;$$

thus there exists $n_2 \in \mathbb{N}$ such that $n_2 \cdot \text{ord}_{p^{k+1}}(a) = p^k(p-1)$. Therefore, $n_1 n_2 = p$ which, by the fact that $p$ is prime, shows that $(n_1, n_2) = (1, p)$ or $(n_1, n_2) = (p, 1)$. If $(n_1, n_2) = (1, p)$, then $\text{ord}_{p^{k+1}}(a) = p^{k-1}(p-1) = \varphi(p^k)$ which further shows that

$$a^{\varphi(p^k)} \bmod p^{k+1} = 1\,,$$

a contradiction to Lemma 6.33. Therefore, $(n_1, n_2) = (p, 1)$ and we then have

$$\text{ord}_{p^{k+1}}(a) = p^k(p-1) = \varphi(p^{k+1})\,.$$

This concludes that $k+1 \in D$. By induction, $D = \mathbb{N}$.

**Case 2** - $a^{\varphi(p)} \bmod p^2 = 1$: First we note that in this case there exists $n_3 \in \mathbb{N}$ such that $a^{p-1} = 1 + n_3 p^2$. Let $r = \text{ord}_p(a+p)$. Then $r | \varphi(p)$ and

$$(a+p)^r \bmod p = 1\,.$$

By binomial expansion, $a^r \bmod p = 1$ which further implies that $\varphi(p) | r$. Therefore, $r = \varphi(p)$; thus $a+p$ is also a primitive root modulo $p$. Next we show that $(a+p)^{\varphi(p)} \bmod p^2 \neq 1$. To see this, by binomial expansion we have

$$\begin{aligned}(a+p)^{p-1} &= a^{p-1} + (p-1)a^{p-2}p + \sum_{\ell=2}^{p-1} C_\ell^{p-1} a^{p-\ell-1} p^\ell \\ &= 1 + n_3 p^2 - pa^{p-2} + p^2 a^{p-2} + p^2 \sum_{\ell=2}^{p-1} C_\ell^{p-1} a^{p-\ell-1} p^{\ell-2} \\ &= 1 + n_4 p^2 - pa^{p-2}\,.\end{aligned}$$

Since (by Fermat little theorem) $a^{p-1} \bmod p = 1$, $p \nmid a^{p-2}$; thus $(a+p)^{\varphi(p)} \bmod p^2 \neq 1$. Therefore, Case 1 shows that $\text{ord}_{p^{k+1}}(a+p) = \varphi(p^{k+1})$. □

**Theorem 6.35.** *Let* $N = \prod_{j=1}^{J} n_j$ *with* $n_j \in \mathbb{N}$ *and* $\gcd(n_i, n_j) = 1$ *if* $i \neq j$. *Then* $g : \mathbb{Z}_N^* \to \mathbb{Z}_{n_1}^* \times \mathbb{Z}_{n_2}^* \times \cdots \times \mathbb{Z}_{n_J}^*$ *defined by*

$$g(a) = \big(a \bmod n_1, a \bmod n_2, \cdots, a \bmod n_J\big)$$

*is a bijection.*

*Proof.* We first show that $g(\mathbb{Z}_N^*) \subseteq \mathbb{Z}_{n_1}^* \times \mathbb{Z}_{n_2}^* \times \cdots \times \mathbb{Z}_{n_J}^*$. For each $1 \leqslant j \leqslant J$, let $g_j(a) = a \bmod n_j$. Then $g = (g_1, \cdots, g_J)$, and $g_j(a) \in \mathbb{Z}_{n_j}^*$ for all $a \in \mathbb{Z}_N^*$. Let $a \in \mathbb{Z}_N^*$ and $j \in \{1, 2, \cdots, J\}$ be given, and $\gamma = \gcd(g_j(a), n_j)$. Then there exist $\ell, k \in \mathbb{N}$ such that $g_j(a) = \gamma\ell$ and $n_j = \gamma k$. Since

$$\gamma\ell = g_j(a) = a - \Big[\frac{a}{n_j}\Big] n_j = a - \Big[\frac{a}{n_j}\Big] \gamma k \,,$$

we find that

$$\frac{a}{\gamma} = \ell + \Big[\frac{a}{n_j}\Big] k \,.$$

Therefore, $\gamma | a$. Moreover, $n_j | N$, we must have $\gamma | N$ as well; thus the fact that $\gcd(a, N) = 1$ implies that $\gamma = 1$. In other words, $\gcd(g_j(a), n_j) = 1$ for all $1 \leqslant j \leqslant J$, and this shows that $g_j(a) \in \mathbb{Z}_{n_j}^*$ for all $1 \leqslant j \leqslant J$; thus $g(\mathbb{Z}_N^*) \subseteq \mathbb{Z}_{n_1}^* \times \mathbb{Z}_{n_2}^* \times \cdots \times \mathbb{Z}_{n_J}^*$.

Next we show that $g$ is injective. Suppose the contrary that there exist $a_1, a_2 \in \mathbb{Z}_N^*$, $a_1 \neq a_2$, such that $g(a_1) = g(a_2)$. W.L.O.G. we assume that $a_1 > a_2$. Then for all $1 \leqslant j \leqslant J$, $g_j(a_1) = g_j(a_2)$; thus

$$a_1 - a_2 = \Big( \Big[\frac{a_1}{n_j}\Big] - \Big[\frac{a_2}{n_j}\Big] \Big) n_j \qquad \forall\, 1 \leqslant j \leqslant J \,.$$

Therefore, $n_j | (a_1 - a_2)$ for all $1 \leqslant j \leqslant J$. Since $\gcd(n_i, n_j) = 1$ if $i \neq j$ and $N = \prod_{j=1}^{J} n_j$, we find that $N | (a_1 - a_2)$, a contradiction. This establishes that $g$ is injective.

Finally, we prove that $g$ is surjective. Let $m_j = N/n_j$. Then $\gcd(m_j, n_j) = 1$; thus there exist $x_j, y_j \in \mathbb{Z}$ such that $m_j x_j + n_j y_j = 1$. For $\boldsymbol{b} = (b_1, \cdots, b_J) \in \mathbb{Z}_{n_1}^* \times \mathbb{Z}_{n_2}^* \times \cdots \times \mathbb{Z}_{n_J}^*$, define

$$h(\boldsymbol{b}) = \Big( \sum_{j=1}^{J} m_j x_j b_j \Big) \bmod N \,. \tag{6.10}$$

This definition of $h$ is well-defined: if $\tilde{x}_j$ and $\tilde{y}_j$ also validate $m_j \tilde{x}_j + n_j \tilde{y}_j = 1$, then for all

$1 \leqslant k \leqslant J$, by the fact that $n_j/m_k \in \mathbb{N}$ if $j \neq k$, we find that

$$\begin{aligned}\frac{1}{n_k}\sum_{j=1}^{J} m_j(x_j-\tilde{x}_j)b_j &= \sum_{j\neq k}\frac{m_j}{n_k}(x_j-\tilde{x}_j)b_j + \frac{m_k}{n_k}(x_k-\tilde{x}_k)b_k \\ &= \sum_{j\neq k}\frac{m_j}{n_k}(x_j-\tilde{x}_j)b_j + \frac{m_kx_k-m_k\tilde{x}_k}{n_k}b_k \\ &= \sum_{j\neq k}\frac{m_j}{n_k}(x_j-\tilde{x}_j)b_j + \frac{(1-n_ky_k)-(1-n_k\tilde{y}_k)}{n_k}b_k \\ &= \sum_{j\neq k}\frac{m_j}{n_k}(x_j-\tilde{x}_j)b_j - (y_k-\tilde{y}_k)b_k \in \mathbb{Z}\,.\end{aligned}$$

This shows that $n_k$ is a factor of $\sum\limits_{j=1}^{J} m_j(x_j-\tilde{x}_j)b_j$ for all $1 \leqslant k \leqslant J$. Since $\gcd(n_i,n_j)=1$ if $i \neq j$, we also have $N$ is a factor of $\sum\limits_{j=1}^{J} m_j(x_j-\tilde{x}_j)b_j$. Therefore,

$$\Big(\sum_{j=1}^{J} m_jx_jb_j\Big) \bmod N = \Big(\sum_{j=1}^{J} m_j\tilde{x}_jb_j\Big) \bmod N\,;$$

thus $h$ given by (6.10) is well-defined.

Now we show that $g$ is surjective by showing that $h(\boldsymbol{b}) \in \mathbb{Z}_N^*$ and $g(h(\boldsymbol{b})) = \boldsymbol{b}$ for all $\boldsymbol{b} \in \mathbb{Z}_{n_1}^* \times \mathbb{Z}_{n_2}^* \times \cdots \times \mathbb{Z}_{n_J}^*$. Let $\boldsymbol{b} \in \mathbb{Z}_{n_1}^* \times \mathbb{Z}_{n_2}^* \times \cdots \times \mathbb{Z}_{n_J}^*$ be given. For a fixed $k \in \{1,2,\cdots,J\}$,

$$\begin{aligned}\frac{1}{n_k}(h(\boldsymbol{b})-b_k) &= \frac{1}{n_k}\Big[\Big(\sum_{j=1}^{J} m_jx_jb_j\Big) \bmod N - b_k\Big] \\ &= \frac{1}{n_k}\left\{\sum_{j=1}^{J} m_jx_jb_j - \left[\frac{\sum_{j=1}^{J} m_jx_jb_j}{N}\right]N - b_k\right\} \\ &= \sum_{j\neq k}\frac{m_j}{n_k}x_jb_j + \frac{m_kx_k-1}{n_k}b_k - \left[\frac{\sum_{j=1}^{J} m_jx_jb_j}{N}\right]\frac{N}{n_k} \in \mathbb{Z}\,.\end{aligned}$$

Therefore, for each $1 \leqslant k \leqslant J$ there exists $z_k \in \mathbb{Z}$ such that

$$h(\boldsymbol{b}) = b_k + z_kn_k\,. \tag{6.11}$$

It then suffices to show that $h(\boldsymbol{b}) \in \mathbb{Z}_N^*$ since then $g_k(h(\boldsymbol{b})) = b_k$ which establishes that $g(h(\boldsymbol{b})) = \boldsymbol{b}$. Nevertheless, (6.11) implies that

$$\gcd(h(\boldsymbol{b}),n_k) = \gcd(b_k,n_k) = 1 \qquad \forall\, 1 \leqslant k \leqslant J\,.$$

The fact that $\gcd(n_i, n_j) = 1$ if $i \neq j$ further shows that $\gcd(h(\boldsymbol{b}), N) = 1$; thus $h(\boldsymbol{b}) \in \mathbb{Z}_N^*$ and we conclude that $g$ is surjective. ▫

**Lemma 6.36.** *Let $p$ be an odd prime, $k \in \mathbb{N}$, and $s \in \mathbb{N} \cup \{0\}$. For a randomly chosen $b$ from $\mathbb{Z}_{p^k}^*$ with equally distributed probability $1/\varphi(p^k)$, the probability of that $\mathrm{ord}_{p^k}(b)/2^s$ is an odd number is not greater than $\frac{1}{2}$. In other words,*

$$(\forall\, p, k, s)\Big(\#\big\{b \in \mathbb{Z}_{p^k}^* \,\big|\, \mathrm{ord}_{p^k}(b) = 2^s t \text{ with an odd } t\big\} \leqslant \frac{1}{2}\varphi(p^k)\Big)\,.$$

*Proof.* Let $p$, $k$ and $s$ be given. By the definition of the Euler function, $\#\mathbb{Z}_{p^k}^* = \varphi(p^k)$. Furthermore, there exist uniquely determined $\mu, \nu \in \mathbb{N}$ with $\nu$ odd such that

$$\varphi(p^k) = p^k(p-1) = 2^\mu \nu\,.$$

By Theorems 6.32 and 6.34 it follows that there exists a primitive root $a \in \mathbb{N}$ for $p^k$ and from Theorem 6.29 it follows that

$$\mathbb{Z}_{p^k}^* = \big\{a^j \bmod p^k \,\big|\, j \in \{1, 2, \cdots, \varphi(p^k)\big\}\,.$$

Hence, via the identification $b = a^j \bmod p^k$, the random selection of one of the equally distributed $b$ in $\mathbb{Z}_{p^k}^*$ is the same as the random selection of an equally distributed $j \in \big\{1, \cdots, \varphi(p^k)\big\}$. Moreover, we know from Theorem 6.29 that

$$\mathrm{ord}_{p^k}(b) = \frac{\varphi(p^k)}{\gcd(j, \varphi(p^k))}$$

which shows that $\mathrm{ord}_{p^k}(b) = 2^s t$ if and only if

$$2^s t = \frac{2^\mu \nu}{\gcd(j, 2^\mu \nu)}\,. \tag{6.12}$$

By (6.12) we can deduce that the case $s > \mu$ cannot occur because in that case we would have $\nu = 2^{s-\mu} t \cdot \gcd(j, 2^\mu \nu)$ and thus $2|\nu$, a contradiction to the assumption of an odd $\mu$ in $\varphi(p^k) = 2^\mu \nu$. Therefore, for the event "$\mathrm{ord}_{p^k}(b)/2^s$ is odd" to happen, we must have $s \leqslant \mu$.

Now consider the case $s \leqslant \mu$ (so that the event "$\mathrm{ord}_{p^k}(b)/2^s$ is odd" could happen). Suppose that $j = 2^\omega x$ for some odd $x$ (in the identification $b = a^j \bmod p^k$). Then

$$\gcd(j, 2^\mu \nu) = 2^{\min\{\omega, \mu\}} \prod_{p:\ \text{odd primes}} p^{\kappa_p} \tag{6.13}$$

with some $\kappa_p \in \mathbb{N} \cup \{0\}$. In order to have $\mathrm{ord}_{p^k}(b) = 2^s t$, using (6.12) we obtain that

$$\gcd(j, 2^\mu \nu) = 2^{\mu - s} \frac{\nu}{t} \,. \tag{6.14}$$

Since $\nu$ and $t$ are assumed odd, it follows that then $\nu/t$ has to be odd as well. It then follows from (6.13) and (6.14) that $\min\{\omega, \mu\} = \mu - s$ which shows $\omega = \mu - s$; thus $j$ takes the form $j = 2^{\mu-s}x$ with an odd $x$ and belong to $\big\{1, \cdots, \varphi(p^k)\big\}$. In this set there exist $2^s\nu$ multiples of $2^{\mu-s}$, namely

$$\big\{2^{\mu-s} \times 1, 2^{\mu-s} \times 2, \cdots, 2^{\mu-s} \times 2^s\nu\big\}.$$

Of these $2^s\nu$ multiples of $2^{\mu-s}$ only half are of the form $j = 2^{\mu-s}x$ with an odd $x$. Therefore, when $s \leqslant u$ the fact that all $j$ are chosen with the same probability implies that the probability of that $\mathrm{ord}_{p^k}(b)/2^s$ is an odd number is given by

$$\frac{\text{Number of possible } j \text{ of the form } j = 2^{\mu-s}x \text{ with } x \text{ odd}}{\text{Number of possible } j}$$

which, using that $s \leqslant \mu$, is not greater than $1/2$. ▫

Finally, we restate and prove our main theorem in this sub-section.

**Theorem 6.37.** *Let $N \in \mathbb{N}$ be odd with prime factorization $N = \prod_{j=1}^{J} p_j^{\nu_j}$, where $p_1, \cdots, p_J$ are distinct prime numbers. For a randomly chosen $b \in \mathbb{Z}_N^*$, the probability of that $r \equiv \mathrm{ord}_N(b)$ is even and $b^{r/2} + 1 \bmod N \neq 0$ is at least $1 - \frac{1}{2^{J-1}}$.*

*Proof.* Since by assumption $N$ is odd, all its prime factors $p_1, \cdots, p_J$ have to be odd as well, and we can apply Lemma 6.36 for their powers $p_j^{\nu_j}$. We establish the theorem by showing that the probability of that "$r$ is odd" or "$r$ is even but $b^{r/2} + 1 = 0$" mod $N$ is not greater than $\frac{1}{2^{J-1}}$.

By Theorem 6.35, we know that every $b \in \mathbb{Z}_N^*$ corresponds uniquely to a set of $b_j \in \mathbb{Z}_{n_j}^*$ with $1 \leqslant j \leqslant J$ and vice versa, where $n_j = p_j^{\nu_j}$ and $b_j \equiv b \bmod n_j$. An arbitrary selection of $b$ is thus equivalent to an arbitrary selection of the tuple $(b_1, \cdots, b_J) \in \mathbb{Z}_{n_1}^* \times \cdots \times \mathbb{Z}_{n_J}^*$.

Suppose that $r = \mathrm{ord}_N(b)$, $r_j = \mathrm{ord}_{n_j}(b_j)$ and write $r = 2^s t$, $r_j = 2^{s_j} t_j$ for some odd numbers $t$ and $t_j$. We first show that

$$r = \mathrm{lcm}(r_1, r_2, \cdots, r_J) \,, \tag{6.15}$$

where $\mathrm{lcm}(r_1, r_2, \cdots, r_J)$ denotes the least common multiple of $r_1$, $r_2$, $\cdots$, $r_J$. To see this, note that for any $k \in \mathbb{N}$,

$$b_j^k \bmod p_j^{\nu_j} = \left(b \bmod p_j^{\nu_j}\right)^k \bmod p_j^{\nu_j} = b^k \bmod p_j^{\nu_j} ;$$

thus $r_j$ is also the smallest natural number satisfying

$$b^{r_j} \equiv 1 \bmod p_j^{\nu_j} . \tag{6.16}$$

In other words, $r_j = \mathrm{ord}_{n_j}(b)$. By the definition of $r$ there exists $z \in \mathbb{N}$ such that

$$b^r = 1 + zN = 1 + z \prod_{j=1}^{J} p_j^{\nu_j} ,$$

thus $b^r \equiv 1 \bmod p_j^{\nu_j}$ for all $1 \leqslant j \leqslant J$. Theorem 6.29 then shows that $r_j | r$ for all $1 \leqslant j \leqslant J$ so that we have

$$\mathrm{lcm}(r_1, r_2, \cdots, r_J) | r . \tag{6.17}$$

Let $L \equiv \mathrm{lcm}(r_1, r_2, \cdots, r_J)$ and $1 \leqslant j \leqslant J$. By Theorem 6.29 again $L$ satisfies $b^L \equiv 1 \bmod p_j^{\nu_j}$; thus $p_j^{\nu_j}$ is a factor of $b^L - 1$. Since $p_1, \cdots, p_J$ are distinct primes, we find that the product of all $p_j^{\nu_j}$ is also a factor of $b^L - 1$. Therefore, $b^L \equiv 1 \bmod N$. Theorem 6.29 then implies that $r | L$. Together with (6.17), we conclude (6.15).

Next we show that

| the event "$r$ is odd" $\vee$ "$r$ is even but $b^{r/2} + 1 \equiv 0 \bmod N$" corresponds to a subset of the set $\left\{(s_1, \cdots, s_J) \,\middle|\, s_1 = s_2 = \cdots = s_J = s \text{ for some } s \in \mathbb{N} \cup \{0\}\right\}$. | (6.18) |
|---|---|

Using (6.15), we find that $r$ is odd if and only if all $r_j' s$ are odd. Therefore,

$$r \text{ is odd if and only if } s_j = 0 \text{ for all } 1 \leqslant j \leqslant J. \tag{6.19}$$

Now we consider the case that $r$ is even but $b^{r/2} + 1 \equiv 0 \bmod N$. Then there exists $\ell \in \mathbb{N}$ such that $b^{r/2} + 1 = \ell N$. Letting $\ell_j = \ell N / p_j^{\nu_j}$, we have

$$b^{r/2} + 1 = \ell_j p_j^{\nu_j} \qquad \forall\, 1 \leqslant j \leqslant J ;$$

thus

$$b^{r/2} + 1 \equiv 0 \bmod p_j^{\nu_j} . \tag{6.20}$$

On the other hand, note that (6.15) implies that $s_j \leqslant s$ for all $1 \leqslant j \leqslant J$. Suppose that $s_j < s$ for some $1 \leqslant j \leqslant J$. Then the fact that

$$2^s t = r = k_j r_j = k_j 2^{s_j} t_j$$

shows that $k_j = 2^{s-s_j} t/t_j$ is even. Let $z_j = k_j/2$. Then $\dfrac{r}{2} = z_j r_j$ with $z_j \in \mathbb{N}$; thus using (6.16) we find that

$$b^{r/2} \bmod p_j^{\nu_j} = b^{z_j r_j} \bmod p_j^{\nu_j} = \big(b^{r_j} \bmod p_j^{\nu_j}\big)^{z_j} \bmod p_j^{\nu_j} = 1 \bmod p_j^{\nu_j} = 1\,,$$

a contradiction to (6.20). Therefore, we must have $s_j = s$ for all $1 \leqslant j \leqslant J$ if $r$ is even but $b^{r/2} + 1 \equiv 0 \bmod N$. Together with (6.19), we conclude (6.18).

Since all $s_j$'s are chosen independently, using (6.18) Lemma 6.36 implies that

$$\begin{aligned}
\mathbf{P}(\text{“}r \text{ is odd”} \vee \text{“}b^{r/2} + 1 \equiv 0 \bmod N\text{”}) &\leqslant \sum_{s=0}^{\infty} \mathbf{P}(\text{“}s_j = s \text{ for all } 1 \leqslant j \leqslant J\text{”}) \\
&= \sum_{s=0}^{\infty} \prod_{j=1}^{J} \mathbf{P}(\text{“}s_j = s\text{”}) = \sum_{s=0}^{\infty} \mathbf{P}(\text{“}s_1 = s\text{”}) \prod_{j=2}^{J} \mathbf{P}(\text{“}s_j = s\text{”}) \\
&= \sum_{s=0}^{\infty} \mathbf{P}(\text{“}s_1 = s\text{”}) \prod_{j=2}^{J} \mathbf{P}\big(\text{“}r_j = 2^s t \text{ with an odd } t\text{”}\big) \\
&\leqslant \sum_{s=0}^{\infty} \mathbf{P}(\text{“}s_1 = s\text{”}) \frac{1}{2^{J-1}} = \frac{1}{2^{J-1}}\,.
\end{aligned}$$

Therefore, the probability of that $r \equiv \operatorname{ord}_N(b)$ is even and $b^{r/2} + 1 \bmod N \neq 0$ is at least $1 - \dfrac{1}{2^{J-1}}$. ▫

# Chapter 7

# Grover's Search Algorithm

## 7.1 The Problem

**The search problem**: For $N = 2^n$, we are given an arbitrary $x \in \{0,1\}^N$. The goal is to find an $i$ such that $x_i = 1$ (and to output 'no solutions' if there are no such $i$). We denote the number of solutions in $x$ by $t$ (that is, $t$ is the Hamming weight of $x$). This problem may be viewed as a simplification of the problem of searching an $N$-slot unordered database. Classically, a randomized algorithm would need $\mathcal{O}(N)$ queries to solve the search problem. Grover's algorithm solves it in $\mathcal{O}(\sqrt{N})$ queries, and $\mathcal{O}(\sqrt{N}\log N)$ other gates (the number of gates can be reduced a bit further, see Exercise 7).

## 7.2 Grover's Algorithm

Let $\mathrm{O}_{x,\pm}|i\rangle = (-1)^{x_i}|i\rangle$ denote the $\pm$-type oracle for the input $x$, and R be the unitary transformation that puts a $-1$ in front of all basis states $|i\rangle$ whenever $i \neq 0$, and that does nothing to the basis state $|0^n\rangle$. The Grover iterate is $\mathcal{G} = \mathrm{H}^{\otimes n}\mathrm{R}\mathrm{H}^{\otimes n}\mathrm{O}_{x,\pm}$. Note that 1 Grover iterate makes 1 query, and uses $\mathcal{O}(\log_2 N)$ other gates. Grover's algorithm starts in the $n$-bit state $|0^n\rangle$, applies a Hadamard transformation to each qubit to get the uniform superposition $|U\rangle = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle$ of all $N$ indices, applies $\mathcal{G}$ to this state $k$ times (for some $k$ to be chosen later), and then measures the final state. Intuitively, what happens is that in each iteration some amplitude is moved from the indices of the 0-bits to the indices of the 1-bits. The algorithm stops when almost all of the amplitude is on the 1-bits, in which case a measurement of the final state will probably give the index of a 1-bit. Figure 7.1

illustrates this.

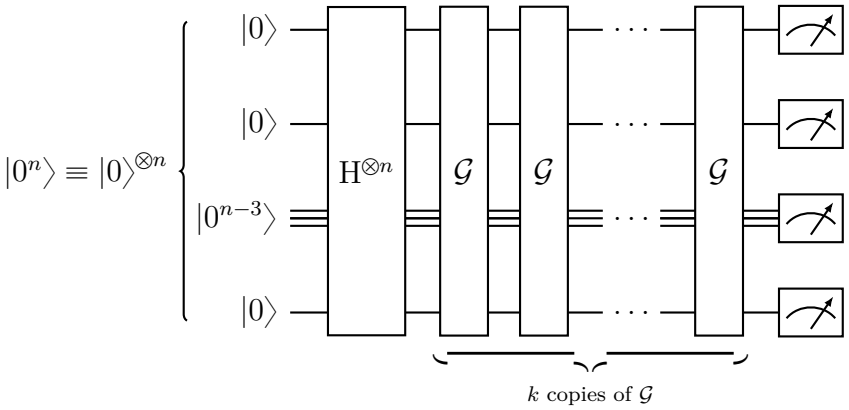



Figure 7.1: Grover's algorithm, with $k$ Grover iterates

To analyze this, define the following "good" and "bad" states:

$$|G\rangle = \frac{1}{\sqrt{\#\{i\,|\,x_i = 1\}}} \sum_{\{i\,|\,x_i=1\}} |i\rangle \qquad \text{and} \qquad |B\rangle = \frac{1}{\sqrt{\#\{i\,|\,x_i = 0\}}} \sum_{\{i\,|\,x_i=0\}} |i\rangle\,.$$

Let $t = \#\{i\,|\,x_i = 1\}$. Then the uniform state over all indices edges can be written as

$$\begin{aligned} |U\rangle &= \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle = \frac{1}{\sqrt{N}} \Big( \sum_{\{i\,|\,x_i=1\}} + \sum_{\{i\,|\,x_i=0\}} \Big) |i\rangle \\ &= \frac{1}{\sqrt{N}} \Big( \sqrt{t}|B\rangle + \sqrt{N-t}|G\rangle \Big) = \sin\theta|G\rangle + \cos\theta|B\rangle\,, \end{aligned}$$

where $\theta = \arcsin\sqrt{\dfrac{t}{N}}$.

The Grover iterate $\mathcal{G}$ is actually the product of two reflections (in the 2-dimensional space spanned by $|G\rangle$ and $|B\rangle$): $\mathrm{O}_{x,\pm}$ is a reflection through $|B\rangle$, and

$$\mathrm{H}^{\otimes n}\mathrm{R}\mathrm{H}^{\otimes n} = \mathrm{H}^{\otimes n}(2|0^n\rangle\langle 0^n| - \mathrm{I})\mathrm{H}^{\otimes n} = 2|U\rangle\langle U| - \mathrm{I}$$

is a reflection through $|U\rangle$. Here is Grover's algorithm restated, assuming we know the fraction of solutions is $\varepsilon = t/N$:

1. Set up the starting state $|U\rangle = \mathrm{H}^{\otimes n}|0^n\rangle$
2. Repeat the following $k = \mathcal{O}(1/\sqrt{\varepsilon})$ times:

(a) Reflect through $|B\rangle$ (that is, apply $\mathrm{O}_{x,\pm}$)

(b) Reflect through $|U\rangle$ (that is, apply $\mathrm{H}^{\otimes n}\mathrm{R}\mathrm{H}^{\otimes n}$).

3. Measure the first register and check that the resulting $i$ is a solution.

**Geometric argument**: There is a fairly simple geometric argument why the algorithm works. The analysis is in the 2-dimensional real plane spanned by $|B\rangle$ and $|G\rangle$. We start with $|U\rangle = \sin\theta|G\rangle + \cos\theta|B\rangle$: The two reflections (a) and (b) increase the angle from $\theta$ to $3\theta$, moving us towards the good state, as illustrated in Figure 7.2.

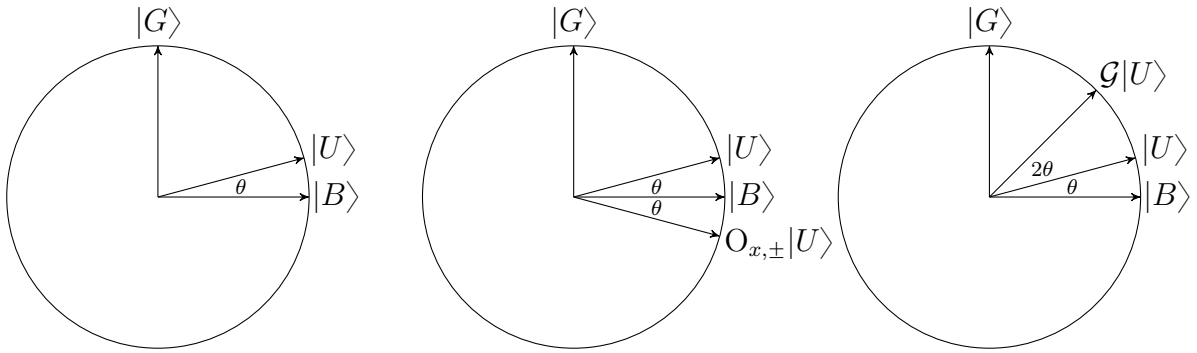



Figure 7.2: The first iteration of Grover: (left) start with $|U\rangle$, (middle) reflect through $|B\rangle$ to get $\mathrm{O}_{x,\pm}|U\rangle$, (right) reflect through $|U\rangle$ to get $\mathcal{G}|U\rangle$

The next two reflections (a) and (b) increase the angle with another $2\theta$, etc. More generally, after $k$ applications of (a) and (b) our state has become

$$\sin((2k+1)\theta)|G\rangle + \cos((2k+1)\theta)|B\rangle\,.$$

If we now measure, the probability of seeing a solution is $P_k = \sin^2((2k+1)\theta)$. We want $P_k$ to be as close to 1 as possible. Note that if we can choose $\widetilde{k} = \dfrac{\pi}{4\theta} - \dfrac{1}{2}$, then $(2\widetilde{k}+1)\theta = \dfrac{\pi}{2}$ and hence $P_{\widetilde{k}} = \sin^2\dfrac{\pi}{2} = 1$. An example where this works is if $t = N/4$, for then $\theta = \pi/6$ and $\widetilde{k} = 1$. Unfortunately $\widetilde{k} = \dfrac{\pi}{4\theta} - \dfrac{1}{2}$ will usually not be an integer, and we can only do an integer number of Grover iterations. However, if we choose $k$ to be the integer closest to $\widetilde{k}$, then our final state will still be close to $|G\rangle$ and the failure probability will still be small (assuming $t \ll N$):

$$\begin{aligned} 1 - P_k &= \cos^2((2k+1)\theta) = \cos^2((2\widetilde{k}+1)\theta + 2(k-\widetilde{k})\theta) = \cos^2\left(\frac{\pi}{2} + 2(k-\widetilde{k})\theta\right) \\ &= \sin^2(2(k-\widetilde{k})\theta) \leqslant \sin^2(\theta) = \frac{t}{N}\,, \end{aligned}$$

where we used $|k-\widetilde{k}| \leqslant 1/2$. Since $\arcsin(\theta) \geqslant \theta$, the number of queries is $k \leqslant \frac{\pi}{4\theta} \leqslant \frac{\pi}{4}\sqrt{\frac{N}{t}}$.

**Algebraic argument**: For those who do not like geometry, here is an alternative (but equivalent) algebraic argument. Let $a_k$ denote the amplitude of the indices of the $t$ 1-bits after $k$ Grover iterates, and $b_k$ the amplitude of the indices of the 0-bits. Initially, for the uniform superposition $|U\rangle$ we have $a_0 = b_0 = \frac{1}{\sqrt{N}}$. Since

$$\mathrm{H}^{\otimes n} = \mathrm{H}_n \qquad \text{and} \qquad \mathrm{R} = \mathrm{diag}(1,-1,-1,\cdots,-1)\,,$$

we find that

$$\mathrm{H}^{\otimes n}\mathrm{R}\mathrm{H}^{\otimes n} = \mathrm{H}_n \left( 2 \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} - \mathrm{I} \right) \mathrm{H}_n = \frac{2}{N} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} - \mathrm{I} \equiv \Big[\frac{2}{N}\Big] - \mathrm{I}\,,$$

where $\big[\frac{2}{N}\big]$ is the $N \times N$ matrix in which all entries are $\frac{2}{N}$; thus we find the following recursion:

$$\begin{aligned} a_{k+1} &= \frac{2}{N}\big[-ta_k + (N-t)b_k\big] + a_k = \frac{N-2t}{N}a_k + \frac{2(N-t)}{N}b_k\,, \\ b_{k+1} &= \frac{2}{N}\big[-ta_k + (N-t)b_k\big] - b_k = \frac{-2t}{N}a_k + \frac{N-2t}{N}b_k\,. \end{aligned}$$

With $\theta = \arcsin\sqrt{\frac{t}{N}}$ as before, we have

$$\begin{bmatrix} a_{k+1} \\ b_{k+1} \end{bmatrix} = \begin{bmatrix} \cos(2\theta) & 2\cos^2\theta \\ -2\sin^2\theta & \cos(2\theta) \end{bmatrix} \begin{bmatrix} a_k \\ b_k \end{bmatrix}.$$

Since

$$\begin{bmatrix} \cos(2\theta) & 2\cos^2\theta \\ -2\sin^2\theta & \cos(2\theta) \end{bmatrix} = \begin{bmatrix} \cos\theta & \cos\theta \\ i\sin\theta & -i\sin\theta \end{bmatrix} \begin{bmatrix} e^{2i\theta} & 0 \\ 0 & e^{-2i\theta} \end{bmatrix} \begin{bmatrix} \cos\theta & \cos\theta \\ i\sin\theta & -i\sin\theta \end{bmatrix}^{-1},$$

we have

$$\begin{bmatrix} \cos\theta & \cos\theta \\ i\sin\theta & -i\sin\theta \end{bmatrix}^{-1} \begin{bmatrix} a_{k+1} \\ b_{k+1} \end{bmatrix} = \begin{bmatrix} e^{2i\theta} & 0 \\ 0 & e^{-2i\theta} \end{bmatrix} \begin{bmatrix} \cos\theta & \cos\theta \\ i\sin\theta & -i\sin\theta \end{bmatrix}^{-1} \begin{bmatrix} a_k \\ b_k \end{bmatrix};$$

thus

$$\begin{aligned}\begin{bmatrix} a_k \\ b_k \end{bmatrix} &= \begin{bmatrix} \cos\theta & \cos\theta \\ i\sin\theta & -i\sin\theta \end{bmatrix} \begin{bmatrix} e^{2i\theta} & 0 \\ 0 & e^{-2i\theta} \end{bmatrix}^k \begin{bmatrix} \cos\theta & \cos\theta \\ i\sin\theta & -i\sin\theta \end{bmatrix}^{-1} \begin{bmatrix} a_0 \\ b_0 \end{bmatrix} \\ &= \frac{1}{\sqrt{N}} \begin{bmatrix} \cos\theta & \cos\theta \\ i\sin\theta & -i\sin\theta \end{bmatrix} \begin{bmatrix} e^{2ki\theta} & 0 \\ 0 & e^{-2ki\theta} \end{bmatrix} \frac{1}{-2i\sin\theta\cos\theta} \begin{bmatrix} -i\sin\theta & -\cos\theta \\ -i\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ &= \frac{1}{\sqrt{N}} \frac{1}{-2i\sin\theta\cos\theta} \begin{bmatrix} \cos\theta & \cos\theta \\ i\sin\theta & -i\sin\theta \end{bmatrix} \begin{bmatrix} -e^{(2k+1)i\theta} \\ e^{-(2k+1)i\theta} \end{bmatrix} = \frac{1}{\sqrt{N}} \begin{bmatrix} \sin(2k+1)\theta/\sin\theta \\ \cos(2k+1)\theta/\cos\theta \end{bmatrix} . \end{aligned}$$

Therefore, we obtain the following formula for $a_k$ and $b_k$:

$$a_k = \frac{1}{\sqrt{t}} \sin((2k+1)\theta) \qquad \text{and} \qquad b_k = \frac{1}{\sqrt{N-t}} \cos((2k+1)\theta) .$$

Accordingly, after $k$ iterations the success probability (the sum of squares of the amplitudes of the locations of the $t$ 1-bits) is the same as in the geometric analysis

$$P_k = t \cdot a_k^2 = \sin^2((2k+1)\theta) .$$

Thus we have a bounded-error quantum search algorithm with $\mathcal{O}(\sqrt{N/t})$ queries, assuming we know $t$. We now list (without full proofs) a number of useful variants of Grover:

1. If we know $t$ exactly, then the algorithm can be tweaked to end up in exactly the good state. Roughly speaking, you can make the angle $\theta$ slightly smaller, such that $\widetilde{k} = \frac{\pi}{4\theta} - \frac{1}{2}$ becomes an integer.

2. If we do not know $t$, then there is a problem: we do not know which $k$ to use, so we do not know when to stop doing the Grover iterates. Note that if $k$ gets too big, the success probability $P_k = \sin^2((2k+1)\theta))$ goes down again! However, a slightly more complicated algorithm (basically running the above algorithm with systematic different guesses for $k$) shows that an expected number of $\mathcal{O}(\sqrt{N/T})$ queries still suffices to find a solution if there are $t$ solutions. If there is no solution ($t = 0$), then we can easily detect that by checking $x_i$ for the $i$ that the algorithm outputs.

3. If we know a lower bound $\tau$ on the actual (possibly unknown) number of solutions $t$, then the above algorithm uses an expected number of $\mathcal{O}(\sqrt{N/\tau})$ queries. If we run this algorithm for up to three times its expected number of queries, then (by Markov's inequality) with probability at least 2/3 it will have found a solution. This way we can turn an expected runtime into a worst-case runtime.

4. If we do not know t but would like to reduce the probability of not finding a solution to some small $\varepsilon > 0$, then we can do this using $\mathcal{O}(\sqrt{N \log(1/\varepsilon)})$ queries. The important part here is that the $\log(1/\varepsilon)$ is inside the square-root; usual error reduction by $\mathcal{O}(\log(1/\varepsilon))$ repetitions of basic Grover would give the worse upper bound of $\mathcal{O}(\sqrt{N} \log(1/\varepsilon))$ queries.

## 7.3 Amplitude Amplification

The analysis that worked for Grover's algorithm is actually much more generally applicable. Let $\chi : \mathbb{Z} \to \{0, 1\}$ be any Boolean function; inputs $z \in \mathbb{Z}$ satisfying $\chi(z) = 1$ are called solutions. Suppose we have an algorithm to check whether $z$ is a solution. This can be written as a unitary $\mathrm{O}_\chi$ that maps $|z\rangle$ to $(-1)^{\chi(z)}|z\rangle$. Suppose also we have some (quantum or classical) algorithm $\mathcal{A}$ that uses no intermediate measurements and has probability $p$ of finding a solution when applied to starting state $|0\rangle$. Classically, we would have to repeat $\mathcal{A}$ roughly $1/p$ times before we find a solution. The amplitude amplification algorithm below only needs to run $\mathcal{A}$ and $\mathcal{A}^{-1}$ $\mathcal{O}(1/\sqrt{p})$ times:

1. Setup the starting state $|U\rangle = \mathcal{A}|0\rangle$.
2. Repeat the following $\mathcal{O}(1/\sqrt{p})$ times:
   (a) Reflect through $|B\rangle$ (that is, apply $\mathrm{O}_\chi$)
   (b) Reflect through $|U\rangle$ (that is, apply $\mathcal{A}\mathrm{R}\mathcal{A}^{-1}$)
3. Measure the first register and check that the resulting element $x$ is marked.

Defining $\theta = \arcsin\sqrt{p}$ and good and bad states $|G\rangle$ and $|B\rangle$ in analogy with the earlier geometric argument for Grover's algorithm, the same reasoning shows that amplitude amplification indeed finds a solution with high probability. This way, we can speed up a very large class of classical heuristic algorithms: any algorithm that has some non-trivial probability of finding a solution can be amplified to success probability nearly 1 (provided we can efficiently check solutions; that is, implement $\mathrm{O}_\chi$). Note that the Hadamard transform $\mathrm{H}^{\otimes n}$ can be viewed as an algorithm with success probability $p = t/N$ for a search problem of size $N$ with $t$ solutions, because $\mathrm{H}^{\otimes n}|0^n\rangle$ is the uniform superposition over all $N$ locations. Hence Grover's algorithm is a special case of amplitude amplification, where $\mathrm{O}_\chi = \mathrm{O}_{x,\pm}$ and $\mathcal{A} = \mathrm{H}^{\otimes n}$.

# Chapter 8

# The HHL Algorithm

## 8.1 The Linear System Problem

In this chapter we present the Harrow-Hassidim-Lloyd (HHL) algorithm for solving large systems of linear equations. Such a system is given by an $N \times N$ matrix $A$ with real or complex entries, and an $N$-dimensional nonzero vector $b$. Assume for simplicity that $N = 2^n$. The linear-system problem is

**LSP**: find an $N$-dimensional vector $x$ such that $Ax = b$.

Solving large systems of linear equations is extremely important in many computational problems in industry, in science, in optimization, in machine learning, etc. In many applications it suffices to find a vector $\widetilde{x}$ that is close to the actual solution $x$.

We will assume $A$ is invertible (equivalently, has rank $N$) in order to guarantee the existence of a unique solution vector $x$, which is then just $A^{-1}b$. This assumption is just for simplicity: if $A$ does not have full rank, then the methods below would still allow to invert it on its support, replacing $A^{-1}$ by the "Moore-Penrose pseudoinverse".

The HHL algorithm can solve "well-behaved" large linear systems very fast (under certain assumptions), but in a rather weak sense: instead of outputting the $N$-dimensional solution vector $x$ itself, its goal is to output the $n$-qubit state

$$|x\rangle = \frac{1}{\|x\|} \sum_{i=0}^{N-1} x_i |i\rangle$$

or some other $n$-qubit state close to $|x\rangle$. This state $|x\rangle$ has the solution vector as its vector of amplitudes, up to normalization. This is called the quantum linear-system problem:

**QLSP**: find an $n$-qubit state $|\widetilde{x}\rangle$ such that $\||x\rangle - |\widetilde{x}\rangle\| \leqslant \varepsilon$ and $Ax = b$.

Note that the QLSP is an inherently quantum problem, since the goal is to produce an $n$-qubit state whose amplitude-vector (up to normalization and up to $\varepsilon$-error) is a solution to the linear system. In general this is not as useful as just having the $N$-dimensional vector $x$ written out on a piece of paper, but in some cases where we only want some partial information about $x$, it may suffice to just (approximately) construct $|x\rangle$.

W.L.O.G. We will assume that $A$ is Hermitian: if $A$ is a non-hermitian $N \times N$ matrix, then we consider the augmented linear system (of size $2N$) $\bar{A}\bar{x} = \bar{b}$, where with $\mathbf{0}_{N\times N}$ denoting the $N \times N$ zeros matrix and $\mathbf{0}_{N\times 1}$ denoting the zero (column) vectors in $\mathbb{R}^N$,

$$\bar{A} \equiv \begin{bmatrix} \mathbf{0}_{N\times N} & A \\ A^\dagger & \mathbf{0}_{N\times N} \end{bmatrix}, \quad \bar{b} = \begin{bmatrix} b \\ \mathbf{0}_{N\times 1} \end{bmatrix}.$$

Note that if $x$ solves $Ax = b$ (or equivalently, $x = A^{-1}b$), then $\bar{x}$ takes the form $\bar{x} = \begin{bmatrix} \mathbf{0}_{N\times 1} \\ x \end{bmatrix}$.

Let us state the more restrictive assumptions that will make the linear system "well-behaved" and suitable for the HHL algorithm:

1. We have a unitary that can prepare the vector $b$ as an $n$-qubit quantum state
$$|b\rangle = \frac{1}{\|b\|} \sum_{i=0}^{N-1} b_i |i\rangle$$
using a circuit of $B$ 2-qubit gates. We also assume for simplicity that $\|b\| = 1$.

2. The matrix $A$ is $s$-sparse and we have sparse access to it. Such sparsity is not essential to the algorithm, and could be replaced by other properties that enable an efficient block-encoding of $A$.

3. The matrix $A$ is well-conditioned: the ratio between its largest and smallest singular value is at most some $\kappa$. For simplicity, assume the smallest singular value is not smaller than $1/\kappa$ while the largest is not greater than 1. In other words, all eigenvalues of $A$ lie in the interval $[-1, -1/\kappa] \cup [1/\kappa, 1]$. The smaller the "condition number" $\kappa$ is, the better it will be for the algorithm. Let us assume our algorithm knows $\kappa$, or at least knows a reasonable upper bound on $\kappa$.

## 8.2 The Basic HHL Algorithm for Linear Systems

Let us start with some intuition. The solution vector $x$ that we are looking for is $A^{-1}b$, so we would like to apply $A^{-1}$ to $b$. If $A$ has spectral decomposition $A = \sum_{j=0}^{N-1} \lambda_j a_j a_j^\dagger$, then the map

$A^{-1}$ is the same as the map $a_j \mapsto \frac{1}{\lambda_j} a_j$: we just want to multiply the eigenvector $a_j$ with the scalar $1/\lambda_j$. The vector $b$ can also be written as a linear combination of the eigenvectors $a_j$: $b = \sum_{j=0}^{N-1} \beta_j a_j$ (we do not need to know the coefficients $\beta_j$ for what follows). We want to apply $A^{-1}$ to $b$ to obtain $A^{-1}b = \sum_{j=0}^{N-1} \beta_j \frac{1}{\lambda_j} a_j$, normalized, as an $n$-qubit quantum state.

Unfortunately the maps $A$ and $A^{-1}$ are not unitary (unless $|\lambda_j| = 1$ for all $j$), so we cannot just apply $A^{-1}$ as a quantum operation to state $|b\rangle$ to get state $|x\rangle$. Fortunately $U = e^{2\pi i A} = \sum_{j=0}^{N-1} e^{2\pi i \lambda_j} a_j a_j^\dagger$ is unitary, and has the same eigenvectors as $A$ and $A^{-1}$. We can implement $U$ and powers of $U$ by Hamiltonian simulation, and then use phase estimation (Section 5.5) to estimate the $\lambda_j$ associated with eigenvector $|a_j\rangle$ with some small approximation error (for this sketch, assume for simplicity that the error is 0).

How does one invert the eigenvalues all together at the same time? This is done through a smart design of multi-controlled rotation gates. Conditioned on our estimate of $\lambda_j$ we can then rotate an auxiliary $|0\rangle$-qubit to

$$\sqrt{1 - \frac{1}{\kappa^2 \lambda_j^2}} |0\rangle + \frac{1}{\kappa \lambda_j} |1\rangle$$

(this is a valid state because $|\kappa \lambda_j| \geqslant 1$). Next we undo the phase estimation to set the register that contained the estimate back to $|0\rangle$. Suppressing the auxiliary qubits containing the temporary results of the phase estimation, we have now unitarily mapped

$$|a_j\rangle|0\rangle \mapsto |a_j\rangle \otimes \left( \sqrt{1 - \frac{1}{\kappa^2 \lambda_j^2}} |0\rangle + \frac{1}{\kappa \lambda_j} |1\rangle \right) .$$

If we prepare a copy of $|b\rangle|0\rangle = \sum_{j=0}^{N-1} \beta_j |a_j\rangle|0\rangle$ and apply the above unitary map to it, then we obtain

$$\sum_{j=0}^{N-1} \beta_j |a_j\rangle \left( \frac{1}{\kappa \lambda_j} |0\rangle + \sqrt{1 - \frac{1}{\kappa^2 \lambda_j^2}} |1\rangle \right) = \frac{1}{\kappa} \underbrace{\sum_{j=0}^{N-1} \beta_j \frac{1}{\lambda_j} |a_j\rangle|0\rangle}_{\propto |x\rangle} + |\phi\rangle|1\rangle ,$$

where we do not care about the (sub-normalized) state $|\phi\rangle$. Note that because $\sum_{j=0}^{N-1} |\beta_j/\lambda_j|^2 \geqslant \sum_{j=0}^{N-1} |\beta_j|^2 = 1$, the norm of the part of the state ending in qubit $|1\rangle$ is at least $1/\kappa^2$. Ac-

cordingly, we can now apply $\mathcal{O}(\kappa)$ rounds of amplitude amplification to amplify this part of the state to have amplitude essentially 1. This prepares state $|x\rangle$, as intended. This rough sketch is the basic idea of HHL. It leads to an algorithm that produces a state $|\widetilde{x}\rangle$ that is $\varepsilon$-close to $|x\rangle$, using roughly $\kappa^2 s/\varepsilon$ queries to H and roughly $\kappa s(\kappa n/\varepsilon + B)$ other 2-qubit gates.

### 8.2.1 Detailed quantum algorithm

The algorithm uses three quantum registers, all of them set to $|0\rangle$ at the beginning of the algorithm. One register, which we will denote with the sub-index $n_\ell$, is used to store a binary representation of the eigenvalues of $A$. A second register, denoted by $n_b$, contains the vector solution, and from now on $N = 2^{n_b}$. There is an extra register, for the auxiliary qubits. These are qubits used as intermediate steps in the individual computations but will be ignored in the following description since they are set to $|0\rangle$ at the beginning of each computation and restored back to the $|0\rangle$ state at the end of the individual operation.

The following is an outline of the HHL algorithm with a high-level drawing of the corresponding circuit. For simplicity all computations are assumed to be exact in the ensuing description.

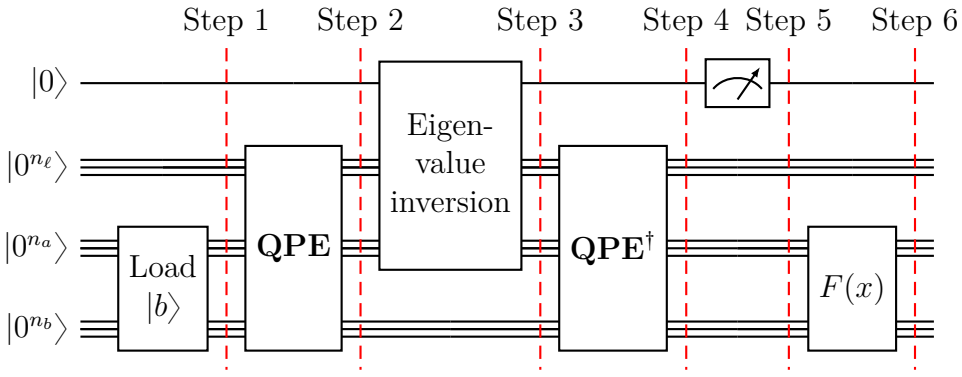



Figure 8.1: The quantum circuit of the HHL algorithm

**Step 1**: Load the data $|b\rangle \in \mathbb{C}^N$; that is, perform the transformation $|0^{n_b}\rangle \mapsto |b\rangle$.

**Step 2**: Apply Quantum Phase Estimation (QPE) with

$$U = e^{iAt} = \sum_{j=0}^{N-1} e^{i\lambda_j t} |a_j\rangle\langle a_j|$$

for a certain $t$ (here we take $t = 1$). The quantum state of the register expressed in the eigenbasis of $A$ is now $\sum_{j=0}^{N-1} b_j|\lambda_j\rangle_{n_\ell}|a_j\rangle$; that is,

$$\mathbf{QPE}(U, |0^{n_\ell}\rangle|b\rangle) = \sum_{j=0}^{N-1} b_j|\lambda_j\rangle_{n_\ell}|a_j\rangle\,.$$

Here we recall that $|\lambda_j\rangle_{n_\ell}$ is the $n_\ell$-bit binary approximation of $\lambda_j$ satisfying $|\lambda_j\rangle_{n_\ell} = |[2^n\lambda]\rangle$.

**Step 3**: Add an auxiliary qubit and apply a rotation conditioned on $|\lambda_j\rangle_{n_\ell}$,

$$\sum_{j=0}^{N-1} b_j|\lambda_j\rangle_{n_\ell}|a_j\rangle\left(\sqrt{1-\frac{1}{\kappa^2\lambda_j^2}}|0\rangle + \frac{1}{\kappa\lambda_j}|1\rangle\right)\,,$$

where $\kappa$ is (an upper bound of) the condition number of $A$.

**Step 4**: Apply $\mathbf{QPE}^\dagger$ (that is, undo **QPE**). Ignoring possible errors from **QPE**, this results in

$$\sum_{j=0}^{N-1} b_j|0^{n_\ell}\rangle|a_j\rangle\left(\sqrt{1-\frac{1}{\kappa^2\lambda_j^2}}|0\rangle + \frac{1}{\kappa\lambda_j}|1\rangle\right)\,.$$

**Step 5**: Measure the auxiliary qubit in the computational basis. If the outcome is 1, the register is in the post-measurement state

$$\left(\frac{1}{\sum_{j=0}^{N-1}|b_j|^2|\lambda_j|^{-2}}\right)^{\frac{1}{2}}\sum_{j=0}^{N-1}\frac{b_j}{\lambda_j}|0^{n_\ell}\rangle|a_j\rangle$$

which up to a normalisation factor corresponds to the solution.

**Step 6**: Apply an observable $M$ to calculate $F(x) \equiv \langle x|M|x\rangle$.

**Example 8.1.** Consider solving the linear system $Ax = b$, where

$$A = \begin{bmatrix} 1 & -1/3 \\ -1/3 & 1 \end{bmatrix} \qquad \text{and} \qquad |b\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}\,.$$

The solution $x = \begin{bmatrix} 9/8 & 3/8 \end{bmatrix}^{\mathrm{T}}$, and the corresponding **QLSP** is $A|x\rangle = |b\rangle$.

We will use $n_b = 1$ qubit to represent $|b\rangle$ and later the solution $|x\rangle$, $n_\ell = 2$ qubits to store the binary representation of the eigenvalues, and 1 auxiliary qubit to store whether the conditioned rotation, hence the algorithm, was successful.

For the purpose of illustrating the algorithm, we will cheat a bit and calculate the eigenvalues of $A$ to be able to choose $t$ to obtain an exact binary representation of the rescaled eigenvalues in the $n_\ell$-register. However, keep in mind that for the HHL algorithm implementation one does not need previous knowledge of the eigenvalues. Having said that, a short calculation will give $\lambda_1 = 2/3$ and $\lambda_2 = 4/3$.

Recall that the **QPE** will output an $n_\ell$-bit (2-bit in this case) binary approximation to $2^n \lambda_j t$. Therefore, if we set $t = \dfrac{3\pi}{4}$ the **QPE** will give a 2-bit binary approximation to $\dfrac{\lambda_1 t}{2\pi} = \dfrac{1}{4}$ and $\dfrac{\lambda_2 t}{2\pi} = \dfrac{1}{2}$, which is, respectively,

$$|01\rangle_{n_\ell} \qquad \text{and} \qquad |10\rangle_{n_\ell}\,.$$

The eigenvectors are, respectively,

$$|u_1\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix} \qquad \text{and} \qquad |u_2\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}\,.$$

Again, keep in mind that one does not need to compute the eigenvectors for the HHL implementation.

We can then write $|b\rangle$ in the eigenbasis of $A$ as

$$|b\rangle = \sum_{j=1}^{2} \frac{1}{\sqrt{2}}|a_j\rangle\,.$$

Now we are ready to go through the different steps of the HHL algorithm.

**Step 1**: State preparation in this example is trivial since $|b\rangle = |0\rangle$.

**Step 2**: Applying QPE will yield

$$\frac{1}{\sqrt{2}}|01\rangle|u_1\rangle + \frac{1}{\sqrt{2}}|10\rangle|u_2\rangle\,.$$

**Step 3**: Conditioned rotation with $\kappa = 8$ which is bigger than the exact condition number. Note, the constant $\kappa$ here needs to be chosen such that it is bigger than the smallest (rescaled) eigenvalue of 14 but as small as possible so that when the auxiliary qubit is

measured, the probability of it being in the state $|0\rangle$ is large:

$$\begin{aligned}
&\frac{1}{\sqrt{2}}|01\rangle|u_1\rangle\left(\frac{1}{8\cdot 1/4}|0\rangle+\sqrt{1-\frac{1}{8^2\cdot 1/4^2}}|1\rangle\right)\\
&\quad+\frac{1}{\sqrt{2}}|10\rangle|u_2\rangle\left(\frac{1}{8\cdot 1/2}|0\rangle+\sqrt{1-\frac{1}{8^2\cdot 1/2^2}}|1\rangle\right)\\
&\qquad=\frac{1}{\sqrt{2}}|01\rangle|u_1\rangle\left(\frac{1}{2}|0\rangle+\frac{\sqrt{3}}{2}|1\rangle\right)+\frac{1}{\sqrt{2}}|10\rangle|u_2\rangle\left(\frac{1}{4}|0\rangle+\frac{\sqrt{15}}{4}|1\rangle\right).
\end{aligned}$$

**Step 4**: After applying $\mathbf{QPE}^\dagger$ the quantum computer is in the state

$$\frac{1}{\sqrt{2}}|00\rangle|u_1\rangle\left(\frac{1}{2}|0\rangle+\frac{\sqrt{3}}{2}|1\rangle\right)+\frac{1}{\sqrt{2}}|00\rangle|u_2\rangle\left(\frac{1}{4}|0\rangle+\frac{\sqrt{15}}{4}|1\rangle\right).$$

**Step 5**: On outcome 0 when measuring the auxiliary qubit, the state is

$$\frac{1}{\sqrt{5/32}}\left(\frac{1}{\sqrt{2}}|00\rangle|u_1\rangle\frac{1}{2}|0\rangle+\frac{1}{\sqrt{2}}|00\rangle|u_2\rangle\frac{1}{4}|0\rangle\right).$$

A quick calculation shows that

$$\frac{1}{\sqrt{5/32}}\left(\frac{1}{2\sqrt{2}}|u_1\rangle+\frac{1}{4\sqrt{2}}|u_2\rangle\right)=\frac{|x\rangle}{\||x\rangle\|}.$$

**Step 6**: Without using extra gates, we can compute the norm of $|x\rangle$: it is the probability of measuring 0 in the auxiliary qubit from the previous step

$$P(|0\rangle)=\left(\frac{1}{2\sqrt{2}}\right)^2+\left(\frac{1}{4\sqrt{2}}\right)^2=\frac{5}{32}.$$