

Nhập môn Tính toán Lượng tử (Introduction to Quantum Computation)

Vũ Quốc Hoàng

Bản thảo 05 – 2025.

Hoan nghênh mọi ý kiến đóng góp!

(<https://github.com/vqhBook>)

Tưởng nhớ ông Vũ Đan Huyền,
cô Vũ Thị Phương Thảo.

Nơi miền ...
〈xa xăm|vương vấn〉

Lời cảm ơn

Tài liệu này không thể có nếu không có sự tham gia của các bạn: Trần Thị Thảo Nhi, Phan Thị Phương Uyên, Nguyễn Ngọc Toàn, Nguyễn Văn Quang Huy, Phan Tuấn Khải và đặc biệt là Lê Trọng Anh Tú.

Tài liệu này cũng không thể hoàn thành nếu không có sự hỗ trợ của thầy Nguyễn Đình Thúc, Trưởng Bộ môn Công nghệ Tri thức, Khoa Công nghệ Thông tin, Trường Đại học Khoa học Tự nhiên, TP Hồ Chí Minh.

Lời nói đầu

Tính toán lượng tử (Quatum Computation, QC) cùng với Trí tuệ nhân tạo (Artificial Intelligence, AI) là hai lĩnh vực đang nhận được nhiều sự quan tâm đầu tư và nghiên cứu. Nếu như AI đã đạt được nhiều thành tựu thì QC lại là công nghệ của tương lai. Sự phát triển của QC hứa hẹn tạo nên các đột phá về hạ tầng tính toán mà mọi hoạt động tính toán, trong đó có AI, đều sẽ thay đổi và được hưởng lợi.

Về mặt học thuật, nếu như AI là một “nồi lẩu thập cẩm” các kỹ thuật thì QC lại là một “món ăn thuần khiết” Toán học. Tài liệu này không bàn về triết học hay vật lý lượng tử và cũng không bàn về công nghệ hay máy tính lượng tử. Tài liệu này trình bày mô hình tính toán lượng tử và các cách vận dụng khéo léo mô hình này để tạo ra các thuật toán hiệu quả hơn nhiều so với mô hình tính toán cổ điển.

Tài liệu này được viết cho các sinh viên ngành Toán, Tin. Các sinh viên ngành Vật lý hay Kỹ thuật cũng có thể có lợi. Học sinh phổ thông (từ lớp 10) hay lập trình viên và mọi người nói chung muốn tìm hiểu (một cách nghiêm túc) về Tính toán lượng tử cũng có thể dùng. Các kiến thức về số phức và đại số tuyến tính sẽ giúp độc giả tiếp thu các nội dung nhanh chóng hơn. Tuy nhiên, mọi kiến thức Toán cần thiết đều được trình bày trong tài liệu.

Tài liệu này trình bày ngắn gọn các kiến thức và kỹ năng tính toán cần thiết để “bước vào thế giới tính toán lượng tử”. Các minh họa hay tính toán chi tiết được cho trong các Ví dụ. Các đề mục có dấu (*) là các nội dung “hơi khó”, có thể được bỏ qua trong lần đọc đầu tiên (và trở lại sau đó). Dấu (♣) đánh dấu các khẳng định mà độc giả nên kiểm tra (hay chứng minh) ngay lúc đọc. Việc làm các bài tập cuối mỗi chương giúp độc giả nắm chắc hơn các nội dung tương ứng.

Tài liệu này chắc chắn còn rất nhiều lỗi và thiếu sót. Các ý kiến đóng góp xin gửi về địa chỉ mail vqhoang.books@gmail.com hoặc/và thảo luận tại <https://github.com/vqhBook>. Được phép sao chép, chia sẻ tài liệu với mục đích học tập, nghiên cứu. Nghiêm cấm mọi hình thức kiếm tiền từ tài liệu này!

Vũ Quốc Hoàng,
Sài Gòn, 05 – 2025.

Mục lục

1	Số phức và vector, ma trận phức	1
1.1	Số phức	1
1.1.1	Khái niệm	1
1.1.2	Các phép toán	3
1.1.3	Dạng mũ	5
1.1.4	Căn và căn của đơn vị *	7
1.2	Vector phức	9
1.2.1	Vector	9
1.2.2	Chuẩn và tích vô hướng	11
1.2.3	Trực giao và chiếu trực giao	12
1.2.4	Kí pháp Dirac	12
1.2.5	Tổ hợp tuyến tính và cơ sở	14
1.3	Ma trận phức	17
1.3.1	Ma trận	17
1.3.2	Toán tử tuyến tính	21
1.3.3	Ma trận khả nghịch	24
1.3.4	Trị riêng và vector riêng *	25
1.3.5	Ma trận unita và ma trận Hermite	26
1.4	Tích tensor	28
	Bài tập	31
2	Tính toán cổ điển	35
2.1	Bit và mã hóa thông tin	35
2.1.1	Bit	35
2.1.2	Mã hóa thông tin	37
2.1.3	Số nhị phân	38
2.2	Cổng và mạch logic	41
2.2.1	Cổng logic	42
2.2.2	Mạch logic	45

2.2.3	Tập cổng toàn năng, đại số Boole và đơn giản mạch	48
2.2.4	Cổng khả nghịch	51
2.3	Bài toán, thuật toán và độ phức tạp tính toán	53
2.3.1	Bài toán và thuật toán	53
2.3.2	Độ phức tạp tính toán	54
2.4	Ngẫu nhiên, bất định và xác suất	59
2.4.1	Xác suất và thuật toán ngẫu nhiên	59
2.4.2	Trạng thái xác suất	61
2.4.3	Thao tác xác suất	64
2.4.4	Hệ nhiều thành phần	68
2.5	Dẫn nhập tính toán lượng tử	70
	Bài tập	75
3	Qubit	79
3.1	Qubit	79
3.1.1	Chồng chất	79
3.1.2	Phép đo	80
3.2	Pha chung, pha tương đối và mặt cầu Bloch	82
3.2.1	Pha chung	82
3.2.2	Pha tương đối	83
3.2.3	Mặt cầu Bloch	84
3.3	Phép toán, cổng và mạch lượng tử	87
3.3.1	Phép toán lượng tử	87
3.3.2	Cổng lượng tử	88
3.3.3	Mạch lượng tử	92
3.4	Qubit vật lý	94
	Bài tập	96
4	Hệ nhiều qubit	101
4.1	Hệ nhiều qubit	101
4.1.1	Trạng thái lượng tử	101
4.1.2	Vướng lượng tử	104
4.1.3	Phép đo theo một cơ sở trực chuẩn	106
4.1.4	Phép đo chiếu *	108
4.2	Phép toán và mạch lượng tử	111
4.3	Các cổng nhiều qubit thông dụng	113
4.3.1	Cổng CNOT	113
4.3.2	Cổng điều khiển	115
4.3.3	Cổng SWAP	115
4.3.4	Cổng Toffoli	116
4.3.5	Tập cổng lượng tử toàn năng	117

4.4	Chuyển mạch logic thành mạch lượng tử	118
4.4.1	Phiên bản lượng tử của các cổng logic thông dụng	118
4.4.2	“Gỡ tính toán”	120
4.5	Định lý không nhân bản	121
4.6	Bất đẳng thức Bell *	123
	Bài tập	127
5	Các thuật toán lượng tử cơ bản	137
5.1	Mã siêu đặc và dịch chuyển lượng tử	137
5.1.1	Nhắc lại vướng lượng tử	137
5.1.2	Mã siêu đặc	138
5.1.3	Dịch chuyển lượng tử	140
5.2	Trao đổi khóa lượng tử	143
5.2.1	Mã hoá và trao đổi khóa	143
5.2.2	Hệ mã hoá khoá công khai	145
5.2.3	Giao thức BB84	146
5.2.4	Giao thức B92	149
5.2.5	Giao thức E91	150
5.3	Các thuật toán truy vấn lượng tử	151
5.3.1	Oracle lượng tử và độ phức tạp truy vấn	152
5.3.2	Thuật toán Deutsch	153
5.3.3	Thuật toán Deutsch-Jozsa	155
5.3.4	Thuật toán Bernstein-Vazirani	158
5.3.5	Thuật toán Simon	159
5.4	Thuật toán Grover	162
5.4.1	Tìm kiếm vết cạn	163
5.4.2	Thuật toán Grover	164
5.4.3	Phản xạ qua trạng thái tổ hợp đều	168
	Bài tập	169
6	Các thuật toán lượng tử nâng cao	177
6.1	Biến đổi Fourier lượng tử	177
6.1.1	Mã pha và biến đổi Fourier lượng tử	177
6.1.2	Mạch biến đổi Fourier lượng tử	182
6.1.3	Biến đổi Fourier lượng tử ngược	185
6.2	Ước lượng pha	185
6.2.1	Bài toán ước lượng pha	186
6.2.2	Ước lượng pha với 1 qubit	186
6.2.3	Ước lượng pha với 2 qubit	189
6.2.4	Thủ tục ước lượng pha	192
6.3	Thuật toán Shor *	195

6.3.1	Phép toán lũy thừa modulo	195
6.3.2	Chu kỳ lũy thừa modulo	197
6.3.3	Mạch lượng tử nhân modulo	198
6.3.4	Thuật toán lượng tử tìm chu kỳ lũy thừa modulo	200
6.3.5	Phân tích số nguyên	203
6.4	Mã Shor	204
6.4.1	Mã sửa lỗi	204
6.4.2	Mã lặp	206
6.4.3	Mã sửa lỗi trong tính toán lượng tử	207
6.4.4	Mã lật bit	208
6.4.5	Mã lật pha	210
6.4.6	Mã Shor	212
	Bài tập	215
Phụ lục A Tập hợp, tích Descartes và ánh xạ		221
A.1	Tập hợp	221
A.1.1	Khái niệm	221
A.1.2	Các phép toán	222
A.2	Tích Descartes	224
A.3	Ánh xạ	225
A.3.1	Khái niệm	225
A.3.2	Đơn ánh, toàn ánh và song ánh	226
A.3.3	Hàm hợp	226
A.3.4	Hàm ngược	227
A.3.5	Lực lượng của tập hợp	228
Appendices		
Phụ lục B Xác suất		229
B.1	Xác suất	229
B.1.1	Không gian mẫu và biến cố	229
B.1.2	Độ đo xác suất	230
B.1.3	Xác suất rời rạc	231
B.2	Xác suất có điều kiện	232
B.2.1	Định nghĩa	232
B.2.2	Công thức nhân, toàn phần và công thức Bayes	233
B.2.3	Các biến cố độc lập	234
B.3	Biến ngẫu nhiên rời rạc	236
B.3.1	Định nghĩa	236
B.3.2	Kỳ vọng và phương sai	237
B.3.3	Phân phối đồng thời	238

B.3.4	Các biến ngẫu nhiên độc lập	239
B.3.5	Hiệp phương sai và hệ số tương quan	239
B.3.6	Hàm đặc trưng của biến cố	240
B.3.7	Các phân phối rời rạc thông dụng	241
Phụ lục C	Các trạng thái và cổng lượng tử thông dụng	245
C.1	Các trạng thái lượng tử thông dụng	245
C.1.1	1 qubit	245
C.1.2	2 qubit	246
C.1.3	3 qubit	246
C.1.4	n qubit	246
C.2	Các cổng lượng tử thông dụng	247
C.2.1	1 qubit	247
C.2.2	3 qubit	247
C.2.3	2 qubit	248
Tài liệu tham khảo		249

Chương 1

Số phức và vector, ma trận phức

Số phức không những quan trọng mà còn là nền tảng cốt yếu của việc hiểu và thực hành tính toán lượng tử. Chương này chuẩn bị các kiến thức số phức cần thiết cho tính toán lượng tử. Các kiến thức Toán cơ bản nhất như tập hợp, tích Descartes và ánh xạ được nhắc lại ở Phụ lục [A](#).

1.1 Số phức

Tập số thực \mathbb{R} với các phép toán (cộng, nhân, ...) tạo nên một cấu trúc tính toán đẹp và hiệu quả. Tuy nhiên, cấu trúc này vẫn còn “khiếm khuyết”: ta không thể khai căn các số thực âm, chẳng hạn, không có số thực x nào thỏa mãn

$$x^2 + 1 = 0 \Leftrightarrow x^2 = -1.$$

Ta muốn có một hệ thống số, kí hiệu là \mathbb{C} , mở rộng \mathbb{R} với các phép toán đẹp tương tự và dễ dàng khai căn.

Để giải quyết vấn đề này, ta cho phép một đối tượng đặc biệt, kí hiệu là i có tính chất

$$i^2 = -1.$$

Bổ sung “số” mới i này vào \mathbb{C} cùng với tất cả các số thực. Hơn nữa, do yêu cầu của các phép toán, ta cũng cần thêm các số khác vào \mathbb{C} như

$$2i, -1.5i, \dots, 1 + 2i, 2 - 1.5i, \dots$$

1.1.1 Khái niệm

Một **số phức** (complex number) z là một biểu thức có dạng

$$z = a + b \times i = a + bi, \quad a, b \in \mathbb{R},$$

trong đó

- i là kí hiệu có tính chất $i^2 = -1$, được gọi là **đơn vị ảo** (imaginary unit),¹
- a được gọi là **phần thực** (real part) của z , kí hiệu $\operatorname{Re}(z)$,
- b được gọi là **phần ảo** (imaginary part) của z , kí hiệu $\operatorname{Im}(z)$.

Tập tất cả các số phức được kí hiệu là \mathbb{C} . Hai số phức bằng nhau ($=$) khi chúng có cùng phần thực và phần ảo. Các **số thực** (real number) được xem là số phức có phần ảo là 0 (dạng $a+0i = a$). Các số phức có phần thực là 0 (dạng $0 + bi = bi$) được gọi là các **số ảo** (imaginary number).

Cho $z = a + bi \in \mathbb{C}$

- số thực không âm $\rho = |z| = \sqrt{a^2 + b^2}$ được gọi là **độ lớn** (magnitude, absolute value, modulus) của z ,
- số thực $\theta = \arg z = \arctan \frac{b}{a}$ được gọi là **pha** (phase, argument) của z ,
- số phức $\bar{z} = a - bi$ được gọi là số **liên hợp** (complex conjugate) của z .²

Nhận xét (♣)

- $\bar{\bar{z}} = z$,
- $\operatorname{Re}(\bar{z}) = \operatorname{Re}(z)$, $\operatorname{Im}(\bar{z}) = -\operatorname{Im}(z)$, $|\bar{z}| = |z|$, $\arg \bar{z} = -\arg z$,
- $\bar{z} = z$ khi và chỉ khi z là số thực,
- $|z| = 0$ khi và chỉ khi $z = 0$.

Các số phức có thể được mô tả trực quan trên **mặt phẳng phức** (complex plane) như minh họa trong Hình 1.1. Số phức $z = a + bi$ được biểu diễn như một điểm hay một vector với hoành độ, tung độ lần lượt là a, b . Độ lớn ρ là khoảng cách đến gốc, pha θ là góc tạo với trục hoành và liên hợp là điểm đối xứng qua trục hoành. Tất cả các số thực nằm trên trục hoành nên trục hoành được gọi là trục thực (Re), tất cả các số ảo nằm trên trục tung nên trục tung được gọi là trục ảo (Im).

Các số phức

$$z = \cos \theta + i \sin \theta, \theta \in [0, 2\pi)$$

có

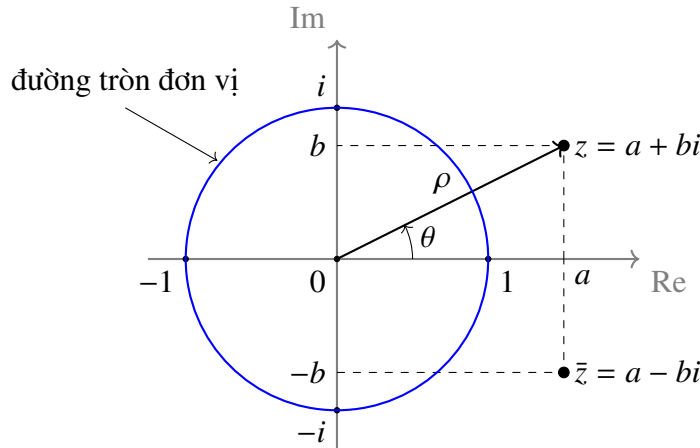
$$|z| = \sqrt{\cos^2 \theta + \sin^2 \theta} = 1, \quad \arg z = \theta,$$

thường được gọi là các số đơn vị. Các số này nằm trên **đường tròn đơn vị** (unit circle). Trong đó $-1, 1$ là các số thực đơn vị và $-i, i$ là các số ảo đơn vị.

¹để dễ nhớ, i là chữ đầu của imaginary; vì $i^2 = -1$ nên đôi khi i còn được kí hiệu là $\sqrt{-1}$.

²liên hợp của z còn được kí hiệu là z^* .

Đặc biệt, số $0 = 0 + 0i$ là số duy nhất có độ lớn 0 và cũng là số duy nhất vừa là số thực vừa là số ảo.



Hình 1.1: Mặt phẳng phức.

Ví dụ 1.1.1. Xét số phức $z = 1 - i = 1 + (-1)i$

- $\operatorname{Re}(z) = 1, \operatorname{Im}(z) = -1$,
- $|z| = \sqrt{1^2 + (-1)^2} = \sqrt{2}, \arg z = \arctan \frac{-1}{1} = -\frac{\pi}{4} (-45^\circ)$,
- $\bar{z} = 1 + i$.

□

1.1.2 Các phép toán

Các phép toán trên số phức là mở rộng “tự nhiên” của các phép toán tương ứng trên số thực và chúng cũng có các tính chất “đẹp” tương tự.

Cho $z_1 = a_1 + b_1i, z_2 = a_2 + b_2i \in \mathbb{C}$, phép toán **cộng** (addition) và **nhân** (multiplication) được định nghĩa là

- $z_1 + z_2 = (a_1 + b_1i) + (a_2 + b_2i) = (a_1 + a_2) + (b_1i + b_2i) = (a_1 + a_2) + (b_1 + b_2)i$,
- $z_1 \times z_2 = z_1 z_2 = (a_1 + b_1i)(a_2 + b_2i) = (a_1 a_2 + b_1 b_2 i^2) + (a_1 b_2 i + a_2 b_1 i) = (a_1 a_2 - b_1 b_2) + (a_1 b_2 + a_2 b_1)i$.

Cho $z = a + bi \in \mathbb{C}$

- **đối** (opposite) của z là $-z = -a - bi$,

- $z\bar{z} = (a + bi)(a - bi) = a^2 + b^2 = |z|^2$, từ đó, nếu $z \neq 0$ thì **nghịch đảo** (reciprocal) của z là

$$z^{-1} = \frac{1}{z} = \frac{\bar{z}}{z\bar{z}} = \frac{\bar{z}}{|z|^2} = \frac{a - bi}{a^2 + b^2} = \frac{a}{a^2 + b^2} - \frac{b}{a^2 + b^2}i.$$

Cho $z_1, z_2 \in \mathbb{C}$, phép toán **trừ** (subtraction) và **chia** (division) được định nghĩa là

$$\begin{aligned} z_1 - z_2 &= z_1 + (-z_2), \\ \frac{z_1}{z_2} &= z_1 z_2^{-1} \quad (z_2 \neq 0). \end{aligned}$$

Ví dụ 1.1.2. Xét $x = 1 - i, y = 2 + 2i$

- $x + y = (1 - i) + (2 + 2i) = 3 + i,$
- $x - y = (1 - i) + -(2 + 2i) = (1 - i) + (-2 - 2i) = -1 - 3i,$
- $xy = (1 - i)(2 + 2i) = 2 + 2i - 2i - 2i^2 = 2 + 0i - 2(-1) = 4,$
- $x^{-1} = \frac{1}{x} = \frac{\bar{x}}{x\bar{x}} = \frac{\bar{x}}{|x|^2} = \frac{1+i}{1^2+(-1)^2} = \frac{1+i}{2} = \frac{1}{2} + \frac{1}{2}i,$
- $\frac{y}{x} = yx^{-1} = (2 + 2i)(\frac{1}{2} + \frac{1}{2}i) = (1 - 1) + (i + i) = 2i,$
- $\left(\frac{y}{x}\right)^{-1} = \frac{-2i}{4} = \frac{-i}{2},$
- $\frac{x}{y} = \frac{x\bar{y}}{y\bar{y}} = \frac{x\bar{y}}{|y|^2} = \frac{(1-i)(2-2i)}{2^2+2^2} = \frac{(2-2)+(-2i-2i)}{8} = \frac{-i}{2} = \left(\frac{y}{x}\right)^{-1}.$

□

Tập các số phức \mathbb{C} bao gồm 0, 1 với các phép toán được định nghĩa ở trên thỏa các tính chất (\clubsuit)³

- Trung hòa: $z + 0 = z, \forall z \in \mathbb{C},$
- Có đối: $z + (-z) = 0, \forall z \in \mathbb{C},$
- Đơn vị: $z1 = z, \forall z \in \mathbb{C},$
- Có nghịch đảo: $zz^{-1} = 1, \forall z \in \mathbb{C}, z \neq 0,$
- Giao hoán: $x + y = y + x, xy = yx, \forall xy \in \mathbb{C},$
- Kết hợp: $(x + y) + z = x + (y + z), (xy)z = x(yz), \forall xyz \in \mathbb{C},$
- Phân phối: $x(y + z) = xy + xz, \forall xyz \in \mathbb{C}.$

³Trong Toán, \mathbb{C} còn được gọi là một **trường** (field), tương tự như \mathbb{R} .

Ngoài ra, với mọi $x, y \in \mathbb{C}$ (\clubsuit)

- $\overline{x + y} = \bar{x} + \bar{y}$,
- $\overline{x^{-1}} = (\bar{x})^{-1}$ ($x \neq 0$).
- $\overline{xy} = \bar{x}\bar{y}$, $\overline{\left(\frac{x}{y}\right)} = \frac{\bar{x}}{\bar{y}}$ ($y \neq 0$).

Ví dụ 1.1.3. Xét phương trình bậc hai

$$x^2 + 2x + 5 = 0 \quad (*)$$

(*) có thể được viết lại là

$$x^2 + 2x + 1 + 4 = 0 \Leftrightarrow (x + 1)^2 + 4 = 0 \Leftrightarrow (x + 1)^2 = -4.$$

(*) vô nghiệm trên \mathbb{R} vì $(x + 1)^2 \geq 0, \forall x \in \mathbb{R}$.

Tuy nhiên, (*) có nghiệm trên \mathbb{C} . Thật vậy, vì có đúng 2 số phức có bình phương bằng -4 là $2i$ và $-2i$ (\clubsuit) nên

$$(*) \Leftrightarrow \begin{cases} x + 1 = 2i \\ x + 1 = -2i \end{cases} \Leftrightarrow \begin{cases} x = -1 + 2i \\ x = -1 - 2i \end{cases}.$$

□

1.1.3 Dạng mũ

Số phức $z = a + bi$ được xác định hoàn toàn bởi độ lớn ρ và pha θ

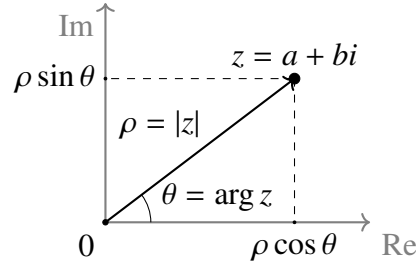
$$\begin{cases} \rho = |z| = \sqrt{a^2 + b^2} \\ \theta = \arg z = \arctan \frac{b}{a} \end{cases} \Leftrightarrow \begin{cases} a = \operatorname{Re}(z) = \rho \cos \theta \\ b = \operatorname{Im}(z) = \rho \sin \theta \end{cases}.$$

Hơn nữa, **công thức Euler** (Euler's formula) cho

$$e^{i\theta} = \cos \theta + i \sin \theta.$$

Do đó, số phức z có thể được mô tả theo

- **dạng đại số** (algebraic form): $z = a + bi$,
- **dạng cực** (polar form): $z = \rho(\cos \theta + i \sin \theta)$,
- **dạng mũ** (exponential form): $z = \rho e^{i\theta}$.



Hình 1.2: Quan hệ giữa phần thực, phần ảo và độ lớn, pha.

Quan hệ giữa phần thực, phần ảo và độ lớn, pha được mô tả trong Hình 1.2. Lưu ý, pha θ xác định góc nên ta xem 2 pha θ_1, θ_2 là như nhau nếu có $k \in \mathbb{Z}$ sao cho

$$\theta_1 - \theta_2 = k2\pi.$$

Thông thường, ta chọn $\theta \in [0, 2\pi)$ hoặc $\theta \in (-\pi, \pi]$. Cũng lưu ý, số 0 có độ lớn $\rho = 0$ còn pha θ không xác định (tùy ý).

Dạng đại số cho thấy rõ phần thực và phần ảo của số phức, phù hợp khi thực hiện các phép toán cộng, trừ, ... Ngược lại, dạng mũ cho thấy rõ độ lớn và pha, phù hợp khi nhân, chia, ... Cụ thể, cho $z_1 = \rho_1 e^{i\theta_1}, z_2 = \rho_2 e^{i\theta_2}$ (♣)

- $z_1 z_2 = \rho_1 \rho_2 e^{i\theta_1} e^{i\theta_2} = \rho_1 \rho_2 e^{i(\theta_1 + \theta_2)}$, tức là

$$|z_1 z_2| = |z_1| |z_2|, \quad \arg z_1 z_2 = \arg z_1 + \arg z_2,$$

- Nếu $z_2 \neq 0$ ($\rho_2 \neq 0$), $\frac{z_1}{z_2} = \frac{\rho_1 e^{i\theta_1}}{\rho_2 e^{i\theta_2}} = \frac{\rho_1}{\rho_2} e^{i(\theta_1 - \theta_2)}$, tức là

$$\left| \frac{z_1}{z_2} \right| = \frac{|z_1|}{|z_2|}, \quad \arg \frac{z_1}{z_2} = \arg z_1 - \arg z_2.$$

Cho $z = \rho e^{i\theta}, n \in \mathbb{N}$, **lũy thừa bậc n** (n -th power) của z , kí hiệu z^n , được định nghĩa là

$$z^n = \underbrace{z z \dots z}_{n \text{ thừa số}} \quad (z^0 = 1, z^1 = z).$$

Dễ thấy (♣)

$$z^n = \rho^n e^{in\theta}.$$

Hơn nữa, khi $z \neq 0$, với $k \in \mathbb{Z}_{<0}, k = -n$, ta định nghĩa

$$z^k = z^{-n} = (z^n)^{-1} = \frac{1}{\rho^n} e^{-in\theta}.$$

Đặc biệt, các số phức z đơn vị có $|z| = 1$, được xác định hoàn toàn bằng pha $z = e^{i\theta}$. Tính toán với các số này là tính toán trên pha, chẳng hạn, nhân $x = \rho e^{i\phi}$ với $z = e^{i\theta}$ được

$$xz = \rho e^{i\phi} e^{i\theta} = \rho e^{i(\phi+\theta)},$$

tức là, ta đã “xoay” pha của x một góc θ (và giữ nguyên độ lớn của x).

Nhờ khả năng mô tả cô động độ lớn và pha trong một con số mà số phức được dùng nhiều trong các tính toán liên quan đến các đại lượng biến thiên tuần hoàn (như sóng trong Vật lý). Một chu kỳ tuần hoàn tương ứng với góc 2π (cũng là một đường tròn hay góc 360°). Hai số cùng pha là hơn kém nhau bội của 2π còn ngược pha là hơn kém nhau bội lẻ của π . Nếu như giữa các số thực chỉ có 2 trường hợp pha này, cùng dấu (cùng pha) và ngược dấu (ngược pha), thì giữa các số phức có thể có vô hạn các trường hợp khác.

Ví dụ 1.1.4. (tiếp Ví dụ 1.1.2) Với $x = 1 - i$, $|x| = \sqrt{2}$, $\arg x = \arctan \frac{-1}{1} = -\frac{\pi}{4}$ nên

$$x = \sqrt{2}e^{-i\frac{\pi}{4}}.$$

Với $y = 2 + 2i$, $|y| = 2\sqrt{2}$, $\arg y = \arctan \frac{2}{2} = \frac{\pi}{4}$ nên

$$y = 2\sqrt{2}e^{i\frac{\pi}{4}}.$$

Do đó

- $xy = \sqrt{2}e^{-i\frac{\pi}{4}} 2\sqrt{2}e^{i\frac{\pi}{4}} = 4e^{i(-\frac{\pi}{4}+\frac{\pi}{4})} = 4e^0 = 4$,
- $x^{-1} = (\sqrt{2}e^{-i\frac{\pi}{4}})^{-1} = (\sqrt{2})^{-1} (e^{-i\frac{\pi}{4}})^{-1} = \frac{1}{\sqrt{2}}e^{i\frac{\pi}{4}} = \frac{1}{\sqrt{2}}(\cos \frac{\pi}{4} + i \sin \frac{\pi}{4}) = \frac{1}{\sqrt{2}}(\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}i) = \frac{1}{2} + \frac{1}{2}i$,
- $\frac{y}{x} = \frac{2\sqrt{2}e^{i\frac{\pi}{4}}}{\sqrt{2}e^{-i\frac{\pi}{4}}} = 2e^{i\frac{\pi}{4}+i\frac{\pi}{4}} = 2e^{i\frac{\pi}{2}} = 2(\cos \frac{\pi}{2} + i \sin \frac{\pi}{2}) = 2i$,
- $x^4 = (\sqrt{2}e^{-i\frac{\pi}{4}})^4 = (\sqrt{2})^4 (e^{-i\frac{\pi}{4}})^4 = 4e^{4(-i\frac{\pi}{4})} = 4e^{-i\pi} = -4$,
- $x^{-4} = (x^4)^{-1} = \frac{1}{x^4} = -\frac{1}{4}$.

□

1.1.4 Căn và căn của đơn vị *

Cho $z = \rho e^{i\theta} \in \mathbb{C}$, $n \in \mathbb{N}_{>0}$, **căn bậc n** (n -th root) của z , kí hiệu $\sqrt[n]{z}$, được định nghĩa là các số x thỏa $x^n = z$. Dễ thấy (♣)

$$\sqrt[n]{z} = \sqrt[n]{\rho} e^{i(\frac{1}{n}(\theta + k2\pi))} = \sqrt[n]{\rho} e^{i(\frac{\theta}{n} + \frac{k2\pi}{n})}, k = 0, \dots, n-1.$$

Như vậy, mọi số phức $z \neq 0$ đều có đúng n căn bậc n khác nhau. Đặc biệt, n số căn bậc n của 1

$$\omega_n^{(k)} = e^{i\frac{k2\pi}{n}} = \left(e^{i\frac{2\pi}{n}}\right)^k, k = 0, \dots, n-1$$

thường được gọi là các **số căn của đơn vị** (root of unity).

Đặt $\omega_n = \omega_n^{(1)} = e^{i\frac{2\pi}{n}}$, ta có các số căn bậc n của đơn vị lần lượt là

$$\omega_n^0 = 1, \omega_n^1 = \omega_n, \omega_n^2, \dots, \omega_n^{n-1}.$$

Nhận xét, với mọi $j, k \in \mathbb{Z}$ (\clubsuit)

- $\omega_n^j \omega_n^k = \omega_n^{j+k},$
- $\omega_n^j \omega_n^{n-j} = \omega_n^n = 1,$
- $(\omega_n^j)^{-1} = \omega_n^{n-j} = \overline{\omega_n^j}.$

Cho các số nguyên dương $1 < k < n$, xét ánh xạ $g_n^k : \mathbb{R} \rightarrow \mathbb{C}$ được định nghĩa là

$$g_n^k(t) = (\omega_n^k)^t = e^{i\frac{k2\pi}{n}t}.$$

Tại mỗi thời điểm t , $g_n^k(t)$ là một số phức nằm trên đường tròn đơn vị. Có thể nói $g_n^k(t)$ mô tả cho một chuyển động tròn đều có vận tốc góc, chu kỳ và tần số lần lượt là

$$\theta = \frac{k2\pi}{n} \text{ rad/s}, \quad T = \frac{n}{k} \text{ s}, \quad f = \frac{k}{n} \text{ Hz}.$$

Chuyển động này có thể là sự chuyển động vật lý của một chất điểm hoặc là sự thay đổi tuần hoàn của một đại lượng trừu tượng nào đó.

Đặc biệt, $g_n^1(t)$ lần lượt là các căn bậc n của đơn vị khi cho $t = 0, 1, 2, \dots$ và $g_n^k(t)$ mô tả cho biến đổi tuần hoàn có tần số nhanh gấp k lần tần số của $g_n^1(t)$.

Ví dụ 1.1.5. Số $x = 1 - i = \sqrt{2}e^{-i\frac{\pi}{4}}$ có đúng 4 căn bậc 4 là

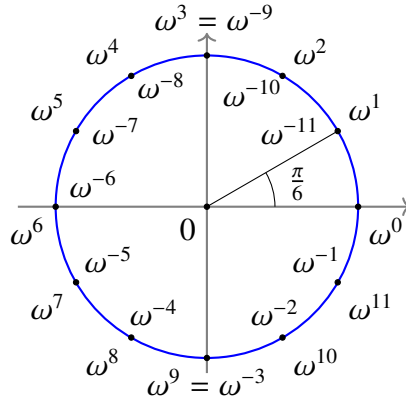
$$\sqrt[4]{x} = \left(\sqrt{2}e^{-i\frac{\pi}{4}}\right)^{\frac{1}{4}} = 2^{\frac{1}{8}}e^{\frac{1}{4}i(-\frac{\pi}{4}+k2\pi)} = \sqrt[8]{2}e^{i(-\frac{\pi}{16}+\frac{k\pi}{2})}, k = 0, \dots, 3.$$

12 số căn bậc 12 của đơn vị là

$$\omega_{12}^k = \left(e^{i\frac{\pi}{6}}\right)^k, k = 0, \dots, 11.$$

Các số này “nằm cách đều” trên đường tròn đơn vị như minh họa ở Hình 1.3.

Ta có thể xem $t \mapsto \omega_{12}^t$ mô tả cho một biến đổi tuần hoàn với tần số $f = \frac{1}{12}$ Hz ($\frac{1}{12}$ chu kỳ trong 1 giây). Trong khi đó, $t \mapsto (\omega_{12}^3)^t = i^t$ mô tả cho biến đổi có tần số nhanh gấp 3 lần ($\frac{1}{4}$ chu kỳ trong 1 giây). \square



Hình 1.3: Các căn bậc 12 của đơn vị.

1.2 Vector phức

Tính toán “đồng thời” trên nhiều số phức được mô tả cô đọng bằng các phép toán trên bộ nhiều số phức, còn được gọi là các vector. Trên không gian vector này, các khái niệm trực quan như chuẩn (độ dài), trực giao (vuông góc), ... cũng được định nghĩa, tạo nên “sân khấu” của tính toán lượng tử. Đặc biệt, kí pháp Dirac, một hệ thống kí hiệu được thiết kế riêng cho tính toán lượng tử cũng được giới thiệu.

1.2.1 Vector

Một bộ gồm $n \in \mathbb{N}_{>0}$ số phức $v = (v_1, v_2, \dots, v_n)$ được gọi là một **vector (phức)** (complex vector) **cỡ** (size) n . Tập tất cả các vector cỡ n được kí hiệu là \mathbb{C}^n , tức là

$$\mathbb{C}^n = \{(v_1, v_2, \dots, v_n) : v_1, v_2, \dots, v_n \in \mathbb{C}\}.$$

Đặc biệt, vector $0 = 0_n = (0, 0, \dots, 0) \in \mathbb{C}^n$ gồm toàn các số 0 được gọi là **vector không** (zero vector).

Vector $v = (v_1, v_2, \dots, v_n)$ có thể được mô tả như là một cột hoặc một dòng số

$$v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}, \quad v = \begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix},$$

được gọi tương ứng là **vector cột** (column vector) và **vector dòng** (row vector).

Cho $v = (v_1, v_2, \dots, v_n) \in \mathbb{C}^n$

- vector $\bar{v} = (\bar{v}_1, \bar{v}_2, \dots, \bar{v}_n)$ được gọi là **liên hợp** (conjugate) của v ,
- v^T là vector dòng nếu v là vector cột hoặc v^T là vector cột nếu v là vector dòng, được gọi là **chuyển vị** (transpose) của v ,
- $v^\dagger = \bar{v}^T$ được gọi là **chuyển vị liên hợp** (dagger, conjugate transpose) của v .

Cho $u = (u_1, u_2, \dots, u_n)$, $v = (v_1, v_2, \dots, v_n) \in \mathbb{C}^n$ và $\alpha \in \mathbb{C}$

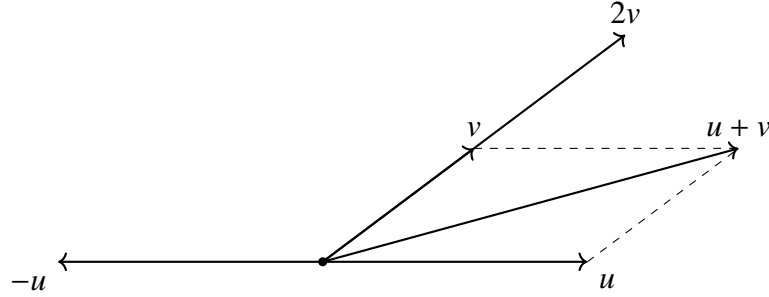
- ta nói $u = v$ nếu $u_1 = v_1, u_2 = v_2, \dots, u_n = v_n$,
- **đối** (opposite) của v là vector $-v = (-v_1, -v_2, \dots, -v_n) \in \mathbb{C}^n$,
- **tổng** (sum) của u với v là vector

$$u + v = (u_1 + v_1, u_2 + v_2, \dots, u_n + v_n) \in \mathbb{C}^n,$$

- **bội vô hướng** (scalar multiple) của α với v là vector

$$\alpha v = (\alpha v_1, \alpha v_2, \dots, \alpha v_n) \in \mathbb{C}^n.$$

Nếu vẽ minh họa các vector bằng các “đoạn thẳng có hướng” (mũi tên) thì Hình 1.4 minh họa vài vector và phép toán.



Hình 1.4: Vector và các phép toán.

Khi u là bội của v , tức là $u = \alpha v$, ta thường nói u, v cùng phương. Hơn nữa, nếu $\alpha > 0$, ta nói u, v cùng hướng; nếu $\alpha < 0$, ta nói u, v ngược hướng. Ví dụ, $2v$ cùng hướng với v còn $-2v$ ngược hướng với v ; cả hai đều cùng phương với v .

Cho $u, v, w \in \mathbb{C}^n$ và $\alpha, \beta \in \mathbb{C}$, các phép toán trên \mathbb{C}^n thỏa (♣)

- $u + v = v + u$,
- $(u + v) + w = u + (v + w)$,
- $v + 0 = v$,

- $v + (-v) = 0$,
- $\alpha(u + v) = \alpha u + \alpha v$,
- $(\alpha + \beta)v = \alpha v + \beta v$,
- $\alpha(\beta v) = (\alpha\beta)v$,
- $1v = v$.

Trong Toán, ta nói, \mathbb{C}^n cùng với các phép toán trên tạo thành một **không gian vector phức** (complex vector sapce) **n chiều** (dimension).

1.2.2 Chuẩn và tích vô hướng

Cho $v = (v_1, v_2, \dots, v_n) \in \mathbb{C}^n$, **chuẩn** (norm, length, magnitude) của v là số thực

$$\|v\| = \sqrt{|v_1|^2 + |v_2|^2 + \dots + |v_n|^2}.$$

Nhận xét, cho $v \in \mathbb{C}^n$ và $\alpha \in \mathbb{C}$ (♣)

- $\|v\| \geq 0$, $\|v\| = 0$ khi và chỉ khi $v = 0$,
- $\|\alpha v\| = |\alpha| \|v\|$.

Nếu $\|v\| = 1$ thì v được gọi là **vector đơn vị** (unit vector).

Nhận xét, cho $v \in \mathbb{C}^n$, $v \neq 0$

$$u = \frac{1}{\|v\|} v$$

là vector bội của v với $\|u\| = \frac{\|v\|}{\|v\|} = 1$ nên là vector đơn vị. Vector u còn được gọi là **chuẩn hóa** (normalization) của v .

Cho $u = (u_1, u_2, \dots, u_n)$, $v = (v_1, v_2, \dots, v_n) \in \mathbb{C}^n$, **tích vô hướng** (inner product) của u với v là số phức

$$\langle u, v \rangle = \overline{u_1} v_1 + \overline{u_2} v_2 + \dots + \overline{u_n} v_n. \quad (1.1)$$

Cho $u, v, w \in \mathbb{C}^n$ và $\alpha, \beta \in \mathbb{C}$, tích vô hướng trên \mathbb{C}^n thỏa (♣)

- $\langle v, v \rangle = \|v\|^2$,
- $\langle u, v \rangle = \overline{\langle v, u \rangle}$,
- $\langle u, \alpha v + \beta w \rangle = \alpha \langle u, v \rangle + \beta \langle u, w \rangle$,
- $\langle \alpha v + \beta w, u \rangle = \bar{\alpha} \langle v, u \rangle + \bar{\beta} \langle w, u \rangle$.

- **Bất đẳng thức Cauchy-Schwarz** (Cauchy-Schwarz inequality)

$$|\langle u, v \rangle|^2 \leq \langle u, u \rangle \langle v, v \rangle. \quad (1.2)$$

Trong Toán, ta nói, không gian vector \mathbb{C}^n cùng với tích vô hướng trên tạo thành một **không gian Hilbert hữu hạn chiều** (finite-dimensional Hilbert space).

1.2.3 Trục giao và chiếu trục giao

Từ bất đẳng thức Cauchy-Schwarz (1.2), **góc** (angle) giữa 2 vector $u, v \in \mathbb{C}^n, u, v \neq 0$ được định nghĩa là số thực $\theta \in [0, \pi]$ thỏa

$$\cos \theta = \frac{|\langle u, v \rangle|}{\|u\| \|v\|}.$$

Cho $u, v \in \mathbb{C}^n, u \neq 0$

- **chiếu (trục giao)** (orthogonal projection) của v lên u là vector

$$\text{proj}_u v = \frac{\langle u, v \rangle}{\|u\|^2} u.$$

Hơn nữa, nếu $\|u\| = 1$ (u là vector đơn vị) thì

$$\text{proj}_u v = \langle u, v \rangle u. \quad (1.3)$$

- ta nói u, v **trục giao** (orthogonal, perpendicular), kí hiệu $u \perp v$, nếu

$$\text{proj}_u v = 0 \Leftrightarrow \langle u, v \rangle = 0.$$

Nhận xét, (♣)

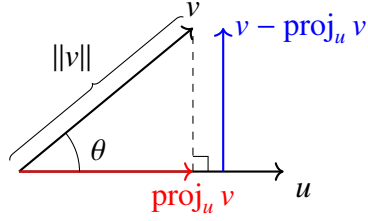
$$(v - \text{proj}_u v) \perp u. \quad (1.4)$$

Hình 1.5 minh họa chuẩn, góc, trục giao và chiếu trục giao.

1.2.4 Kí pháp Dirac

Cho $v = (v_1, \dots, v_n) \in \mathbb{C}^n$, theo **kí pháp Dirac** (Dirac notation)⁴

⁴đặt theo tên Paul Dirac (1902 - 1984), nhà Toán học và Vật lý học người Anh, một trong những cha đẻ của cơ học lượng tử.



Hình 1.5: Chuẩn, góc, trực giao và chiếu trực giao.

- vector cột của v được kí hiệu là $|v\rangle$ và được gọi là **ket**

$$|v\rangle = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix},$$

- vector dòng $|v\rangle^\dagger$ được kí hiệu là $\langle v|$ và được gọi là **bra**

$$\langle v| = [\overline{v_1} \quad \dots \quad \overline{v_n}].$$

Hơn nữa, cho $u = (u_1, \dots, u_n)$, $v = (v_1, \dots, v_n) \in \mathbb{C}^n$, tích vô hướng (1.1) có thể được viết bằng phép nhân ma trận (như sẽ thấy trong Phần 1.3.1) là

$$\langle u, v \rangle = \sum_{i=1}^n \overline{u_i} v_i = [\overline{u_1} \quad \overline{u_2} \quad \dots \quad \overline{u_n}] \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = |u\rangle^\dagger |v\rangle = \langle u || v \rangle$$

nên tích vô hướng của u với v còn được kí hiệu là $\langle u | v \rangle$ và được gọi là **bra-ket** hay **bracket**.

Ví dụ 1.2.1. $\mathbb{C}^2 = \{(\alpha, \beta) : \alpha, \beta \in \mathbb{C}\}$, tập tất cả các bộ 2 số phức, là không gian vector phức 2 chiều.

Trong \mathbb{C}^2 , các ket sau hay được dùng trong tính toán lượng tử

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, |+\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}, |-\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}, |i\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} \end{bmatrix}, |-i\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{i}{\sqrt{2}} \end{bmatrix}. \quad (1.5)$$

Lưu ý, các kí hiệu bên trong dấu ket $|\cdot\rangle$ chỉ là nhãn (tên gọi). Ta có thể chọn nhãn tùy ý (như u, v) nhưng các nhãn ở trên $(0, 1, +, -, i, -i)$ đã được chọn thông dụng trong tính toán lượng tử. Cũng lưu ý, ket $|0\rangle$ mô tả vector $(1, 0)$ còn vector không $(0, 0)$ được kí hiệu là 0 (rõ hơn là 0_2).

Các ket trong (1.5) đều là các vector đơn vị, chẳng hạn

$$\| |i\rangle \| = \sqrt{\left| \frac{1}{\sqrt{2}} \right|^2 + \left| \frac{i}{\sqrt{2}} \right|^2} = \sqrt{\frac{1}{2} + \frac{1}{2}} = 1.$$

Các cặp ket $\{|0\rangle, |1\rangle\}$; $|+\rangle, |-\rangle\}$; $|i\rangle, |-i\rangle\}$ trực giao nhau, chẳng hạn

$$\langle i|-i\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-i}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{-i}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}} + \frac{-i}{\sqrt{2}} \frac{-i}{\sqrt{2}} = \frac{1}{2} - \frac{1}{2} = 0.$$

Chiều của $|i\rangle$ lên $|+\rangle$ là vector

$$\text{proj}_{|+\rangle} |i\rangle = \langle +|i\rangle |+\rangle = \left(\frac{1}{2} + \frac{i}{2} \right) \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1+i}{2\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Góc giữa $|i\rangle$ và $|+\rangle$ là

$$\arccos \left(\frac{|\langle i|+\rangle|}{\| |i\rangle \| \| |+\rangle \|} \right) = \arccos \frac{1}{\sqrt{2}} = \frac{\pi}{4} (45^\circ).$$

□

1.2.5 Tổ hợp tuyến tính và cơ sở

Cho $v_1, v_2, \dots, v_k \in \mathbb{C}^n$ và $\alpha_1, \alpha_2, \dots, \alpha_k \in \mathbb{C}$, vector

$$v = \alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_k v_k$$

được gọi là một **tổ hợp tuyến tính** (linear combination) của v_1, v_2, \dots, v_k với các **hệ số** (coefficient) $\alpha_1, \alpha_2, \dots, \alpha_k$.

Tập các vector $S = \{v_1, v_2, \dots, v_k\}$ được gọi là **độc lập tuyến tính** (linearly independent) nếu không có vector nào trong S là một tổ hợp tuyến tính của các vector còn lại.

Mệnh đề 1.2.1. $S = \{v_1, v_2, \dots, v_k\}$ độc lập tuyến tính nếu

$$\alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_k v_k = 0$$

chỉ thỏa khi $\alpha_1 = 0, \alpha_2 = 0, \dots, \alpha_k = 0$. △

Tập các vector $S = \{v_1, v_2, \dots, v_k\}$ được gọi là **trực giao** (orthogonal) nếu

$$\langle v_i, v_j \rangle = 0, \forall i \neq j = 1, 2, \dots, k.$$

Hơn nữa, nếu

$$\|v_i\| = 1, \forall i = 1, 2, \dots, k$$

thì S được gọi là **trực chuẩn** (orthonormal).

Mệnh đề 1.2.2. S trực giao thì S độc lập tuyến tính. \triangle

Một tập $B = \{v_1, v_2, \dots, v_n\}$ gồm n vector độc lập tuyến tính trong \mathbb{C}^n được gọi là một **cơ sở** (basis) của \mathbb{C}^n . Hơn nữa, nếu B trực chuẩn thì B được gọi là một **cơ sở trực chuẩn** (orthonormal basis).

Trong \mathbb{C}^n , đặt e_i ($i = 1, \dots, n$) là vector gồm toàn các số 0 ngoại trừ phần tử thứ i là 1, tức là $e_i = (v_1, \dots, v_n)$ với

$$v_j = \delta_{ji} = \begin{cases} 1 & j = i \\ 0 & j \neq i \end{cases} \quad (j = 1, \dots, n)$$

thì

$$\langle e_i, e_j \rangle = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad (i, j = 1, \dots, n)$$

nên $\{e_1, \dots, e_n\}$ là một cơ sở trực chuẩn của \mathbb{C}^n gọi là **cơ sở chuẩn tắc** (standard basis, canonical basis).

Mệnh đề 1.2.3. Nếu $B = \{v_1, v_2, \dots, v_n\}$ là một cơ sở của \mathbb{C}^n thì với mọi $v \in \mathbb{C}^n$ tồn tại duy nhất một bộ n số $\alpha_1, \alpha_2, \dots, \alpha_n$ sao cho

$$v = \alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_n v_n.$$

Bộ $(\alpha_1, \alpha_2, \dots, \alpha_n)$ được gọi là tọa độ của v theo B , kí hiệu $[v]_B$. \triangle

Mệnh đề 1.2.4. Nếu $B = \{v_1, v_2, \dots, v_n\}$ là một cơ sở trực chuẩn của \mathbb{C}^n thì với mọi $v \in \mathbb{C}^n$

$$v = \langle v_1, v \rangle v_1 + \langle v_2, v \rangle v_2 + \dots + \langle v_n, v \rangle v_n,$$

tức là

$$[v]_B = (\langle v_1, v \rangle, \langle v_2, v \rangle, \dots, \langle v_n, v \rangle).$$

\triangle

Ví dụ 1.2.2. (tiếp Ví dụ 1.2.1) Các tập

$$B_Z = \{|0\rangle, |1\rangle\}, \quad B_X = \{|+\rangle, |-\rangle\}, \quad B_Y = \{|i\rangle, |-i\rangle\}$$

là các cơ sở trực chuẩn của \mathbb{C}^2 . B_Z là cơ sở chuẩn tắc.

Ket $|\phi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \in \mathbb{C}^2$ có thể được viết là

$$|\phi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \alpha|0\rangle + \beta|1\rangle \Rightarrow [|\phi\rangle]_{B_Z} = (\alpha, \beta).$$

Để tìm tọa độ của $|\phi\rangle$ trong B_X , ta có

$$\begin{cases} |+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \\ |-\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \end{cases} \Rightarrow \begin{cases} |0\rangle = \frac{1}{\sqrt{2}}(|+\rangle + |-\rangle) \\ |1\rangle = \frac{1}{\sqrt{2}}(|+\rangle - |-\rangle) \end{cases}.$$

Do đó

$$\begin{aligned} |\phi\rangle &= \alpha|0\rangle + \beta|1\rangle = \frac{\alpha}{\sqrt{2}}(|+\rangle + |-\rangle) + \frac{\beta}{\sqrt{2}}(|+\rangle - |-\rangle) \\ &= \frac{\alpha + \beta}{\sqrt{2}}|+\rangle + \frac{\alpha - \beta}{\sqrt{2}}|-\rangle \\ \Rightarrow [|\phi\rangle]_{B_X} &= \left(\frac{\alpha + \beta}{\sqrt{2}}, \frac{\alpha - \beta}{\sqrt{2}} \right). \end{aligned}$$

Nhanh hơn, vì B_X là cơ sở trực chuẩn nên ta có thể dùng công thức

$$[|\phi\rangle]_{B_X} = (\langle +|\phi\rangle, \langle -|\phi\rangle).$$

Chẳng hạn,

$$[|i\rangle]_{B_X} = (\langle +, i\rangle, \langle -, i\rangle) = \left(\frac{1+i}{2}, \frac{1-i}{2} \right).$$

□

Ví dụ 1.2.3. Trong \mathbb{C}^4 , tập các ket

$$\left\{ |00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, |01\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, |10\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, |11\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

là cơ sở chuẩn tắc của \mathbb{C}^4 . Tập các ket

$$\left\{ |\Phi^+\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}, |\Phi^-\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ -\frac{1}{\sqrt{2}} \end{bmatrix}, |\Psi^+\rangle = \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}, |\Psi^-\rangle = \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} \right\}$$

là một cơ sở trực chuẩn của \mathbb{C}^4 , được gọi là **cơ sở Bell** (Bell basis). Nhớ là, kí hiệu bên trong ket $|\cdot\rangle$ chỉ là nhãn (nhưng những nhãn này được dùng thông dụng trong tính toán lượng tử).

Ket $|\Phi^+\rangle$ được viết theo cơ sở chuẩn tắc

$$|\Phi^+\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle.$$

Ket $|00\rangle$ được viết theo cơ sở Bell

$$\begin{aligned} |00\rangle &= \langle\Phi^+|00\rangle|\Phi^+\rangle + \langle\Phi^-|00\rangle|\Phi^-\rangle + \langle\Psi^+|00\rangle|\Psi^+\rangle + \langle\Psi^-|00\rangle|\Psi^-\rangle \\ &= \frac{1}{\sqrt{2}}|\Phi^+\rangle + \frac{1}{\sqrt{2}}|\Phi^-\rangle. \end{aligned}$$

□

1.3 Ma trận phức

Khi cần tính toán trên nhiều vector, ta bố trí các vector thành một bảng số gồm nhiều dòng và nhiều cột, gọi là ma trận. Các ma trận cũng là phương tiện mô tả các toán tử tuyến tính. Đặc biệt, ma trận unita và ma trận Hermite là mô hình của các phép toán và phép đo lượng tử.

1.3.1 Ma trận

Một bảng gồm m dòng, n cột các số phức

$$A = (a_{ij}) = (a_{ij})_{m \times n} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

được gọi là một **ma trận (phức)** (complex matrix) **cỡ** (size) $m \times n$. Tập tất cả các ma trận cỡ $m \times n$ được kí hiệu là $\mathbb{C}^{m \times n}$. Nếu $m = n$, $A \in \mathbb{C}^{n \times n}$ được gọi là **ma trận vuông** (square matrix) **cấp** (order) n .

Đặc biệt, ma trận $0 = 0_{m \times n} = (0_{ij})_{m \times n}$ gồm toàn các số 0 được gọi là **ma trận không** (zero matrix). Vector $v \in \mathbb{C}^n$ khi viết như là vector cột hay vector dòng được xem tương ứng là ma trận gồm 1 cột (cỡ $n \times 1$) hay 1 dòng (cỡ $1 \times n$).

Cho $A = (a_{ij})_{m \times n}$

- $\bar{A} = (\overline{a_{ij}})_{m \times n}$ là **liên hợp** (conjugate) của A ,

- $A^T = (a_{ji})_{n \times m}$ là **chuyển vị** (transpose) của A ,
- $A^\dagger = (\bar{A})^T$ là **chuyển vị liên hợp** (dagger, conjugate transpose, adjoint) của A .

Cho $A = (a_{ij})_{m \times n}$, $B = (b_{ij})_{m \times n}$ và $\alpha \in \mathbb{C}$

- ta nói $A = B$ nếu $a_{ij} = b_{ij}$, $\forall i = 1, \dots, m, \forall j = 1, \dots, n$,
- **đối** (opposite) của A là ma trận $-A = (-a_{ij})_{m \times n}$,
- **tổng** (sum) của A với B là ma trận $A + B = (a_{ij} + b_{ij})_{m \times n}$,
- **bội vô hướng** (scalar multiple) của α với A là ma trận $\alpha A = (\alpha a_{ij})_{m \times n}$.

Các phép toán này có các tính chất tương tự như các phép toán tương ứng trên \mathbb{C}^n , chẳng hạn

$$A + 0_{m \times n} = A, \forall A \in \mathbb{C}^{m \times n}.$$

Cho $A = (a_{ij})_{m \times p}$, $B = (b_{ij})_{p \times n}$, **tích** (product) của A với B là ma trận $C = AB = (c_{ij})_{m \times n}$ với

$$c_{ij} = \sum_{k=1}^p a_{ik} b_{kj}.$$

Ma trận vuông $I = I_n = (\delta_{ij})_{n \times n}$ với

$$\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad (i, j = 1, \dots, n)$$

được gọi là **ma trận đơn vị** (identity matrix) cấp n .

Cho $A \in \mathbb{C}^{n \times n}$ và $n \in \mathbb{N}$, **lũy thừa** bậc n của A là

$$A^n = \underbrace{AA \dots A}_{n \text{ thừa số}} \quad (A^0 = I, A^1 = A).$$

Cho A, B, C là các ma trận có cỡ phù hợp và $\alpha, \beta \in \mathbb{C}$, phép nhân ma trận có các tính chất (♣)

- $AI = IA = A$,
- $(AB)C = A(BC)$,
- $A(B + C) = AB + AC$,
- $(B + C)A = BA + CA$,

- $\alpha(\beta C) = (\alpha\beta)C$,
- $\alpha(BC) = (\alpha B)C = B(\alpha C)$,
- $\overline{AB} = \bar{A}\bar{B}$,
- $(AB)^T = B^T A^T$,
- $(AB)^\dagger = B^\dagger A^\dagger$.

Lưu ý, phép nhân ma trận không có tính giao hoán, nghĩa là có các ma trận A, B thỏa $AB \neq BA$ (\clubsuit).

Trong kí pháp Dirac, cho $v \in \mathbb{C}^n$

- ket $|v\rangle$ là vector cột, được xem như là ma trận gồm 1 cột, $|v\rangle \in \mathbb{C}^{n \times 1}$,
- bra $\langle v| = |v\rangle^\dagger \in \mathbb{C}^{1 \times n}$ được xem như là ma trận gồm 1 dòng.

Do đó, cho $\alpha \in \mathbb{C}$; $u, v \in \mathbb{C}^n$; $A \in \mathbb{C}^{n \times n}$, các phép toán sau đều là các phép nhân ma trận

- nhân vô hướng: $|w\rangle = \alpha|v\rangle$ được $|w\rangle \in \mathbb{C}^{n \times 1}$, xem như $\alpha \in \mathbb{C}^{1 \times 1}$.
- tích vô hướng: $\beta = \langle u|v\rangle = \langle u|v\rangle$ được $\beta \in \mathbb{C}^{1 \times 1}$ xem như $\beta \in \mathbb{C}$.
- nhân ma trận với ket: $|w\rangle = A|v\rangle$ được $|w\rangle \in \mathbb{C}^{n \times 1}$, nếu $A = \begin{bmatrix} |a_1\rangle & |a_2\rangle & \dots & |a_n\rangle \end{bmatrix}$ và $v = (v_1, v_2, \dots, v_n)$ thì

$$|w\rangle = A|v\rangle = v_1|a_1\rangle + v_2|a_2\rangle + \dots + v_n|a_n\rangle.$$

- **tích ngoài** (outer product): $P = |u\rangle\langle v|$ được $P \in \mathbb{C}^{n \times n}$, mà với $|w\rangle \in \mathbb{C}^{n \times 1}$

$$P|w\rangle = |u\rangle\langle v||w\rangle = |u\rangle\langle v|w\rangle = \langle v|w\rangle|u\rangle.$$

Cho vector đơn vị u ($\|u\| = 1$), xét các ma trận

$$P_u = |u\rangle\langle u|, \quad (1.6)$$

$$R_u = 2|u\rangle\langle u| - I. \quad (1.7)$$

Vì một vector v bất kỳ có thể được viết là

$$v = \underbrace{\text{proj}_u v}_{v_u} + \underbrace{(v - \text{proj}_u v)}_{v_{\perp u}} = v_u + v_{\perp u}$$

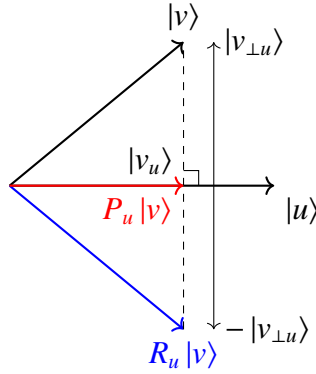
nên từ (1.3) và (1.4) ta có v_u cùng phương với u còn $v_{\perp u}$ vuông góc với u . Bây giờ

$$P_u |v\rangle = |u\rangle\langle u|(|v_u\rangle + |v_{\perp u}\rangle) = |u\rangle\langle u|v_u\rangle + |u\rangle\langle u|v_{\perp u}\rangle = v_u,$$

và

$$R_u |v\rangle = (2|u\rangle\langle u| - I)(|v_u\rangle + |v_{\perp u}\rangle) = 2|v_u\rangle - |v_u\rangle - |v_{\perp u}\rangle = |v_u\rangle - |v_{\perp u}\rangle.$$

Do đó, P_u được gọi là **ma trận chiếu** (projection matrix) lên trục u còn R_u được gọi là **ma trận phản xạ** (reflection matrix) qua trục u . Hình 1.6 minh họa các phép biến đổi này.



Hình 1.6: Phép chiếu và phép phản xạ.

Nhận xét (♣)

- $P_u |u\rangle = |u\rangle, R_u |u\rangle = |u\rangle,$
- $P_u |w\rangle = 0, R_u |w\rangle = -|w\rangle$ với mọi $w \perp u$.

Mệnh đề 1.3.1. Cho $A = (a_{ij}) \in \mathbb{C}^{n \times n}$ và $\{e_1, \dots, e_n\}$ là cơ sở chuẩn tắc của \mathbb{C}^n thì

$$A = \sum_{i=1}^n \sum_{j=1}^n a_{ij} |e_i\rangle \langle e_j|.$$

△

Đặc biệt, nếu A là **ma trận đường chéo** (diagonal matrix), là ma trận có các phần tử bên ngoài đường chéo chính là 0, thì

$$A = \sum_{i=1}^n a_{ii} |e_i\rangle \langle e_i|.$$

Chẳng hạn,

$$I = \sum_{i=1}^n |e_i\rangle \langle e_i|.$$

1.3.2 Toán tử tuyến tính

Một ánh xạ $T : \mathbb{C}^n \rightarrow \mathbb{C}^n$ được gọi là một **toán tử (tuyến tính)** (linear operator) trên \mathbb{C}^n nếu với mọi $u, v \in \mathbb{C}^n$ và $\alpha, \beta \in \mathbb{C}$

$$T(\alpha u + \beta v) = \alpha T(u) + \beta T(v).$$

Khi đó, giá trị của T tại v , $T(v)$, thường được viết gọn là Tv .

Kí hiệu 2 toán tử đặc biệt sau trên \mathbb{C}^n

- $I : v \mapsto Iv = v$ là **toán tử đơn vị** (identity operator).
- $0 : v \mapsto 0v = 0$ là **toán tử không** (zero operator).

Cho $S, T : \mathbb{C}^n \rightarrow \mathbb{C}^n$, **hợp** (composition) của S với T được định nghĩa là ánh xạ $S \circ T : \mathbb{C}^n \rightarrow \mathbb{C}^n$ thỏa

$$(S \circ T)(v) = S(T(v)).$$

Mệnh đề 1.3.2. Nếu S, T là các toán tử trên \mathbb{C}^n thì $S \circ T$ cũng vậy. Khi đó, $S \circ T$ được viết gọn là ST . Δ

Nhận xét, cho $A \in \mathbb{C}^{n \times n}$, do tính chất của các phép toán trên ma trận mà

$$T_A : v \mapsto T_A(v) = Av$$

là một toán tử trên \mathbb{C}^n . (♣)

Ngược lại, cho T là một toán tử trên \mathbb{C}^n và $B = \{v_1, v_2, \dots, v_n\}$ là một cơ sở của \mathbb{C}^n , gọi

$$a_i = [Tv_i]_B \quad (i = 1, \dots, n)$$

là tọa độ của Tv_i theo cơ sở B , ma trận

$$A = \begin{bmatrix} |a_1\rangle & |a_2\rangle & \dots & |a_n\rangle \end{bmatrix}$$

được gọi là **ma trận biểu diễn** của T theo B , kí hiệu $A = [T]_B$. Chẳng hạn, các phép chiếu và phản xạ qua trục $u \in \mathbb{C}^n$ là các toán tử trên \mathbb{C}^n với ma trận biểu diễn theo cơ sở chuẩn tắc tương ứng là P_u, R_u như trong (1.6), (1.7). (♣)

Đặc biệt, ma trận biểu diễn của toán tử đơn vị, toán tử không lần lượt là ma trận đơn vị, ma trận không ($[I]_B = I, [0]_B = 0$).

Mệnh đề 1.3.3. Cho S, T là các toán tử trên \mathbb{C}^n và B là một cơ sở của \mathbb{C}^n

- $[Tv]_B = [T]_B[v]_B, \forall v \in \mathbb{C}^n$.
- $[ST]_B = [S]_B[T]_B$.

△

Ví dụ 1.3.1. (tiếp Ví dụ 1.2.2) Các ma trận sau hay được dùng trong \mathbb{C}^2 , được gọi là các **ma trận Pauli** (Pauli matrix)

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

Các ma trận này tương ứng với các toán tử tuyến tính hay dùng trên \mathbb{C}^2 . Chẳng hạn, với cơ sở chuẩn tắc $B_Z = \{|0\rangle, |1\rangle\}$,

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

biểu diễn cho toán tử “lật bit (NOT)” vì $X|0\rangle = |1\rangle, X|1\rangle = |0\rangle$. Tương tự, Z là toán tử “lật pha” vì $Z|0\rangle = |0\rangle, Z|1\rangle = -|1\rangle$.

Toán tử X “biến đổi” $|\phi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \in \mathbb{C}^2$ thành

$$X|\phi\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix} \in \mathbb{C}^2.$$

Dùng tính chất tuyến tính ta cũng có

$$X|\phi\rangle = X(\alpha|0\rangle + \beta|1\rangle) = \alpha X|0\rangle + \beta X|1\rangle = \beta|0\rangle + \alpha|1\rangle.$$

Tương tự

$$Z|\phi\rangle = Z(\alpha|0\rangle + \beta|1\rangle) = \alpha Z|0\rangle + \beta Z|1\rangle = \alpha|0\rangle - \beta|1\rangle.$$

Dùng tích ngoài ta cũng có thể viết

$$\begin{aligned} X &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} = |0\rangle\langle 1| + |1\rangle\langle 0|. \end{aligned}$$

“Tác động” của X lên $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ có thể được viết là

$$\begin{aligned} X|\phi\rangle &= (|0\rangle\langle 1| + |1\rangle\langle 0|)(\alpha|0\rangle + \beta|1\rangle) \\ &= \alpha|0\rangle \underbrace{\langle 1|0\rangle}_0 + \beta|0\rangle \underbrace{\langle 1|1\rangle}_1 + \alpha|1\rangle \underbrace{\langle 0|0\rangle}_1 + \beta|1\rangle \underbrace{\langle 0|1\rangle}_0 \\ &= \beta|0\rangle + \alpha|1\rangle. \end{aligned}$$

□

Ví dụ 1.3.2. (tiếp Ví dụ 1.3.1) Ma trận sau cũng hay được dùng trong \mathbb{C}^2 , được gọi là **ma trận Hadamard** (Hadamard matrix)

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

Ta thấy $H = \begin{bmatrix} |+\rangle & |-\rangle \end{bmatrix}$ nên $H|0\rangle = |+\rangle, H|1\rangle = |-\rangle$. H còn được gọi là **ma trận chuyển cơ sở** (change-of-basis matrix), “chuyển” $B_Z = \{|0\rangle, |1\rangle\}$ sang $B_X = \{|+\rangle, |-\rangle\}$.

H hay được dùng vì giúp “biến” $|0\rangle$ thành “tổ hợp đều” của $|0\rangle, |1\rangle$

$$H|0\rangle = |+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle.$$

Đẳng thức ma trận

$$HXH = Z \quad (*)$$

nói rằng

$$HXH|\phi\rangle = H(X(H|\phi)) = Z|\phi\rangle, \forall |\phi\rangle \in \mathbb{C}^2$$

tức là từ $|\phi\rangle$ biến đổi bằng H rồi biến đổi tiếp bằng X rồi biến đổi tiếp bằng H thì được cùng kết quả như từ $|\phi\rangle$ biến đổi bằng Z .

(*) có thể được chứng minh bằng cách nhân ma trận

$$\begin{aligned} HXH &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix} = Z. \end{aligned}$$

Nhận xét (♣)

$$Z|0\rangle = |0\rangle, Z|1\rangle = -|1\rangle, \quad X|+\rangle = |+\rangle, X|-\rangle = -|-\rangle,$$

$$H|0\rangle = |+\rangle, H|1\rangle = |-\rangle, \quad H|+\rangle = |0\rangle, H|-\rangle = |1\rangle,$$

nên (*) cũng có thể được chứng minh bằng cách xét tác động của các toán tử trên các vector cơ sở

$$\begin{array}{ccccccc} |0\rangle & \xrightarrow{H} & |+\rangle & \xrightarrow[X]{Z} & |+\rangle & \xrightarrow{H} & |0\rangle \\ |1\rangle & \xrightarrow{H} & |-\rangle & \xrightarrow[X]{Z} & -|-\rangle & \xrightarrow{H} & -|1\rangle \end{array}$$

□

Ví dụ 1.3.3. (tiếp Ví dụ 1.2.3) Trong \mathbb{C}^4 , ma trận sau hay được dùng

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

Vì $\text{CNOT} = [|00\rangle \ |01\rangle \ |11\rangle \ |10\rangle]$ nên ta có CNOT biến các ket $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ lần lượt thành $|00\rangle, |01\rangle, |11\rangle, |10\rangle$. Nói cách khác, CNOT giữ nguyên $|00\rangle, |01\rangle$ nhưng hoán đổi $|10\rangle, |11\rangle$.

Cho $|\phi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \in \mathbb{C}^4$ thì

$$\begin{aligned} \text{CNOT}|\phi\rangle &= \text{CNOT}(\alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle) \\ &= \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{11}|10\rangle + \alpha_{10}|11\rangle. \end{aligned}$$

Chẳng hạn

$$\text{CNOT}|\Phi^+\rangle = \text{CNOT}\left(\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle\right) = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle.$$

Ta cũng có điều này bằng phép nhân ma trận

$$\begin{aligned} \text{CNOT}|\Phi^+\rangle &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} \\ &= \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle. \end{aligned}$$

□

1.3.3 Ma trận khả nghịch

Ma trận vuông $A \in \mathbb{C}^{n \times n}$ được gọi là **khả nghịch** (invertible) nếu có (duy nhất) $B \in \mathbb{C}^{n \times n}$ sao cho

$$AB = BA = I_n.$$

Khi đó, B được gọi là **nghịch đảo** của A , kí hiệu $B = A^{-1}$.

Ví dụ, ma trận chiếu P_u (1.6) không khả nghịch còn ma trận phản xạ R_u (1.7) khả nghịch với $R_u^{-1} = R_u$. (♣)

Nhận xét, cho $A, B \in \mathbb{C}^{n \times n}$ (♣)

- $(A^{-1})^{-1} = A$,
- $(A^\dagger)^{-1} = (A^{-1})^\dagger$,
- $(AB)^{-1} = B^{-1}A^{-1}$.

Nếu $A \in \mathbb{C}^{n \times n}$ thì A^{-1} biểu diễn cho **toán tử ngược** (inverse operator) của toán tử mà A biểu diễn vì

$$A^{-1}(A|v\rangle) = (A^{-1}A)|v\rangle = I|v\rangle = |v\rangle, \forall |v\rangle \in \mathbb{C}^n.$$

Nói cách khác, A^{-1} “undo” A .

1.3.4 Trị riêng và vector riêng *

Cho $A \in \mathbb{C}^{n \times n}$, nếu có $\lambda \in \mathbb{C}$ và $v \in \mathbb{C}^n, v \neq 0$ sao cho

$$Av = \lambda v$$

thì λ được gọi là một **trị riêng** (eigenvalue) của A và v là một **vector riêng** (eigenvector) tương ứng.

Ví dụ, các ma trận chiếu P_u (1.6) và ma trận phản xạ R_u (1.7) đều có một trị riêng là 1 với vector riêng tương ứng là u vì

$$P_u u = R_u u = u = 1u.$$

Ngoài ra, P_u, R_u còn có các trị riêng khác. (♣)

Ma trận $A \in \mathbb{C}^{n \times n}$ được gọi là **ma trận chuẩn tắc** (normal matrix) nếu A giao hoán với A^\dagger , tức là

$$AA^\dagger = A^\dagger A.$$

Mệnh đề 1.3.4. (Phân tích phổ, spectral decomposition) $A \in \mathbb{C}^{n \times n}$ chuẩn tắc khi và chỉ khi A có các vector riêng $\{v_1, \dots, v_n\}$ lập thành một cơ sở trực chuẩn của \mathbb{C}^n . Khi đó, đặt $U = \begin{bmatrix} |v_1\rangle & \dots & |v_n\rangle \end{bmatrix}$ và $D \in \mathbb{C}^{n \times n}$ là ma trận đường chéo gồm các trị riêng $\lambda_1, \dots, \lambda_n$ tương ứng trên đường chéo chính thì

$$A = UDU^\dagger = \sum_{i=1}^n \lambda_i |v_i\rangle \langle v_i|.$$

△

Ví dụ 1.3.4. (tiếp Ví dụ 1.3.2) Trong \mathbb{C}^2 , ta có

$$\begin{aligned} X|+\rangle &= X\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{\sqrt{2}}X|0\rangle + \frac{1}{\sqrt{2}}X|1\rangle \\ &= \frac{1}{\sqrt{2}}|1\rangle + \frac{1}{\sqrt{2}}|0\rangle = |+\rangle, \\ X|-\rangle &= X\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{\sqrt{2}}X|0\rangle - \frac{1}{\sqrt{2}}X|1\rangle \\ &= \frac{1}{\sqrt{2}}|1\rangle - \frac{1}{\sqrt{2}}|0\rangle = -|-\rangle. \end{aligned}$$

Do đó, $|+\rangle, |-\rangle$ là các vector riêng của X với các trị riêng tương ứng là $1, -1$.

Hơn nữa $\{|+\rangle, |-\rangle\}$ lập thành một cơ sở trực chuẩn của \mathbb{C}^2 nên nếu đặt

$$H = \begin{bmatrix} |+\rangle & |-\rangle \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}, \quad D = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

thì

$$X = HDH^\dagger = |+\rangle\langle +| - |-\rangle\langle -|.$$

Lưu ý, H chính là ma trận Hadamard.

Vì $B_X = \{|+\rangle, |-\rangle\}$ là một cơ sở trực chuẩn gồm các vector riêng của X nên B_X còn được gọi là **cơ sở X** (X-basis).

Tương tự, $B_Z = \{|0\rangle, |1\rangle\}$, $B_Y = \{|i\rangle, |-i\rangle\}$ lần lượt được gọi là **cơ sở Z** (Z-basis), **cơ sở Y** (Y-basis). \square

1.3.5 Ma trận unita và ma trận Hermite

Ma trận $U \in \mathbb{C}^{n \times n}$ được gọi là **ma trận unita** (unitary matrix) nếu U khả nghịch và $U^{-1} = U^\dagger$, tức là

$$U^\dagger U = UU^\dagger = I.$$

Ví dụ, ma trận phản xạ R_u (1.7) là ma trận unita. (\clubsuit)

Mệnh đề 1.3.5. Cho $U \in \mathbb{C}^{n \times n}$, các khẳng định sau là tương đương

1. U là ma trận unita,
2. Các cột của U tạo thành cơ sở trực chuẩn của \mathbb{C}^n ($U^\dagger U = I$),
3. U bảo toàn tích vô hướng, tức là $\langle Uv, Uw \rangle = \langle v, w \rangle, \forall v, w \in \mathbb{C}^n$,
4. U **bảo toàn chuẩn**, tức là $\|Uv\| = \|v\|, \forall v \in \mathbb{C}^n$,

5. U chuẩn tắc với các trị riêng có độ lớn là 1.

△

Mệnh đề 1.3.6. Cho $U, V \in \mathbb{C}^{n \times n}$ là các ma trận unita

- $U^{-1} = U^\dagger$ là ma trận unita,
- UV là ma trận unita,
- αU là ma trận unita nếu $\alpha \in \mathbb{C}, |\alpha| = 1$.

△

Ma trận $A \in \mathbb{C}^{n \times n}$ được gọi là **ma trận Hermite** (Hermitian matrix) nếu

$$A = A^\dagger.$$

Ví dụ, ma trận chiếu P_u (1.6) là ma trận Hermite. (♣)

Mệnh đề 1.3.7. Cho $A \in \mathbb{C}^{n \times n}$, các khẳng định sau là tương đương

1. $A = A^\dagger$,
2. $\langle u, Av \rangle = \langle Au, v \rangle, \forall u, v \in \mathbb{C}^n$,
3. $\langle v, Av \rangle \in \mathbb{R}, \forall v \in \mathbb{C}^n$,
4. A chuẩn tắc với các trị riêng là các số thực.

△

Mệnh đề 1.3.8. Cho $A, B \in \mathbb{C}^{n \times n}$ là các ma trận Hermite

- $A + B$ là ma trận Hermite,
- αA là ma trận Hermite nếu $\alpha \in \mathbb{R}$,
- AB là ma trận Hermite khi và chỉ khi $AB = BA$,
- ABA là ma trận Hermite.

△

Ví dụ 1.3.5. (tiếp Ví dụ 1.3.2) Các ma trận Pauli I, X, Y, Z và ma trận Hadamard H đều là các ma trận unita và Hermite, chẳng hạn

$$Y^\dagger = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = Y,$$

$$Y^\dagger Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I.$$

Do đó nghịch đảo của các ma trận này đều là chính nó, chẳng hạn

$$Y^{-1} = Y^\dagger = Y.$$

Từ đó, vì $H|0\rangle = |+\rangle$, $H|1\rangle = |-\rangle$ nên $H|+\rangle = |0\rangle$, $H|-\rangle = |1\rangle$ do $H^{-1} = H$. \square

1.4 Tích tensor

Tích tensor là công cụ của Toán cho phép “ghép” các không gian vector nhỏ thành các không gian vector lớn. Nếu như trong tính toán cổ điển, ta chỉ cần tích Descartes (hay phép nối chuỗi, string concatenation) thì trong tính toán lượng tử, ta cần tích tensor để ghép các hệ lượng tử nhỏ thành các hệ lượng tử lớn.

Cho $A = (a_{ij}) \in \mathbb{C}^{m \times n}$, $B = (b_{ij}) \in \mathbb{C}^{p \times q}$, **tích Kronecker** (Kronecker product) hay **tích tensor** (tensor product) của A với B là

$$\begin{aligned} A \otimes B &= \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix} \in \mathbb{C}^{mp \times nq} \\ &= \begin{bmatrix} a_{11}b_{11} & \dots & a_{11}b_{1q} & \dots & a_{1n}b_{11} & \dots & a_{1n}b_{1q} \\ \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & \dots & a_{11}b_{pq} & \dots & a_{1n}b_{p1} & \dots & a_{1n}b_{pq} \\ \vdots & & \vdots & & \vdots & & \vdots \\ a_{m1}b_{11} & \dots & a_{m1}b_{1q} & \dots & a_{mn}b_{11} & \dots & a_{mn}b_{1q} \\ \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ a_{m1}b_{p1} & \dots & a_{m1}b_{pq} & \dots & a_{mn}b_{p1} & \dots & a_{mn}b_{pq} \end{bmatrix}. \end{aligned}$$

Cho A, B, C là các ma trận phù hợp và $\alpha \in \mathbb{C}$, tích tensor thỏa (\clubsuit)

- $(A \otimes B) \otimes C = A \otimes (B \otimes C)$,
- $A \otimes (B + C) = A \otimes B + A \otimes C$,
- $(B + C) \otimes A = B \otimes A + C \otimes A$,
- $(\alpha A) \otimes B = A \otimes (\alpha B) = \alpha(A \otimes B)$,
- **tích hỗn hợp** (mixed-product)

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD),$$

- $(A \otimes B)^T = A^T \otimes B^T$,

- $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$,
- $(A \otimes B)^\dagger = A^\dagger \otimes B^\dagger$,

Lưu ý, tích tensor không có tính giao hoán.

Nếu $|\phi\rangle \in \mathbb{C}^n$, $|\psi\rangle \in \mathbb{C}^m$ thì $|\phi\rangle \otimes |\psi\rangle \in \mathbb{C}^{nm}$, kí pháp Dirac viết

$$|\phi\rangle \otimes |\psi\rangle = |\phi\rangle|\psi\rangle = |\phi, \psi\rangle = |\phi\psi\rangle,$$

$$\langle\phi| \otimes \langle\psi| = (|\phi\rangle)^\dagger \otimes (|\psi\rangle)^\dagger = (|\phi\rangle \otimes |\psi\rangle)^\dagger = (|\phi\psi\rangle)^\dagger = \langle\phi\psi|.$$

Khi đó, cho các ma trận A, B và các ket $|a\rangle, |b\rangle, |c\rangle, |d\rangle$ phù hợp, dùng tính tích hỗn hợp của tích tensor ta có

$$\begin{aligned} A|a\rangle B|b\rangle &= (A|a\rangle) \otimes (B|b\rangle) = (A \otimes B)(|a\rangle \otimes |b\rangle) \\ &= (A \otimes B)|ab\rangle, \\ \langle ab|cd\rangle &= \langle ab||cd\rangle = (\langle a| \otimes \langle b|)(|c\rangle \otimes |d\rangle) = (\langle a||c\rangle) \otimes (\langle b||d\rangle) \\ &= \langle a|c\rangle \otimes \langle b|d\rangle = \langle a|c\rangle \langle b|d\rangle. \quad (\langle a|c\rangle, \langle b|d\rangle \text{ là các số.}) \end{aligned}$$

Cho $n, m \in \mathbb{N}, n, m > 1$, $|\chi\rangle \in \mathbb{C}^{nm}$ được gọi là **tách được** (separable) nếu có $|\phi\rangle \in \mathbb{C}^n, |\psi\rangle \in \mathbb{C}^m$ sao cho

$$|\chi\rangle = |\phi\rangle|\psi\rangle.$$

Ngược lại, $|\chi\rangle$ được gọi là không tách được hay **rối** (entangled).

Ví dụ 1.4.1. (tiếp Ví dụ 1.2.1, 1.2.3) Trong \mathbb{C}^4 , ta có

$$|00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = |0\rangle \otimes |0\rangle = |0\rangle|0\rangle.$$

Tương tự, $|01\rangle = |0\rangle|1\rangle, |10\rangle = |1\rangle|0\rangle, |11\rangle = |1\rangle|1\rangle$.

Ta cũng có

$$\begin{aligned} X|0\rangle &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle, \\ Y|0\rangle &= \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ i \end{bmatrix} = i|1\rangle, \\ X \otimes Y &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \\ 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \end{bmatrix}, \end{aligned}$$

$$(X \otimes Y)|00\rangle = \begin{bmatrix} 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \\ 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ i \end{bmatrix} = i \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = i|11\rangle,$$

Dùng kí pháp Dirac ta có ngay

$$(X \otimes Y)|00\rangle = (X|0\rangle)(Y|0\rangle) = |1\rangle(i|1\rangle) = i|1\rangle|1\rangle = i|11\rangle.$$

Vì $|00\rangle = |0\rangle|0\rangle$ nên $|00\rangle$ tách được. Tuy nhiên, các vector cơ sở Bell là không tách được. Chẳng hạn, nếu $|\Phi^+\rangle$ tách được thì có

$$|\phi\rangle = \begin{bmatrix} a \\ b \end{bmatrix}, |\psi\rangle = \begin{bmatrix} c \\ d \end{bmatrix} \in \mathbb{C}^2$$

sao cho

$$|\Phi^+\rangle = |\phi\rangle|\psi\rangle \Leftrightarrow \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} \otimes \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} ac \\ ad \\ bc \\ bd \end{bmatrix} \Rightarrow \begin{cases} ad = 0 \\ bc = 0 \\ acbd = \frac{1}{2} \end{cases} \quad (\text{vô lý!})$$

□

Ví dụ 1.4.2. (tiếp Ví dụ 2.2.4) Trong \mathbb{C}^4 , ta có

$$\begin{aligned} \text{CNOT}|+\rangle|0\rangle &= \text{CNOT} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle = \text{CNOT} \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \\ &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = |\Phi^+\rangle. \end{aligned}$$

Vì các cột của CNOT tạo thành một cơ sở trực chuẩn của \mathbb{C}^4 (chính là cơ sở chuẩn tắc $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$) nên CNOT unita. Hơn nữa, $\text{CNOT}^\dagger = \text{CNOT}$ nên CNOT cũng Hermite, do đó

$$\text{CNOT}^{-1} = \text{CNOT}^\dagger = \text{CNOT}$$

nên ta cũng có $\text{CNOT}|\Phi^+\rangle = |+\rangle|0\rangle$.

Tương tự, ta có CNOT “chuyển qua lại” giữa các vector

$$\begin{aligned} | +0 \rangle &\longleftrightarrow |\Phi^+\rangle, \\ | -0 \rangle &\longleftrightarrow |\Phi^-\rangle, \\ | +1 \rangle &\longleftrightarrow |\Psi^+\rangle, \\ | -1 \rangle &\longleftrightarrow |\Psi^-\rangle. \end{aligned}$$

Ta cũng có

$$B_{XZ} = \{|+0\rangle, |-0\rangle, |+1\rangle, |-1\rangle\}$$

là một cơ sở trực chuẩn của \mathbb{C}^4 (♣) nên CNOT là ma trận chuyển cơ sở “giữa” cơ sở B_{XZ} và cơ sở Bell của \mathbb{C}^4 . □

Bài tập

1.1 Cho $x = 3 - i, y = 1 + 4i$.

- (a) Vẽ x, y trên mặt phẳng phức. (c) Tính $\bar{x}, -x, x^{-1}$.
 (b) Tính $\operatorname{Re}(x), \operatorname{Im}(x), |x|, \arg x$. (d) Tính $x + y, x - y, xy, \frac{x}{y}, \frac{y}{x}$.

1.2 Giải phương trình $x^4 + 2x^2 + 1 = 0$ trên \mathbb{C} .

1.3 Cho $n \in \mathbb{N}$, tính

- (a) i^n . (b) $(-i)^n$. (c) $\left(\frac{1+i}{\sqrt{2}}\right)^n$. (d) $(1+i)^n$.

1.4 Cho $x = e^{i\frac{\pi}{3}}, y = 2e^{i\frac{\pi}{6}}$.

- (a) Vẽ x, y trên mặt phẳng phức. (e) Tính $\bar{x}, -x, x^{-1}$.
 (b) Tìm dạng cực của x, y . (f) Tính $x + y, x - y, xy, \frac{x}{y}, \frac{y}{x}$.
 (c) Tìm dạng đại số của x, y . (g) Tính x^4 và $x^n, n \in \mathbb{Z}$.
 (d) Tính $\operatorname{Re}(x), \operatorname{Im}(x), |x|, \arg x$. (h) Tính $\sqrt[n]{x}$ và $\sqrt[n]{y}, n \in \mathbb{N}^+$.

1.5 Vẽ minh họa các căn bậc 5 của đơn vị trên mặt phẳng phức.

1.6 Chứng minh \mathbb{C} là một trường với các phép toán cộng, nhân, đối, nghịch đảo.

1.7 Cho $x, y \in \mathbb{C}$, chứng minh

- (a) $x = \bar{x} \Leftrightarrow x \in \mathbb{R}$. (e) $\overline{xy} = \bar{x}\bar{y}$. (i) $|xy| = |x||y|$.
 (b) $x\bar{x} = \bar{x}x = |x|^2$. (f) $\overline{x^{-1}} = (\bar{x})^{-1}$ ($x \neq 0$). (j) $\left|\frac{1}{x}\right| = \frac{1}{|x|}$ ($x \neq 0$).
 (c) $\bar{\bar{x}} = x$. (g) $\overline{\left(\frac{x}{y}\right)} = \frac{\bar{x}}{\bar{y}}$ ($y \neq 0$). (k) $\left|\frac{x}{y}\right| = \frac{|x|}{|y|}$ ($y \neq 0$).
 (d) $\overline{x+y} = \bar{x} + \bar{y}$. (h) $|\bar{x}| = |x|$. (l) $|x+y| \leq |x| + |y|$.

1.8 Cho $|\phi\rangle = \frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle, |\psi\rangle = \frac{2}{3}|0\rangle + \frac{1-2i}{3}|1\rangle$, tính

- (a) $\langle\phi|$ và $\langle\psi|$. (e) $\|\phi\|$ và $\|\psi\|$.
 (b) $\langle\phi|\psi\rangle$ và $\langle\psi|\phi\rangle$. (f) góc giữa $|\phi\rangle$ với $|\psi\rangle$.
 (c) $|\phi\rangle\langle\psi|$ và $|\psi\rangle\langle\phi|$. (g) $\operatorname{proj}_{|\psi\rangle} |\phi\rangle$.
 (d) $|\phi\rangle|\psi\rangle$ và $|\psi\rangle|\phi\rangle$. (h) $\operatorname{proj}_{|\phi\rangle} |\psi\rangle$.

1.9 Tiếp Bài tập 1.8

- (a) Chuẩn hóa $\text{proj}_{|\psi\rangle} |\phi\rangle$ và $\text{proj}_{|\phi\rangle} |\psi\rangle$.
 (b) Tìm tọa độ của $|\phi\rangle$ và $|\psi\rangle$ trong các cơ sở

$$B_Z = \{|0\rangle, |1\rangle\}, B_X = \{|+\rangle, |-\rangle\}, B_Y = \{|i\rangle, |-i\rangle\}.$$

- (c) Cho $|a\rangle = \frac{\sqrt{3}}{2}|0\rangle + \frac{i}{2}|1\rangle$, $|b\rangle = \frac{i}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle$, chứng minh $B = \{a, b\}$ là một cơ sở trực chuẩn của \mathbb{C}^2 và tìm tọa độ của $|\phi\rangle, |\psi\rangle$ theo B .

1.10 Cho $\alpha, \beta \in \mathbb{C}$, chứng minh

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle \Leftrightarrow \langle\phi| = \bar{\alpha}\langle 0| + \bar{\beta}\langle 1|.$$

1.11 Cho $u, v, w \in \mathbb{C}^n$ và $\alpha, \beta \in \mathbb{C}$, chứng minh

- (a) $\|\alpha v\| = |\alpha| \|v\|$.
 (b) $\langle v, v \rangle = \|v\|^2$.
 (c) $\langle u, v \rangle = \overline{\langle v, u \rangle}$.
 (d) $\langle u, \alpha v + \beta w \rangle = \alpha \langle u, v \rangle + \beta \langle u, w \rangle$.
 (e) $\langle \alpha v + \beta w, u \rangle = \bar{\alpha} \langle v, u \rangle + \bar{\beta} \langle w, u \rangle$.

1.12 Cho A, B là các ma trận, u, v là các vector và α là vô hướng phù hợp, chứng minh

- (a) $(A^\dagger)^\dagger = A$.
 (b) $(A + B)^\dagger = A^\dagger + B^\dagger$.
 (c) $(\alpha A)^\dagger = \bar{\alpha} A^\dagger$.
 (d) $(AB)^T = B^T A^T$.
 (e) $\overline{AB} = \bar{A} \bar{B}$.
 (f) $(AB)^\dagger = B^\dagger A^\dagger$.
 (g) A khả nghịch khi và chỉ khi A^\dagger khả nghịch, khi đó $(A^\dagger)^\dagger = (A^{-1})^\dagger$.
 (h) Trị riêng của A^\dagger là liên hợp phức của trị riêng của A .
 (i) $\langle Au, v \rangle = \langle u, A^\dagger v \rangle$.

1.13 Cho U là toán tử trên \mathbb{C}^2 với

$$U|0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix}, \quad U|1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} -i \\ 1 \end{bmatrix}.$$

- (a) Tìm biểu diễn của U trong cơ sở chuẩn tắc $B_Z = \{|0\rangle, |1\rangle\}$.
 (b) Cho $|\phi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \in \mathbb{C}^2$, tìm $U|\phi\rangle$.
 (c) U có unita không?

- (d) U có Hermite không? (f) Tìm $HUH|0\rangle, HUH|1\rangle$ và HUH (H là ma trận Hadamard).
- (e) Tìm U^\dagger, U^{-1} . (g) Tìm $UHU|0\rangle, UHU|1\rangle$ và UHU .

1.14 Trong \mathbb{C}^2 tìm ma trận chiếu và phản xạ qua các ket

- (a) $|0\rangle$. (c) $|+\rangle$. (e) $|i\rangle$.
 (b) $|1\rangle$. (d) $|-\rangle$. (f) $|-i\rangle$.

Cho biết tác động của các ma trận này lên $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$.

1.15 Chứng minh $XY = iZ$ bằng cách

- (a) Nhân ma trận.
 (b) Xét tác động của các toán tử trên $|0\rangle, |1\rangle$.

1.16 Tìm một cơ sở trực chuẩn gồm các vector riêng của ma trận Hadamard H .

1.17 Tính

- (a) $\langle 11|11\rangle$. (c) $\langle + -|01\rangle$. (e) $|+ -\rangle$.
 (b) $\langle 10|11\rangle$. (d) $\langle 1+0|1-0\rangle$. (f) $|1+0\rangle$.

1.18 Cho $|\phi\rangle = \frac{1}{2}|00\rangle + \frac{i}{\sqrt{2}}|10\rangle + \frac{\sqrt{3}+i}{4}|11\rangle$

- (a) Cho thấy $|\phi\rangle$ là vector đơn vị.
 (b) Tính $\text{proj}_{|+-\rangle} |\phi\rangle$ và chuẩn hóa $\text{proj}_{|+-\rangle} |\phi\rangle$.
 (c) Tính tọa độ của $|\phi\rangle$ theo cơ sở Bell.

1.19 Kiểm tra các vector sau có tách được

- (a) $|\phi_1\rangle = \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle)$. (c) $|\phi_3\rangle = \frac{1}{2\sqrt{2}}(\sqrt{3}|00\rangle - \sqrt{3}|01\rangle + |10\rangle - |11\rangle)$.
 (b) $|\phi_2\rangle = \frac{1}{\sqrt{2}}(|10\rangle + i|11\rangle)$. (d) $|\phi_4\rangle = \frac{1}{\sqrt{3}}|0+\rangle + \sqrt{\frac{2}{3}}|1-\rangle$.

1.20 Tính

- (a) $H \otimes X, X \otimes H, X \otimes Z, X \otimes Y$.
 (b) $(H \otimes X)|00\rangle, (H \otimes X)|01\rangle, (X \otimes Z)|10\rangle, (X \otimes Z)|11\rangle$.

$$(c) (H \otimes X)|\Phi^+\rangle, (H \otimes X)|\Phi^-\rangle, (X \otimes Z)|\Psi^+\rangle, (X \otimes Z)|\Psi^-\rangle.$$

$$1.21 \text{ Cho } |\phi\rangle = \frac{1}{4}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle + \frac{\sqrt{3}}{4}|11\rangle$$

$$(a) \text{ Tính } (H \otimes X)|\phi\rangle.$$

$$(b) \text{ Tính } \text{CNOT}|\phi\rangle.$$

1.22 Chứng minh

$$\text{CNOT}|++\rangle = |++\rangle,$$

$$\text{CNOT}|+-\rangle = |--\rangle,$$

$$\text{CNOT}|-+\rangle = |-+\rangle,$$

$$\text{CNOT}|--\rangle = |+-\rangle.$$

1.23 Chứng minh

$$(a) \text{CNOT}(X \otimes I) = (X \otimes X) \text{CNOT}. \quad (c) \text{CNOT}(Z \otimes I) = (Z \otimes I) \text{CNOT}.$$

$$(b) \text{CNOT}(I \otimes X) = (I \otimes X) \text{CNOT}. \quad (d) \text{CNOT}(I \otimes Z) = (Z \otimes Z) \text{CNOT}.$$

1.24 Chứng minh

$$(a) \text{ Nếu } |\phi\rangle \in \mathbb{C}^n, |\psi\rangle \in \mathbb{C}^m \text{ là các vector đơn vị thì } |\phi\rangle|\psi\rangle, |\psi\rangle|\phi\rangle \in \mathbb{C}^{mn} \text{ cũng là các vector đơn vị.}$$

$$(b) \text{ Nếu } A \in \mathbb{C}^{n \times n}, B \in \mathbb{C}^{m \times m} \text{ là các ma trận unita thì } A \otimes B, B \otimes A \in \mathbb{C}^{mn \times mn} \text{ cũng là các ma trận unita.}$$

Chương 2

Tính toán cổ điển

Về mặt công nghệ, **điện toán lượng tử** (quantum computing)¹ hiện nay đang giống với điện toán cổ điển những năm mới ra đời. Về mặt lý thuyết, tính toán lượng tử là phiên bản rộng và sâu hơn của tính toán cổ điển. Hơn nữa, tính toán lượng tử không thay thế cho tính toán cổ điển. Hai công nghệ sẽ hoạt động cùng nhau trong tương lai, mỗi công nghệ chịu trách nhiệm cho phần là thế mạnh của mình.

Chương này trình bày các vấn đề cơ bản của tính toán cổ điển. Đây cũng là các kiến thức nền tảng để hiểu tính toán lượng tử. Có thể nói, nhiều thành tố của tính toán cổ điển được mở rộng tự nhiên sang tính toán lượng tử. Đặc biệt, chương này cũng trình bày sự chuyển tiếp từ tính toán cổ điển sang tính toán lượng tử.

2.1 Bit và mã hóa thông tin

2.1.1 Bit

Bit là đơn vị nhỏ nhất của thông tin trong tính toán cổ điển. Mỗi bit biểu diễn một trạng thái với chỉ một trong hai lựa chọn, còn gọi là giá trị. Hai giá trị này thường được kí hiệu là 0, 1 là hai **kí số nhị phân** (binary digit, bit). Tập $\{0, 1\}$ thường được kí hiệu là \mathbb{B} .

Về mặt Toán học, $b \in \mathbb{B} = \{0, 1\}$ là một bit, nó có thể mô tả cho các trạng thái trừu tượng như đúng/sai, có/không, được/mất, ... nên thường được gọi rõ là **bit logic** (logical bit). Bit logic có thể được hiện thực bằng các hệ thống vật lý hai

¹thuật ngữ tính toán lượng tử (quantum computation) thiên về mô hình tính toán và thuật toán còn điện toán lượng tử (quantum computing) thiên về kỹ thuật, công nghệ và sự triển khai của tính toán lượng tử trong thực tế.

trạng thái, được gọi là **bit vật lý** (physical bit).² Bảng 2.1 minh họa vài trường hợp thông dụng.

Hệ thống vật lý	Trạng thái	
Đồng xu	xấp	ngửa
Công tắc điện	tắt	bật
Đèn pin	tối	sáng
Còi	tit	te
Điện thế mạch	thấp	cao
Dòng điện mạch	không	có
Bit (logic)	0	1

Bảng 2.1: Bit logic và bit vật lý.

Dãy gồm n bit

$$s = s_1 s_2 \dots s_n = (s_1, s_2, \dots, s_n) \in \mathbb{B}^n \quad (2.1)$$

được gọi là **chuỗi n bit** (bit string) hay **chuỗi nhị phân** (binary string) độ dài n . Ta cũng kí hiệu $|s| = n$. Chuỗi 4 bit thường được gọi là **nibble**, 8 bit được gọi là **byte**. Các chuỗi n bit gồm toàn 0 và toàn 1 được kí hiệu lần lượt là $0^n, 1^n$.

Ví dụ 2.1.1. Để mô tả mặt ra khi tung một đồng xu ta chỉ cần dùng 1 bit, với qui ước 0 là mặt xấp còn 1 là mặt ngửa. Một xúc xắc khi tung có thể ra các mặt từ 1 chấm đến 6 chấm. Để mô tả mặt ra của xúc xắc ta có thể dùng chuỗi 3 bit với qui ước như Bảng 2.2.³ Lưu ý, vì có 8 chuỗi 3 bit khác nhau trong khi chỉ có 6 mặt nên ta bỏ không dùng 2 chuỗi 000, 111. \square

Chuỗi 3 bit	Mặt ra của xúc xắc
000	-
001	1 chấm
010	2 chấm
011	3 chấm
100	4 chấm
101	5 chấm
110	6 chấm
111	-

Bảng 2.2: Chuỗi 3 bit mô tả mặt ra khi tung xúc xắc.

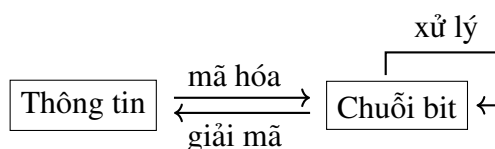
²chữ logic có nghĩa là “luân lý”, là cốt lõi, bỏ qua các chi tiết thực tế vật lý.

³ta cũng có thể dùng các qui ước khác.

2.1.2 Mã hóa thông tin

Một bit chỉ mô tả được trạng thái có 2 giá trị. Bằng cách ghép các bit, tức là dùng chuỗi bit, ta mô tả được trạng thái nhiều giá trị hơn. Chẳng hạn, chuỗi 2 bit mô tả được 4 giá trị còn một nibble mô tả được 16 giá trị. Tổng quát, chuỗi n bit mô tả được 2^n giá trị, chẳng hạn, một byte mô tả được $2^8 = 256$ giá trị khác nhau.

Để lưu trữ, truyền nhận hay xử lý trên máy tính, **thông tin** (information) cần được biểu diễn thành các chuỗi bit. Quá trình này thường được gọi là **mã hóa** (encode). Các chuỗi bit, sau đó, được diễn giải lại thành thông tin bằng quá trình ngược lại gọi là **giải mã** (decode). Việc xử lý thông tin trên máy tính, như vậy, là xử lý các chuỗi bit. Hình 2.1 minh họa các quá trình này.



Hình 2.1: Mã hóa, giải mã và xử lý thông tin.

Ví dụ 2.1.2. Mã Morse (Morse code)⁴ là một phương pháp viễn thông (truyền thông tin đi xa) được dùng phổ biến trong thế kỷ 19-20. Mục tiêu là truyền tải các thông điệp gồm các chữ cái, chữ số và dấu câu bằng 2 tín hiệu cơ bản là: dit (dot, kí hiệu là \cdot) và dah (dash, kí hiệu là $-$). Hình 2.2 là bảng mã Morse quốc tế.

Mã Morse có thể được hiện thực bằng các loại tín hiệu, thiết bị vật lý khác nhau, chẳng hạn, nếu dùng còi (âm thanh) thì có thể qui ước: tit (âm ngắn) là \cdot , te (âm dài) là $-$ và nghỉ (im lặng) là space. Lưu ý, như Hình 2.2 cho thấy, độ dài tín hiệu rất quan trọng trong mã Morse. Chẳng hạn, nếu dùng còi thì te phải dài gấp 3 lần tit hay nghỉ giữa mỗi lần tit-te trong cùng một kí tự phải bằng tit, ...

Vì bit không có thông tin độ dài nên mã Morse có thể được mã hóa thành chuỗi bit bằng “lược đồ mã” (encoding scheme) đơn giản là:⁵ 1 là tín hiệu, 0 là nghỉ và số lượng bit mô tả độ dài. Cụ thể, 1 là \cdot , 111 là $-$, 0 là nghỉ trong kí tự, 000 là nghỉ giữa các kí tự, 0000000 là nghỉ giữa các từ. Chẳng hạn, thông điệp cầu cứu SOS có mã Morse là $\cdot \cdot \cdot - - - \cdot \cdot \cdot$ (không có nghỉ giữa các kí tự) có chuỗi bit là

10101011101110111010101

□

⁴được đặt tên theo Samuel Morse (1791 – 1872), một nhà phát minh người Mỹ.

⁵có các lược đồ mã phức tạp nhưng hiệu quả hơn, xem chi tiết tại https://en.wikipedia.org/wiki/Morse_code.

International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.

A	• —	U	• • —
B	— • • •	V	• • • —
C	— • — •	W	— • —
D	— • •	X	— • • —
E	•	Y	— • — —
F	• • — •	Z	— — • •
G	— — •		
H	• • • •		
I	• •		
J	• — — —		
K	— • —	1	• — — — —
L	• — • •	2	• • — — —
M	— —	3	• • • — —
N	— •	4	• • • • —
O	— — —	5	• • • • •
P	• — — •	6	— • • • •
Q	— — • —	7	— — • • •
R	• — •	8	— — — • •
S	• • •	9	— — — — •
T	—	0	— — — — —

Hình 2.2: Bảng mã Morse quốc tế (nguồn: Wikipedia).

2.1.3 Số nhị phân

Chuỗi bit được dùng để mã hóa mọi thông tin trên máy tính mà quan trọng nhất trong số đó là mã hóa các số nguyên không âm bằng hệ thống **số nhị phân** (binary number). Trước hết, để thuận tiện, các bit trong chuỗi n bit s được kí hiệu là

$$s = s_{n-1}s_{n-2} \dots s_1s_0.$$

Lưu ý, thay vì đánh chỉ số các bit từ trái sang phải và bắt đầu từ 1 như trong (2.1), ta đã đánh chỉ số từ phải sang trái và bắt đầu từ 0. Khi đó, giá trị số nguyên mà s biểu diễn, kí hiệu $[s]$, được định nghĩa là

$$[s] = s_0 + 2s_1 + \dots + 2^{n-2}s_{n-2} + 2^{n-1}s_{n-1} = \sum_{i=0}^{n-1} s_i 2^i. \quad (2.2)$$

Chẳng hạn, byte $s = 11000101$ biểu diễn cho số nguyên

$$[s] = \sum_{i=0}^7 s_i 2^i = 2^0 + 2^2 + 2^6 + 2^7 = 1 + 4 + 64 + 128 = 197.$$

Tổng quát, trong **hệ thống số theo hàng** (positional numeral system) **cơ số** (base, radix) b , ta dùng chuỗi $s = s_{n-1} \dots s_1 s_0$ gồm các kí số $s_i \in \{0, 1, \dots, b-1\}$, $i = 0, \dots, n-1$ biểu diễn cho số nguyên

$$[s] = \sum_{i=0}^{n-1} s_i b^i. \quad (2.3)$$

Theo chỉ số tăng dần (từ phải sang trái), ta thường gọi các kí số chỉ số nhỏ (phải) là hàng thấp và chỉ số lớn (trái) là hàng cao.

Đối chiếu (2.2) với (2.3), ta thấy số nhị phân là hệ thống số theo hàng cơ số 2 với 2 kí số là $\{0, 1\}$. Ta cũng nói bit tận phải là bit thấp nhất và bit tận trái là bit cao nhất, tương ứng chính là bit cuối cùng và bit đầu tiên nếu tính từ trái sang phải.

Trong cách viết số hằng ngày, ta thường dùng hệ thống số theo hàng cơ số 10 được gọi là **số thập phân** (decimal number), dùng 10 kí số là $\{0, 1, \dots, 9\}$, chẳng hạn, số thập phân 197 có giá trị

$$1 \times 10^2 + 9 \times 10^1 + 7 \times 10^0 = 100 + 90 + 7.$$

Vì một nibble, gồm 4 bit, mô tả được 16 con số từ số 0 đến số 15 nên nếu kí hiệu các số này bằng các kí số như trong Bảng 2.3 thì ta có thể viết số nhị phân bằng **số thập lục phân** (hexadecimal number) cơ số 16 ngắn gọn hơn.⁶ Chẳng hạn, byte $s = 1100\ 0101$ gồm 2 nibble, được viết thành số thập lục phân với 2 kí số là C5 có giá trị là

$$5 \times 16^0 + 12 \times 16^1 = 5 + 192 = 197.$$

Bin	Dec	Hex	Bin	Dec	Hex
0000	0	0	1000	8	8
0001	1	1	1001	9	9
0010	2	2	1010	10	A
0011	3	3	1011	11	B
0100	4	4	1100	12	C
0101	5	5	1101	13	D
0110	6	6	1110	14	E
0111	7	7	1111	15	F

Bảng 2.3: Các nibble và kí số thập lục phân.

⁶Bin, Dec, Hex thường được viết tắt cho binary, decimal và hexadecimal number.

Nhận xét, dùng chuỗi n bit ta biểu diễn được các số nguyên không âm từ 0 đến $2^n - 1$, do đó, để biểu diễn cho số nguyên không âm N ta cần dùng số bit là

$$n = \begin{cases} 1 & N = 0, \\ \lfloor \log N \rfloor + 1 & N \geq 1. \end{cases} \quad (2.4)$$

Trong đó \log được hiểu là logarith cơ số 2 và kí hiệu $\lfloor x \rfloor$ chỉ số nguyên lớn nhất không quá x .

Ví dụ 2.1.3. Để mã văn bản, là chuỗi các kí tự, một hệ thống mã hiện đại hơn mã Morse (Ví dụ 2.1.2), được dùng phổ biến trên các máy tính ngày nay, là **mã ASCII** (American Standard Code for Information Interchange). Mã ASCII cũng đơn giản hơn mã Morse ở chỗ mỗi ký tự được mã bằng chuỗi 8 bit (tức là 1 byte).⁷ Bảng 2.4 mô tả mã của vài kí tự thông dụng.⁸ Để mã cho văn bản nhiều kí tự, ta

Bin	Dec	Hex	Kí tự	Bin	Dec	Hex	Kí tự
0010 0000	32	20	(space)	0100 0001	65	41	A
0010 0001	33	21	!	0100 0002	66	42	B
0010 0010	34	22	"	⋮	⋮	⋮	⋮
0010 0011	35	23	#	0101 1010	90	5A	Z
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0011 0000	48	30	0	0110 0001	97	61	a
0011 0001	49	31	1	0110 0010	98	62	b
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0011 1001	57	39	9	0111 1010	122	7A	z

Bảng 2.4: Mã ASCII của vài kí tự thông dụng.

đơn giản ghép các byte mã của từng kí tự lại theo thứ tự, chẳng hạn thông điệp “Ok!” được mã bằng 3 byte (chuỗi 24 bit) sau

$$\underbrace{01001111}_O \underbrace{01101011}_k \underbrace{00100001}_!$$

Vì mã ASCII chỉ dùng 1 byte cho mỗi ký tự nên chỉ mã được 256 kí tự khác nhau. Để mã cho lượng lớn các kí hiệu (bao gồm kí tự tiếng Việt, Trung, ... các kí hiệu Toán, Emoji, ...), một bảng mã mở rộng mã ASCII hay được dùng là mã Unicode (Bài tập 2.7). □

⁷vốn dĩ mã ASCII chỉ dùng 7 bit cho mỗi ký tự, nhưng trên máy tính, để chẵn byte, thông thường dùng đủ 8 bit.

⁸xem chi tiết và đầy đủ mã ASCII tại <https://en.wikipedia.org/wiki/ASCII>.

Số nhị phân cũng có thể được dùng để mã hóa số thực. Cụ thể, chuỗi số nhị phân

$$s = 0.s_1s_2\dots s_m, \quad s_i \in \{0, 1\},$$

biểu diễn cho số thực

$$[s] = \frac{s_1}{2} + \frac{s_2}{4} + \dots + \frac{s_m}{2^m} = \sum_{i=1}^m \frac{s_i}{2^i} = \sum_{i=1}^m s_i 2^{-i}.$$

Dấu chấm (.) được gọi là “dấu chấm nhị phân”. Nhận xét (♣)

$$[0.s_1s_2\dots s_m] = \sum_{i=1}^m s_i 2^{-i} = \frac{1}{2^m} \sum_{i=1}^m s_i 2^{m-i} = \frac{[s_1s_2\dots s_m]}{2^m}.$$

Chẳng hạn, số nhị phân $s = 0.1011$ biểu diễn cho số thực

$$[s] = \frac{1}{2} + \frac{1}{8} + \frac{1}{16} = 0.6875 = \frac{[1011]}{16} = \frac{11}{16}.$$

Có thể nói các tính toán trên số nhị phân tương tự như các tính toán trên số thập phân quen thuộc với cơ số 10 được thay bằng cơ số 2. Chẳng hạn khi chia cho 2 ta dời dấu chấm nhị phân sang trái còn nhân cho 2 thì dời dấu chấm nhị phân sang phải một kí số.

2.2 Cổng và mạch logic

Sau khi thông tin được mã hóa thành chuỗi bit, các xử lý trên thông tin chính là các tính toán trên chuỗi bit. Về mặt Toán học, các tính toán này được mô hình bằng các ánh xạ

$$f : \mathbb{B}^n \rightarrow \mathbb{B}^m$$

biến chuỗi n bit **đầu vào** (input) $x \in \mathbb{B}^n$ thành chuỗi m bit **đầu ra** (output)

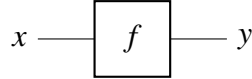
$$y = f(x) \in \mathbb{B}^m.$$

Phần sau trình bày cách mô tả f bằng các **mạch** (circuit), là một **mô hình tính toán** (model of computation) thông dụng (đặc biệt phù hợp khi mở rộng sang tính toán lượng tử).⁹

⁹Có các mô hình tính toán khác như máy Turing (Turing machine) hay RAM (Random-Access Machine), là mô hình gần gũi hơn với máy tính cổ điển hiện nay.

2.2.1 Cổng logic

Cổng logic mô hình các tính toán cơ bản nhất trên bit, còn gọi là các **phép toán** (operation). Đơn giản nhất là các phép toán trên một bit $f : \mathbb{B} \rightarrow \mathbb{B}$, khi cả đầu vào và đầu ra chỉ gồm 1 bit. Các phép toán này thường được vẽ bằng sơ đồ gồm **cổng** (gate) với một **đường dây** (wire) vào và ra như sau



Ta đọc sơ đồ từ trái sang phải, một cách hình tượng, bit đầu vào x sau khi đi qua cổng f biến thành bit đầu ra y .

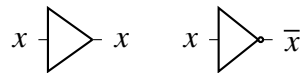
Vì một bit chỉ có thể có 2 trường hợp là 0, 1 nên ta chỉ có 4 phép toán 1 bit ứng với 4 ánh xạ khác nhau từ \mathbb{B} vào \mathbb{B} . 4 ánh xạ này được cho trong **bảng chân trị** (truth table) 2.5.

x	f_1	f_2	f_3	f_4
0	0	1	0	1
1	1	0	0	1

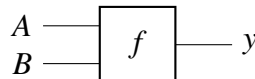
Bảng 2.5: Bảng chân trị của các cổng logic 1 bit.

- f_1 giữ nguyên đầu vào, $f_1(x) = x$, nên còn được kí hiệu là ID (identity).
- f_2 đảo (còn gọi là lật) đầu vào, $f_2(x) = \bar{x}$, nên còn được kí hiệu là NOT.
- f_3 đặt đầu ra là 0, $f_3(x) = 0$, nên còn được kí hiệu là CLEAR.
- f_4 đặt đầu ra là 1, $f_4(x) = 1$, nên còn được kí hiệu là SET.

Ta thấy ID “không làm gì cả” còn NOT có ý nghĩa luận lý là phủ định vì biến Đúng thành Sai và Sai thành Đúng (xem 1 là Đúng, 0 là Sai).¹⁰ ID và NOT thường được vẽ tương ứng như sau



Trường hợp đầu vào $x = (B, A) \in \mathbb{B}^2$ gồm 2 bit, đầu ra $y \in \mathbb{B}$ vẫn là 1 bit, ta vẽ sơ đồ như sau



¹⁰nhiều tài liệu khác, kí hiệu phủ định x , là $\neg x$, thay vì \bar{x} .

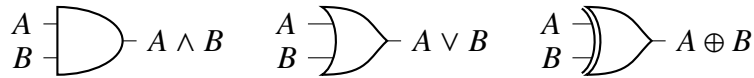
Lưu ý, ta thường vẽ bit thấp (A) ở trên, bit cao (B) ở dưới.

Vì 2 bit có thể có 4 trường hợp là 00, 01, 10, 11 nên ta có 4 đầu vào mà mỗi đầu vào có thể có 2 đầu ra là 0, 1 nên ta có 16 ánh xạ khác nhau từ \mathbb{B}^2 vào \mathbb{B} . Bảng chân trị 2.6 liệt kê các phép toán 2 bit điển hình.

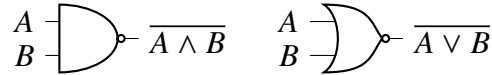
A	B	$A \wedge B$	$A \vee B$	$A \oplus B$	$\overline{A \wedge B}$	$\overline{A \vee B}$
0	0	0	0	0	1	1
0	1	0	1	1	1	0
1	0	0	1	1	1	0
1	1	1	1	0	0	0

Bảng 2.6: Bảng chân trị của các phép toán 2 bit điển hình.

Các phép toán \wedge, \vee, \oplus có ý nghĩa luận lý lần lượt là Và (Đúng nếu cả 2 cùng Đúng), Hoặc (Đúng nếu có ít nhất 1 trong 2 Đúng), Hoặc-Loại-Trừ (Đúng nếu có đúng 1 trong 2 Đúng) nên được gọi lần lượt là AND, OR, XOR (Exclusive OR) và được vẽ tương ứng như sau



NAND, NOR lần lượt là phủ định của AND, OR và được vẽ tương ứng như sau



Vì có ý nghĩa luận lý nên các phép toán này thường được gọi là các phép toán logic và các cổng tương ứng được gọi là **cổng logic** (logic gate).

Các phép toán 2 bit có thể được mở rộng tự nhiên thành phép toán trên 2 chuỗi n bit bằng cách thao tác trên từng cặp bit. Chẳng hạn, cho $x = x_{n-1} \dots x_1 x_0, y = y_{n-1} \dots y_1 y_0$ ta định nghĩa $z = x \oplus y$ là chuỗi $z = z_{n-1} \dots z_1 z_0$ thỏa

$$z_i = x_i \oplus y_i, \quad i = 0, \dots, n-1.$$

Từ chuỗi n bit $x = x_{n-1} \dots x_1 x_0$, bằng cách thao tác lần lượt trên từng bit của x , ta có thể “thu” x thành 1 bit, chẳng hạn

$$\text{AND}(x) = x_{n-1} \wedge \dots \wedge x_1 \wedge x_0,$$

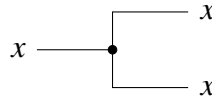
$$\text{OR}(x) = x_{n-1} \vee \dots \vee x_1 \vee x_0,$$

$$\text{XOR}(x) = x_{n-1} \oplus \dots \oplus x_1 \oplus x_0.$$

Ta thấy

- $\text{AND}(x) = 1$ khi và chỉ khi tất cả các bit của x đều là 1.
- $\text{OR}(x) = 1$ khi và chỉ khi có bit nào đó của x là 1.
- $\text{XOR}(x) = 1$ khi và chỉ khi số lượng bit 1 của x là số lẻ, do đó, $\text{XOR}(x)$ được gọi là **tính lẻ** (parity) của x nên còn được kí hiệu là $\text{PARITY}(x)$.

Đặc biệt, phép toán “sao chép” bit: $\mathbb{B} \rightarrow \mathbb{B}^2$ biến 0 thành 00, 1 thành 11 được mô tả bằng “**cổng chẻ**” (fanout) như sau



Ví dụ 2.2.1. Mạch nửa cộng (Half Adder). Khi chuỗi bit là số nhị phân, **phép toán số học** (arithmetic operation) quan trọng nhất khi đó là phép cộng. Đơn giản nhất là cộng 1 bit: cộng bit A với bit B được tổng là bit S và nhớ là bit C . Vì có 2 bit đầu vào và 2 bit đầu ra nên đây là phép toán $\mathbb{B}^2 \rightarrow \mathbb{B}^2$. Từ qui tắc cộng: 0 cộng 0 được 0 nhớ 0, 0 cộng 1 hoặc 1 cộng 0 đều được 1 nhớ 0 và 1 cộng 1 được 0 nhớ 1, ta có Bảng chân trị [2.7](#).

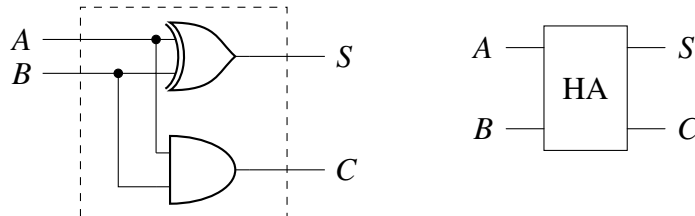
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Bảng 2.7: Bảng chân trị của phép cộng 1 bit.

Mặc dù là phép toán số học nhưng ta thấy phép cộng 1 bit có thể được mô tả bằng các phép toán logic như sau

$$S = A \oplus B, \quad C = A \wedge B.$$

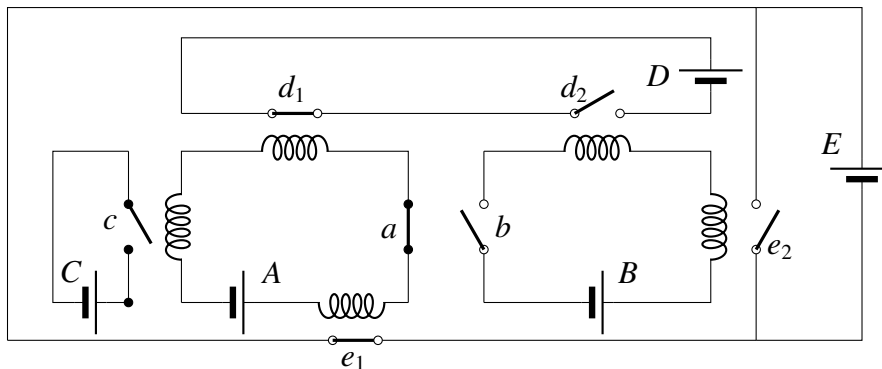
Do đó ta có sơ đồ mô tả phép cộng 1 bit, còn được gọi là **bộ nửa cộng** (half adder)



Chi tiết sơ đồ “đi dây” và cổng (2 cổng fanout, 1 cổng XOR, 1 cổng AND) được vẽ bên trái. Bên phải là dạng “**hộp đen**” (black box), chỉ mô tả “giao tiếp” vào/ra

các bit và ẩn đi chi tiết “cài đặt”. Tên gọi HA (half adder) mô tả chức năng của hộp đen. \square

Tương tự việc hiện thực bit logic bằng bit vật lý, các cổng logic cũng có thể được hiện thực bằng nhiều hệ thống vật lý khác nhau. Chẳng hạn nếu dùng dòng điện trong mạch (có dòng điện là 1, không có dòng điện là 0) thì các cổng NOT, AND, OR có thể được hiện thực bằng mạch điện như minh họa trong Hình 2.3.



Hình 2.3: Mạch điện minh họa cách hiện thực các cổng logic.

Cụ thể, các bit đầu vào là A, B . Trong mạch A , công tắc a mặc định được đóng (kí hiệu bằng chấm tròn đặc ở 2 đầu) nên có dòng điện qua pin A nên bit $A = 1$. Ngược lại, trong mạch B , công tắc b mặc định được mở (kí hiệu bằng chấm tròn rỗng ở 2 đầu) nên không có dòng qua pin B nên bit $B = 0$. Công tắc c trong mạch C mặc định đóng, nếu có dòng trong A thì cuộn cảm của A sẽ tác động làm mở c do đó không có dòng qua C . Ngược lại, nếu không có dòng trong A thì cuộn cảm của A không hoạt động, c đóng nên có dòng qua C . Như vậy, $C = \bar{A} = 0$. Tương tự ta có $D = A \wedge B = 0$ và $E = A \vee B = 1$.

2.2.2 Mạch logic

Các tính toán phức tạp trên chuỗi nhiều bit được thực hiện bằng cách kết hợp các cổng logic cơ bản và thường được mô tả bằng sơ đồ **mạch logic** (logic circuit). Thông thường, tính toán càng phức tạp thì càng dùng nhiều cổng, hơn nữa, số cổng cũng thường tăng theo số bit đầu vào. Nếu gọi số bit đầu vào là n thì số cổng trong mạch, tính theo n , thường được dùng để đánh giá **độ phức tạp của mạch** (circuit complexity).

Ví dụ 2.2.2. Mạch toàn cộng (Full Adder). Mạch trong Ví dụ 2.2.1 được gọi là mạch nửa cộng vì ta không để ý đến nhớ đầu vào. Đầy đủ hơn, khi cộng bit A với

bit B ta cần tính đến bit nhớ đầu vào C_{in} . Đầu ra gồm bit tổng S và bit nhớ đầu ra C_{out} được tính từ các bit đầu vào A, B, C_{in} theo qui tắc được cho trong Bảng 2.8.

A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Bảng 2.8: Bảng chân trị của phép cộng 1 bit đầy đủ.

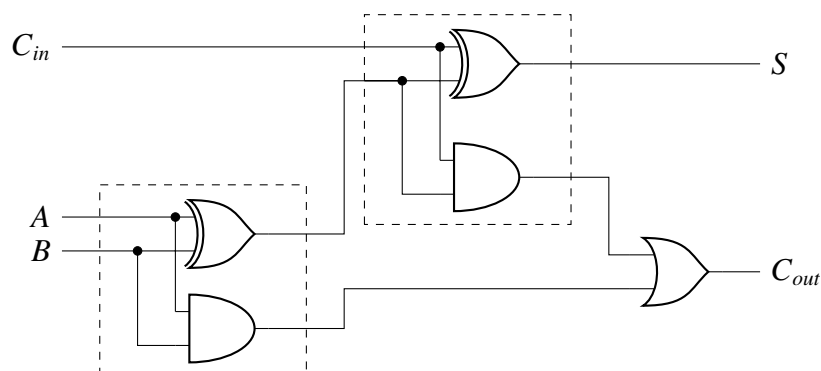
Từ bảng chân trị ta thấy

$$S = A \oplus B \oplus C_{in}, \quad C_{out} = (A \wedge B) \vee (C_{in} \wedge (A \oplus B)).$$

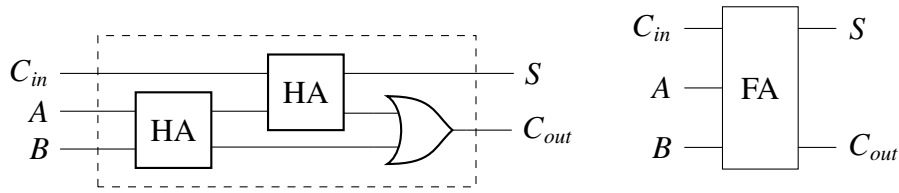
Khi biểu thức logic trở nên phức tạp ta thường kí hiệu $A \wedge B, A \vee B$ lần lượt là $AB, A + B$. Hơn nữa, ta thường cho phép toán \wedge ưu tiên hơn các phép toán $+, \oplus$. Chẳng hạn, các biểu thức logic trên có thể được viết gọn hơn là

$$S = A \oplus B \oplus C_{in}, \quad C_{out} = AB + C_{in}(A \oplus B).$$

Sơ đồ mạch chi tiết của **bộ cộng đầy đủ** (full adder) được vẽ như bên dưới



Quan sát sơ đồ trên, ta thấy bộ cộng đầy đủ có thể được ghép từ 2 bộ nửa cộng và 1 cổng OR. Như vậy, một sơ đồ mạch có tính “module hóa” cao hơn cho bộ cộng đầy đủ có thể được vẽ như hình bên trái sau



Sơ đồ bên phải mô tả hộp đen của bộ cộng đầy đủ.

Lưu ý, ta cũng có thể vẽ ngay được sơ đồ module hóa của bộ cộng đầy đủ mà không cần thông qua sơ đồ chi tiết ở trên bằng “**thuật toán**” (algorithm) 1. Khái niệm thuật toán sẽ được trình bày chi tiết ở Phần 2.3.1. Hiện giờ ta có thể xem thuật toán là một dãy các bước tính đầu ra từ đầu vào.

Thuật toán 1 Thuật toán cộng 1 bit đầy đủ.

Input: A, B, C_{in}

Output: tổng S và nhớ C_{out} từ $A + B + C_{in}$

- 1: $S_1, C_1 \leftarrow \text{HA}(A, B)$ \triangleright cộng A với B bằng bộ nửa cộng được tổng S_1 , nhớ C_1
 - 2: $S_2, C_2 \leftarrow \text{HA}(C_{in}, S_1)$
 - 3: $S \leftarrow S_2$
 - 4: $C_{out} \leftarrow \text{OR}(C_1, C_2)$ $\triangleright C_{out} \leftarrow C_1 + C_2$
-

Lưu ý, trong thuật toán trên, vì C_1, C_2 không thể cùng là 1 (\clubsuit), nên $C_1 + C_2$ không có nhớ và $C_1 + C_2 = C_1 \vee C_2$. \square

Ví dụ 2.2.3. (tiếp Ví dụ 2.2.2) **Mạch cộng RCA (Ripple Carry Adder).** Để cộng 2 số nhị phân nhiều bit, ta cộng lần lượt mỗi bit (có nhớ) từ bit thấp đến bit cao (từ phải sang trái). Chẳng hạn, số nhị phân 0110 (giá trị 6) cộng số nhị phân 1111 (giá trị 15) được số nhị phân 10101 (giá trị 21) như sơ đồ bên trái dưới đây.

$$\begin{array}{r}
 \text{(nhớ)} \ 11100 \\
 \phantom{\text{(nhớ)}} \ 0110 \\
 + \phantom{\text{(nhớ)}} \ 1111 \\
 \hline
 \text{(tổng)} \ 10101
 \end{array}
 \qquad
 \begin{array}{r}
 \text{(nhớ)} \ C_4 C_3 C_2 C_1 C_0 \\
 \phantom{\text{(nhớ)}} \ A_3 A_2 A_1 A_0 \\
 + \phantom{\text{(nhớ)}} \ B_3 B_2 B_1 B_0 \\
 \hline
 \text{(tổng)} \ S_4 S_3 S_2 S_1 S_0
 \end{array}$$

Sơ đồ bên phải cho thấy cách cộng số nhị phân 4 bit A với số nhị phân 4 bit B được tổng 5 bit S . Lưu ý, bit nhớ đầu tiên C_0 có giá trị 0 (ban đầu chưa có nhớ) và S_4 chính là bit nhớ C_4 (bit nhớ sau cùng được ghi xuống tổng).

Thuật toán 2 cho thấy cách cộng 2 số nhị phân n bit $A, B \in \mathbb{B}^n$ để được số nhị phân $n + 1$ bit $S \in \mathbb{B}^{n+1}$ bằng cách dùng các bộ cộng 1 bit đầy đủ. Thuật toán này có thể được cài đặt bằng “phần mềm” (software) với các cấu trúc lặp hoặc bằng

Thuật toán 2 Thuật toán cộng số nhị phân nhiều bit.

Input: A, B là các số nhị phân n bit

Output: tổng $S = A + B$ là số nhị phân $n + 1$ bit

1: $C_0 \leftarrow 0$

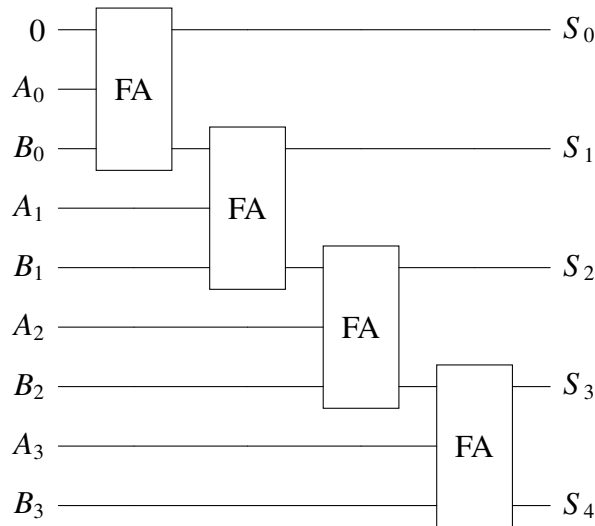
2: **Lặp** $i = 1, \dots, n$

3: $S_{i-1}, C_i \leftarrow \text{FA}(C_{i-1}, A_{i-1}, B_{i-1})$

4: $S_n \leftarrow C_n$

“phần cứng” (hardware) với nhiều cổng logic như sơ đồ mạch ở Hình 2.4 minh họa trường hợp cộng 2 số nhị phân 4 bit.

Trong Thuật toán 2, ta thấy mạch cộng 2 số nhị phân n bit cần n bộ cộng FA mà mỗi FA cần 5 cổng logic nên tổng cộng mạch cần $5n$ cổng logic. Do đó, ta cũng nói rằng Thuật toán 2 có độ phức tạp mạch là tuyến tính, nghĩa là số cổng cần dùng tỉ lệ thuận với độ dài của chuỗi nhị phân đầu vào. \square



Hình 2.4: Mạch cộng 2 số nhị phân 4 bit.

2.2.3 Tập cổng toàn năng, đại số Boole và đơn giản mạch

Ta đã thấy, trong tính toán cổ điển, thông tin được mã bằng các chuỗi bit, việc xử lý thông tin được thực hiện bằng cách dùng kết hợp các cổng logic, là các thao tác cơ bản trên bit. Những cổng nào là “cơ bản nhất” và tập gồm số ít những cổng cơ bản nào là đủ dùng cho mọi thao tác xử lý?

Trước hết với $F : \mathbb{B}^n \rightarrow \mathbb{B}^m$ là phép toán có n bit đầu vào và m bit đầu ra, ta có thể thiết kế mạch riêng cho từng bit đầu ra (mặc dù trong nhiều trường hợp ta có thể dùng chung các thành phần của mạch cho nhiều bit đầu ra để có mạch hiệu quả hơn). Tiếp đến, mọi phép toán $f : \mathbb{B}^n \rightarrow \mathbb{B}$ đều có thể được thực hiện với chỉ các cổng NOT, AND và OR qua **dạng tuyển chuẩn** (Disjunctive Normal Form, DNF) từ bảng chân trị của f .

Ví dụ, cho $f : \mathbb{B}^3 \rightarrow \mathbb{B}$ có 3 bit đầu vào A, B, C và 1 bit đầu ra với Bảng chân trị 2.9. Ta thấy f đúng (có giá trị 1) tại Dòng 2 hoặc Dòng 4 hoặc Dòng 8. Với Dòng

A	B	C	$f(A, B, C)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Bảng 2.9: Bảng chân trị của một phép toán $f : \mathbb{B}^3 \rightarrow \mathbb{B}$.

2, ta cần không A ($A = 0$) và không B ($B = 0$) và C ($C = 1$) nên cần $\bar{A} \bar{B} C$. Tương tự với Dòng 4 ta cần $\bar{A} B C$ và Dòng 8 cần $A B C$. Từ đó, f có thể được viết ở dạng DNF là

$$f(A, B, C) = \bar{A} \bar{B} C \vee \bar{A} B C \vee A B C. \quad (2.5)$$

Như vậy chỉ cần dùng 3 cổng là NOT, AND và OR ta có thể thực hiện mọi thao tác xử lý trên bit nên tập cổng {NOT, AND, OR} được gọi là **tập cổng toàn năng** (universal gate set).

Bằng cách xét các trường hợp, ta thấy, với mọi bit A, B

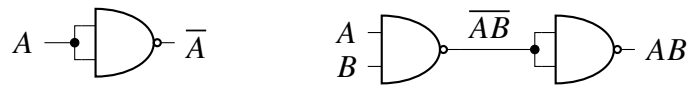
$$A \vee B = \overline{\bar{A} \bar{B}} \quad (2.6)$$

nghĩa là cổng OR có thể được thực hiện bằng cổng NOT và cổng AND nên {NOT, AND} cũng là tập cổng toàn năng. Tương tự ta cũng có

$$AB = \overline{\bar{A} \vee \bar{B}} \quad (2.7)$$

nên {NOT, OR} cũng là tập cổng toàn năng.

Ta cũng thấy $\bar{A} = \overline{AA}$ và $AB = \overline{\bar{A} \bar{B}}$ nên cổng NOT và cổng AND có thể được cài đặt bằng cổng NAND như sau



Vậy {NAND} là tập cổng toàn năng (nghĩa là chỉ cần dùng cổng NAND là có thể thực hiện được mọi thao tác logic). Tương tự, {NOR} cũng là tập cổng toàn năng.

Nếu dùng tập cổng {NOT, AND, OR} thì với f ở (2.5) ta cần 11 cổng (3 cổng NOT, 2 cổng OR và 6 cổng AND). Liệu ta có thể cài đặt f với ít cổng hơn? (Dĩ nhiên là dùng càng nhiều cổng thì càng tốn kém).

Ta có thể thực hiện việc đơn giản mạch bằng cách dùng các **tương đương logic** (logical equivalence) tương tự như (2.6) hay (2.7). Cũng bằng cách kiểm tra các trường hợp (chẳng hạn dùng bảng chân trị), ta có nhiều tương đương logic khác. Bảng 2.10 liệt kê vài qui tắc thông dụng. Tập các qui tắc tương đương logic cùng với việc dùng chúng trong các biến đổi (chẳng hạn đơn giản biểu thức logic) được gọi là **đại số Bool** (Boolean algebra).¹¹

Đẳng thức	Luật	Đẳng thức	Luật
$A(BC) = (AB)C$ $A \vee (B \vee C) = (A \vee B) \vee C$	Kết hợp	$A1 = A$ $A \vee 0 = A$	Trung hòa
$AB = BA$ $A \vee B = B \vee A$	Giao hoán	$A0 = 0$ $A \vee 1 = 1$	Thống trị
$A(B \vee C) = AB \vee AC$ $A \vee (BC) = (A \vee B)(A \vee C)$	Phân phối	$A\bar{A} = 0$ $A \vee \bar{A} = 1$	Phản bù
$AA = A$ $A \vee A = A$	Lũy đẳng	$\overline{AB} = \bar{A} \vee \bar{B}$ $\overline{A \vee B} = \bar{A} \bar{B}$	De Morgan
$\overline{\bar{A}} = A$	Phủ định kép		

Bảng 2.10: Các luật tương đương logic thông dụng.

¹¹được đặt tên theo George Boole, nhà logic học người Anh.

Ví dụ, dùng các tương đương logic ta đơn giản biểu thức (2.5) như sau

$$\begin{aligned}
 f(A, B, C) &= \bar{A} \bar{B} C \vee \bar{A} B C \vee A B C \\
 &= \bar{A} \bar{B} C \vee \bar{A} B C \vee \bar{A} B C \vee A B C \quad (\text{lũy đẳng}) \\
 &= \bar{A}(\bar{B} \vee B)C \vee (\bar{A} \vee A)BC \quad (\text{phân phối}) \\
 &= \bar{A}1C \vee 1BC \quad (\text{phần bù}) \\
 &= \bar{A}C \vee BC \quad (\text{đơn vị}) \\
 &= (\bar{A} \vee B)C \quad (\text{phân phối})
 \end{aligned}$$

Như vậy, ta chỉ cần 3 cổng để cài đặt f (thay vì 11 cổng như trong (2.5)).

2.2.4 Cổng khả nghịch

Cổng logic G thực hiện phép toán logic $f : \mathbb{B}^n \rightarrow \mathbb{B}^m$ là một ánh xạ từ tập các chuỗi n bit sang tập các chuỗi m bit. Nếu f là song ánh, ta nói cổng G là **khả nghịch** (reversible). Khi đó, có tương ứng 1-1 giữa chuỗi đầu ra và chuỗi đầu vào, nên ta cần $n = m$ và có thể xem f là một hoán vị của \mathbb{B}^n (xem Phụ lục A.3). Hơn nữa, nếu G' là cổng thực hiện ánh xạ ngược f^{-1} của f thì ta nói G' là **cổng nghịch đảo** (inverse gate) của G , cũng kí hiệu G' là G^{-1} . Nói nôm na, G^{-1} “undo” G .

Ví dụ, trong 4 cổng logic 1 bit ở Bảng 2.5, ta có cổng ID và cổng NOT là khả nghịch (các cổng CLEAR, SET không khả nghịch). ID giữ nguyên bit còn NOT “lật” bit, tức là hoán đổi $0 \leftrightarrow 1$. Cổng nghịch đảo của ID và NOT đều là chính nó. NOT “undo” NOT, nói nôm na, 2 lần NOT thì cũng như không làm gì cả. Các cổng 2 bit ở Bảng 2.6 không khả nghịch vì chỉ có 1 bit đầu ra trong khi có 2 bit đầu vào nên các cổng này không đơn ánh. Chẳng hạn với cổng AND ta thấy cả 00, 01, 10 đều cho đầu ra là 0 nên với đầu ra 0 ta không xác định được đầu vào.

Ví dụ 2.2.4. Bảng 2.11 cho thấy chân trị của 3 phép biến đổi trên chuỗi 2 bit ($\mathbb{B}^2 \rightarrow \mathbb{B}^2$) với đầu vào là AB và đầu ra là CD .

HA				CNOT _{AB}				CNOT _{BA}			
A	B	C	D	A	B	C	D	A	B	C	D
0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	1	0	1	0	1	1	1
1	0	1	0	1	0	1	1	1	0	1	0
1	1	0	1	1	1	1	0	1	1	0	1

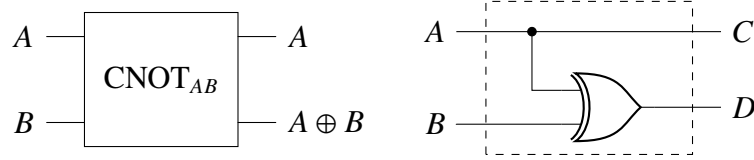
Bảng 2.11: Bảng chân trị của một số cổng biến đổi 2 bit.

Phép toán bên trái thực hiện phép cộng 1 bit với C là bit tổng và D là bit nhớ mà ta có thể gọi là cổng HA như trong Ví dụ 2.2.1. Ta thấy rằng HA không khả nghịch vì với đầu ra 10 ta không xác định được đầu vào (là 01 hay 10). Với phép toán ở giữa, ta thấy

$$C = A, \quad D = \begin{cases} B & \text{nếu } A = 0, \\ \overline{B} & \text{nếu } A = 1. \end{cases} \quad (2.8)$$

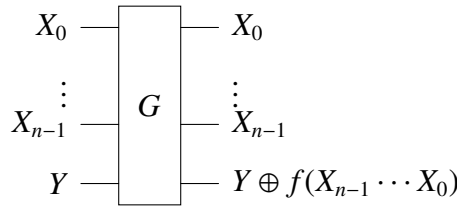
Như vậy A điều khiển thao tác NOT trên B nên cổng này thường được gọi là **cổng điều khiển NOT** (Controlled-NOT gate) và được kí hiệu là CNOT_{AB} . Nhận xét, CNOT_{AB} khả nghịch với cổng nghịch đảo là chính nó (\clubsuit). Tương tự, cổng bên phải là CNOT_{BA} cũng khả nghịch với nghịch đảo là chính nó (\clubsuit).

Lưu ý, biểu thức (2.8) xác định đầu ra của cổng CNOT_{AB} cũng có thể được viết lại là $C = A, D = A \oplus B$ (\clubsuit) nên CNOT_{AB} có thể được mô tả như hình bên trái và được cài đặt bằng các cổng cơ bản như hình bên phải dưới đây.



□

Cổng khả nghịch đóng vai trò rất quan trọng trong tính toán (ta sẽ thấy các cổng trong tính toán lượng tử đều khả nghịch). Rất may, ta dễ dàng biến các cổng không khả nghịch thành khả nghịch (hay nói cách khác là tìm phiên bản khả nghịch của cổng không khả nghịch) bằng kĩ thuật đơn giản sau. Gọi $f : \mathbb{B}^n \rightarrow \mathbb{B}$ là phép toán biến chuỗi n bit đầu vào $X_{n-1} \cdots X_1 X_0$ thành 1 bit đầu ra Y . Xét cổng G gồm $n + 1$ bit đầu vào và đầu ra như Hình 2.5.



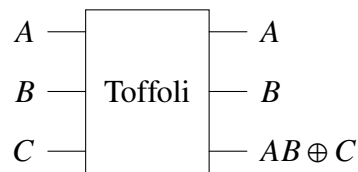
Hình 2.5: Cổng khả nghịch tính $f : \mathbb{B}^n \rightarrow \mathbb{B}$.

G giữ nguyên các bit đầu vào X_{n-1}, \dots, X_0 và dùng thêm một bit Y để chứa kết quả tính f . Nếu cho bit Y đầu vào là 0 thì ta có bit đầu ra tương ứng là

$$0 \oplus f(X_{n-1} \cdots X_0) = f(X_{n-1} \cdots X_0).$$

Ta cũng có G khả nghịch (Bài tập 2.15). Như vậy, G là phiên bản khả nghịch tính f . Kỹ thuật này cũng có thể được mở rộng dễ dàng cho trường hợp f có nhiều bit đầu ra (Bài tập 2.16).

Ví dụ 2.2.5. Ta thấy cổng AND có 2 bit đầu vào và 1 bit đầu ra nên không khả nghịch. Dùng kỹ thuật thiết kế mạch ở Hình 2.5 ta có cổng sau đây, thường được gọi là **cổng Toffoli** (Toffoli gate), là phiên bản khả nghịch của cổng AND



Khi C được đặt đầu vào là 0 thì đầu ra tương ứng là $AB \oplus 0 = AB = A \wedge B$. Lưu ý, khi C được đặt đầu vào là 1 thì đầu ra tương ứng là $AB \oplus 1 = \overline{AB} = \overline{A \wedge B}$ nên cổng Toffoli cũng là phiên bản khả nghịch của cổng NAND. Do đó cổng Toffoli là một cổng khả nghịch toàn năng.

Ta cũng thấy C bị phủ định khi và chỉ khi cả A, B đều là 1 nên cổng Toffoli cũng có thể được xem là mở rộng của cổng CNOT dùng 2 bit điều khiển và thường được gọi là CCNOT. \square

2.3 Bài toán, thuật toán và độ phức tạp tính toán

2.3.1 Bài toán và thuật toán

Bài toán (problem) là một yêu cầu tính toán.¹² Sau khi thông tin được mã hóa thành chuỗi bit, bài toán được hình thức thành phép toán $f : \mathbb{B}^n \rightarrow \mathbb{B}^m$ biến chuỗi n bit đầu vào thành chuỗi m bit đầu ra mà sau khi giải mã trở thành **lời giải** (solution, answer) của bài toán. Ở mức chi tiết nhất, yêu cầu của bài toán được mô tả bằng bảng chân trị của f . Tuy nhiên, trong nhiều trường hợp, bài toán được mô tả bằng “ngôn ngữ trừu tượng” hơn như các công thức Toán hay lời văn của các ngôn ngữ tự nhiên (natural language) như tiếng Việt, tiếng Anh, ...

Thuật toán (algorithm) là dãy các bước cụ thể để giải bài toán. Ở mức chi tiết nhất, các bước của thuật toán được mô tả bằng mạch logic hiện thực hay **cài đặt** (implement) f . Tuy nhiên, trong nhiều trường hợp, thuật toán được mô tả bằng các phương tiện trừu tượng hơn như mã giả (pseudocode),¹³ lưu đồ (flowchart) hay mã

¹²Để rõ ràng hơn, bài toán dạng này được gọi là **bài toán tính toán** (computational problem) hay **bài toán Toán học** (mathematical problem).

¹³<https://en.wikipedia.org/wiki/Pseudocode>.

nguồn (source code) của ngôn ngữ lập trình (programming language) nào đó như Python, C, ...

Ví dụ 2.3.1. Bài toán cộng 2 số nguyên 1 bit được hình thức thành một phép toán $\mathbb{B}^2 \rightarrow \mathbb{B}^2$ và được mô tả bằng bảng chân trị 2.7. Một thuật toán để giải bài toán này được cài đặt bằng mạch nửa cộng HA ở Ví dụ 2.2.1. Bài toán cộng đầy đủ (toàn cộng) được hình thức thành một phép toán $\mathbb{B}^3 \rightarrow \mathbb{B}^2$ và được mô tả bằng bảng chân trị 2.8. Một thuật toán để giải bài toán này được cài đặt chi tiết bằng mạch toàn cộng FA hay được mô tả bằng mã giả trong Thuật toán 1 ở Ví dụ 2.2.2.

Tổng quát hơn, trong Ví dụ 2.2.3, bài toán cộng 2 số nguyên n bit A, B để được tổng $S = A + B$ là số nguyên $n + 1$ bit được hình thức thành phép toán $f : \mathbb{B}^{2n} \rightarrow \mathbb{B}^{n+1}$. Ta không thể dùng bảng chân trị để mô tả f được nữa (bảng chân trị cho f sẽ có 2^{2n} dòng), ta chỉ mô tả ngắn gọn lời giải $S = A + B$. Một thuật toán để giải bài toán này được mô tả bằng mã giả trong Thuật toán 2 mà trường hợp $n = 4$ được mô tả chi tiết hơn bằng mạch “được module hóa” 2.4. \square

2.3.2 Độ phức tạp tính toán

Rõ ràng mạch dùng nhiều cổng logic sẽ phức tạp hơn mạch dùng ít cổng. Càng nhiều cổng càng tốn kém chi phí và tốn thời gian đợi mạch xử lý. Khi một thuật toán được mô tả bằng mạch logic, ta có thể dùng số lượng cổng (cơ bản) trong mạch để đánh giá sự phức tạp của thuật toán. Trường hợp thuật toán được mô tả bằng các phương tiện mức cao hơn (như mã giả), ta dùng số lượng “phép toán cơ bản cần thực hiện” để đánh giá sự phức tạp. Nếu xem các cổng cơ bản thực hiện các phép toán cơ bản thì hai cách đánh giá này như nhau. Dĩ nhiên, khi mạch được module hóa, ta không nên xem các cổng như FA, HA hay RCA là một cổng được vì chúng gồm nhiều cổng cơ bản bên trong.

Sự phức tạp của thuật toán cũng phụ thuộc số bit đầu vào n , thông thường, n càng lớn thì càng tốn nhiều phép toán. Chẳng hạn, mạch cộng RCA cho 2 số nguyên n bit cần $5n$ cổng logic, tăng tuyến tính theo n . Hơn nữa, trong vài trường hợp, số phép toán còn phụ thuộc đầu vào cụ thể chứ không chỉ số bit đầu vào. Khi đó ta thường dùng số phép toán trong “**trường hợp xấu nhất**” (worst-case), tức trường hợp tốn nhiều nhất, để đánh giá sự phức tạp. Như vậy, nếu gọi n là số bit đầu vào, gọi $T(n)$ là số phép toán cơ bản cần dùng nhiều nhất của thuật toán thì $T(n)$ được gọi là **độ phức tạp tính toán** (computational complexity) của thuật toán.

Kí hiệu **O -lớn** (Big-O notation) hay được dùng để đánh giá độ phức tạp tính toán. Cụ thể, ta nói

$$T(n) = O(g(n))$$

khi có số nguyên n_0 và số thực c sao cho

$$T(n) \leq cg(n), \forall n \geq n_0.$$

Nói nôm na, $T(n) = O(g(n))$ nghĩa là $T(n)$ bị chặn trên bởi $g(n)$ khi bỏ qua một hệ số tỉ lệ (c) và các trường hợp n nhỏ ($n < n_0$). Thông thường, $g(n)$ được chọn là dạng đơn giản nhất để mô tả các “cấp độ” (order) tăng trưởng theo n . Cách đánh giá “khi n đủ lớn” này còn được gọi là đánh giá **tiệm cận** (asymptotic). Ví dụ, trong mạch cộng RCA ta có độ phức tạp

$$T(n) = 5n = O(n)$$

là cấp độ “tuyến tính” (linear) theo n . Khi dùng kí hiệu O -lớn, ta cũng nên cố gắng chọn $g(n)$ “nhỏ nhất có thể”. Chẳng hạn ta cũng có $5n = O(n^2)$ hay thậm chí $5n = O(2^n)$ nhưng ta đã chọn $5n = O(n)$.

Một kí hiệu tiệm cận khác cũng hay được dùng là Θ -lớn (Big-Theta), ta nói $T(n) = \Theta(g(n))$ nếu $T(n) = O(g(n))$ và $g(n) = O(T(n))$, nghĩa là $T(n)$ “không lớn và cũng không nhỏ hơn” $g(n)$, tức là $T(n)$ “cỡ như” $g(n)$. Ví dụ

$$T(n) = 5n = \Theta(n).$$

Lưu ý, ta có $5n = O(n^2)$, nghĩa là $5n$ không quá n^2 nhưng ta không có chiều ngược lại nên $5n$ không cùng cỡ như n^2 ($5n$ “bé hơn hẳn” n^2).

Việc dùng đánh giá tiệm cận có nhiều lợi ích như

- Nếu một mạch G có k cổng cơ bản (không phụ thuộc số lượng bit đầu vào) thì ta có thể xem G là cổng cơ bản vì nếu số lượng G cần dùng là $T(n) = O(g(n))$ thì số cổng cơ bản là $T'(n) = kO(g(n)) = O(g(n))$. Chẳng hạn, mạch FA dùng 3 cổng cố định nên RCA dùng n cổng FA có độ phức tạp là $O(n)$. Lưu ý, ta không thể xem mạch RCA là cổng cơ bản vì số cổng cơ bản trong mạch phụ thuộc vào số lượng bit đầu vào n .
- Mặc dù $T(n)$ là số phép toán cơ bản (hay số cổng cơ bản), ta vẫn có thể xem $T(n)$ như là **thời gian chạy** (running time), tức là thời gian xử lý của mạch vì nếu mỗi cổng cơ bản cần không quá k đơn vị thời gian (k cố định) thì thời gian chạy không quá $kT(n)$ nên $T(n) = O(g(n))$ thì thời gian chạy $kT(n) = O(g(n))$.

Ví dụ 2.3.2. Xét bài toán:¹⁴ cho chuỗi n bit b trong đó bit 0 chiếm một nửa (còn lại là 1), xác định một vị trí chứa 1 trong b . Thuật toán 3 giải bài toán này bằng chiến lược “tìm kiếm tuần tự”, tức là kiểm tra từng vị trí có thể. Chọn phép so sánh

¹⁴https://en.wikipedia.org/wiki/Randomized_algorithm#Motivation

Thuật toán 3 Thuật toán tìm kiếm tuần tự bit 1.

Input: $b = b_{n-1} \dots b_1 b_0$ là chuỗi n bit gồm phân nửa 0 và phân nửa 1.

Output: một vị trí i sao cho $b_i = 1$.

```

1: for  $i = 0$  to  $n - 1$  do
2:   if  $b_i = 1$  then return  $i$ 
3:   end if
4: end for

```

bit $b_i = 1$ ở Dòng 2 làm phép toán cơ bản. Trường hợp “tốt nhất”, chuỗi b có kí số thấp nhất là bit 1 ($b_0 = 1$) thì chỉ tốn 1 lần so sánh. Trường hợp “xấu nhất”, chuỗi b có nửa cao là các kí số 1 và nửa thấp là các kí số 0 thì tốn $n/2 + 1$ lần so sánh. Như vậy độ phức tạp (tính theo trường hợp xấu nhất) là

$$T(n) = \frac{n}{2} + 1 = O(n).$$

Rõ ràng hơn, ta có $T(n) = \Theta(n)$. □

Ví dụ 2.3.3. Dùng thuật toán cộng 2 số nhị phân n bit trong Ví dụ 2.2.3, ta có thể giải bài toán nhân hai số nhị phân n bit. Ý tưởng tương tự phép “nhân tay” ta đã học ở tiểu học (với số thập phân). Ví dụ, ta nhân 2 số 3 bit là $A = 011$ (giá trị 3) với $B = 101$ (giá trị 5) để được tích P gồm 6 bit như sau

$$\begin{array}{rcl}
 A & 011 & \text{(giá trị 3)} \\
 \times B & 101 & \text{(giá trị 5)} \\
 \hline
 C_0 & 000011 & \text{(giá trị 3)} \\
 + C_1 & 000000 & \text{(giá trị 0)} \\
 + C_2 & 001100 & \text{(giá trị 12)} \\
 \hline
 = P & 001111 & \text{(giá trị 15)}
 \end{array}$$

Với từng kí số thứ i của B từ thấp đến cao (trái qua phải), tính C_i : nếu $B_i = 1$ thì dịch trái A đi i bit để được C_i , ngược lại ($B_i = 0$) thì $C_i = 0$. Sau đó, cộng dồn các C_i để được kết quả là P . Dĩ nhiên, nếu $C_i = 0$ thì ta không cần cộng. Ý tưởng này được hiện thực trong Thuật toán 4. Thuật toán này tốn $O(n)$ lần cộng 2 số $O(n)$ bit mà mỗi lần cộng RCA tốn $O(n)$ thao tác trên bit nên độ phức tạp của thuật toán là $O(n^2)$, độ phức tạp bình phương (quadratic) theo n . □

Ví dụ 2.3.4. Bài toán tìm ước chung lớn nhất (greatest common divisor): cho 2 số nguyên không âm A, B , tìm $G = \gcd(A, B)$ là số nguyên lớn nhất trong số các ước của cả A lẫn B . Trong đó x là ước của y nếu y chia hết cho x , tức là y chia x dư 0. (Ta qui ước, $\gcd(0, 0) = 0$.)

Thuật toán 4 Thuật toán nhân 2 số n bit.

Input: A, B là các số nhị phân n bit.

Output: tích $P = AB$ là số nhị phân $2n$ bit.

```

1:  $P \leftarrow 0^{2n}$ 
2:  $C \leftarrow A$ 
3: for  $i = 0$  to  $n - 1$  do
4:   if  $B_i = 1$  then
5:      $P \leftarrow \text{RCA}(P, C)$ 
6:   end if
7:    $C \leftarrow \text{Shift-Left}(C)$  ▷ dịch trái  $C$  đi 1 bit
8: end for
9: return  $P$ 

```

Cách đây hơn 2000 năm đã có một thuật toán rất hiệu quả để giải bài toán tìm ước chung lớn nhất được gọi là thuật toán Euclid¹⁵ như mô tả trong Thuật toán 5.

Thuật toán 5 Thuật toán Euclid.

Input: A, B là các số nhị phân n bit.

Output: $\text{gcd}(A, B)$.

```

1: while  $B \neq 0$  do
2:    $T \leftarrow B$ 
3:    $B \leftarrow \text{MOD}(A, B)$ 
4:    $A \leftarrow T$ 
5: end while
6: return  $A$ 

```

$\text{MOD}(A, B)$ thực hiện phép chia lấy phần dư của A cho B . Bài tập 2.22 yêu cầu thiết kế thuật toán cho tính toán này với độ phức tạp là $O(n^2)$. Thuật toán Euclide có độ phức tạp $O(n^3)$, độ phức tạp lập phương (cubic) theo n . (Phân tích kĩ hơn cho thấy thuật toán Euclide có độ phức tạp là $O(n^2)$.) \square

Ta đã thấy cách đánh giá độ phức tạp giản lược với kí hiệu O -lớn. Giản lược hơn nữa, các thuật toán có độ phức tạp $T(n)$ không quá đa thức (polynomial), tức là $T(n) = O(n^k)$ với k cố định (chẳng hạn $O(n)$, $O(n^2)$, ..., $O(n^{1000})$), đều được xem là **hiệu quả** (efficient). Ngược lại, $T(n)$ “lớn hơn đa thức” (superpolynomial) được xem là **không hiệu quả** (inefficient), chẳng hạn, $T(n) = \Theta(2^n)$ là độ phức tạp mũ (exponential).

Một bài toán được gọi là “dễ” nếu có thuật toán đa thức cho nó, ngược lại, bài toán

¹⁵được đặt tên theo Euclid, một nhà toán học Hy Lạp cổ đại.

được gọi là “khó” nếu không có (hoặc không tìm được) thuật toán đa thức cho nó. Chỉ tiết hơn, ta có thể phân loại các bài toán dựa trên thuật toán tốt nhất ($T(n)$ nhỏ nhất) cho bài toán đó.

Ví dụ 2.3.5. Bài toán **phân tích số nguyên** (integer factorization): cho số nguyên $N \geq 2$, tìm các số nguyên p, q thỏa $1 < p, q < N$ và $N = pq$ hoặc thông báo “không có” nếu không có p, q nào như vậy. Ví dụ, $N = 15$ thì $p = 3, q = 5$ còn $N = 13$ thì “không có”.

Khi N nhỏ, ta (hoặc máy tính) có thể “mò” ra lời giải (chẳng hạn thử p lần lượt từ 2 đến $N - 1$ và xem thử N có chia hết cho p). Khi N lớn, việc phân tích là rất khó. Chẳng hạn, số RSA-1024 (gồm 1024 kí số nhị phân hay 309 kí số thập phân)

$N = 135066410865995223349603216278805969938881475605667027524485$
 $143851526510604859533833940287150571909441798207282164471551$
 $373680419703964191743046496589274256239341020864383202110372$
 $958725762358509643110564073501508187510676594629205563685529$
 $475213500852879416377328533906109750544334999811150056977236$
 890927563

được RSA Laboratories treo thưởng 100000\$ để phân tích trong cuộc thi RSA Factoring Challenge.¹⁶

Thuật toán (cổ điển) hiệu quả nhất đến giờ để phân tích số nguyên là thuật toán sàng trường số (number field sieve). Nếu số nguyên N gồm $n = \lfloor \log N \rfloor + 1$ bit thì độ phức tạp của thuật toán này là

$$O(2^{n^{1/3}(\log n)^{2/3}}).$$

Độ phức tạp này là trên đa thức. Cho đến nay, chưa có thuật toán cổ điển nào có thể phân tích số nguyên với độ phức tạp đa thức được công bố. Như vậy, bài toán phân tích số nguyên “được xem” là bài toán khó. \square

Ví dụ 2.3.6. Bài toán **kiểm tra số nguyên tố** (primality test): cho số nguyên $N \geq 2$, kiểm tra N có phải là số nguyên tố hay không. Số nguyên N được gọi là số nguyên tố khi và chỉ khi N không chia hết cho p nào với $1 < p < N$. Bài toán này có thể được giải bằng thuật toán AKS với độ phức tạp $O(n^6)$ nên được xem là dễ.¹⁷ \square

¹⁶https://en.wikipedia.org/wiki/RSA_Factoring_Challenge.

¹⁷Thuật toán AKS do Manindra Agrawal, Neeraj Kayal và Nitin Saxena đề xuất năm 2002.

2.4 Ngẫu nhiên, bất định và xác suất

Những thảo luận trong các phần trước là **tất định** (certain, deterministic). Tại mỗi thời điểm bất kỳ, một bit chắc chắn có giá trị 0 hoặc 1, không thể vừa 0 vừa 1 hay không 0 cũng chẳng 1. Một cổng logic luôn cho một đầu ra cố định với mỗi đầu vào cho trước, chẳng hạn nếu đầu vào của cổng AND là 00 thì đầu ra chắc chắn là 0. Các bước của một thuật toán cũng xác định và luôn cho ra cùng một kết quả với mỗi đầu vào.

Thực tế, ta thường xuyên phải đối mặt với các tình huống **bất định** (uncertain, non-deterministic). Một bit có thể không chắc là 0 hay 1. Với một đầu vào, một cổng có thể cho các đầu ra khác nhau. Các bước của một thuật toán cũng có thể được lựa chọn một cách **ngẫu nhiên** (random). Phần này trình bày cách mở rộng tính toán tất định cho trường hợp bất định bằng lý thuyết xác suất mà ta vẫn gọi chung là **tính toán cổ điển** (classical computation) khi so với tính toán lượng tử.

2.4.1 Xác suất và thuật toán ngẫu nhiên

Lý thuyết xác suất (probability theory) là ngành Toán học giúp định lượng, tính toán và suy diễn trên các hiện tượng không chắc chắn. Phụ lục B trình bày tóm lược các kiến thức xác suất cần thiết cho việc hiểu và thực hành tính toán lượng tử. Độc giả chưa có kiến thức cần đọc qua và tham khảo lại khi cần.

Thuật toán ngẫu nhiên (randomized algorithm, probabilistic algorithm) là thuật toán có dùng các lựa chọn ngẫu nhiên trong quá trình thực thi. Nói cách khác, đó là thuật toán có “dùng đồng xu” khi tính toán. Các thuật toán ngẫu nhiên thường đơn giản và hiệu quả hơn các **thuật toán tất định** (deterministic algorithm) thông thường. Để có được điều này, các thuật toán ngẫu nhiên thường cho kết quả không chắc chắn đúng, tuy nhiên, xác suất lỗi có thể được làm cho rất nhỏ!

Ví dụ 2.4.1. Trong Ví dụ 2.3.2, ta thấy Thuật toán 3 là một thuật toán tất định để giải bài toán “xác định một vị trí chứa bit 1 trong chuỗi b bit gồm phân nửa các bit 0 và phân nửa các bit 1”. Ta cũng thấy rằng số phép so sánh bit cần dùng (trong trường hợp xấu nhất) của Thuật toán này là $\Theta(n)$.

Thuật toán 6 là một thuật toán ngẫu nhiên để giải cùng bài toán. Thuật toán này thường được nói là thuộc thể loại **Las Vegas** (Las Vegas algorithm) vì chắc chắn cho kết quả đúng. Gọi T là số phép so sánh bit cần dùng (cũng là số lần lặp của vòng lặp 1-5) thì T có kì vọng là $E(T) = 2$ (Bài tập 2.24). Như vậy, trong trường hợp trung bình, Thuật toán 6 có thời gian chạy là $\Theta(1)$, hiệu quả hơn so với thời gian chạy của Thuật toán 3 là $\Theta(n)$ (Bài tập 2.23). Lưu ý, mặc dù số lần lặp không bị chặn nhưng chắc chắn thuật toán sẽ dừng!

Thuật toán 6 Một thuật toán Las Vegas tìm kiếm bit 1.

Input: $b = b_{n-1} \dots b_1 b_0$ là chuỗi n bit gồm phân nửa 0 và phân nửa 1.

Output: một vị trí i sao cho $b_i = 1$.

```

1: loop
2:    $i \leftarrow \text{RANDINT}(1, n)$    ▶ chọn ngẫu nhiên một số nguyên từ tập  $\{1, \dots, n\}$ 
3:   if  $b_i = 1$  then return  $i$ 
4:   end if
5: end loop

```

Thuật toán 7 là một thuật toán ngẫu nhiên khác để giải cùng bài toán. Thuật toán này thường được nói là thuộc thể loại **Monte Carlo** (Monte Carlo algorithm) vì có thể cho kết quả sai. Chọn k cố định thì Thuật toán 7 có thời gian chạy (trường hợp xấu nhất) là $\Theta(1)$, hiệu quả hơn so với thời gian chạy của Thuật toán 3 là $\Theta(n)$. Xác suất lỗi của Thuật toán 7 là (Bài tập 2.25)

$$P(\text{fail}) = \left(\frac{1}{2}\right)^k,$$

rất nhỏ nếu k vừa phải. Chẳng hạn, chọn $k = 100$ thì xác suất lỗi là $2^{-100} < 10^{-30}$ (nhỏ hơn xác suất máy tính “bỗng dưng bị lỗi”!). \square

Thuật toán 7 Một thuật toán Monte Carlo tìm kiếm bit 1.

Input: $k \in \mathbb{N}_{>0}$ và $b = b_{n-1} \dots b_1 b_0$ là chuỗi n bit gồm phân nửa 0 và phân nửa 1.

Output: một vị trí i sao cho $b_i = 1$.

```

1: for  $j = 1$  to  $k$  do
2:    $i \leftarrow \text{RANDINT}(1, n)$ 
3:   if  $b_i = 1$  then return  $i$ 
4:   end if
5: end for
6: return fail

```

Ví dụ 2.4.2. Xét bài toán “kiểm tra phép nhân ma trận”: cho 3 ma trận nhị phân $A, B, C \in \mathbb{B}^{n \times n}$, kiểm tra $AB \stackrel{?}{=} C$. Ma trận nhị phân là ma trận có các phần tử là 0 hoặc 1. Các phép toán trên ma trận nhị phân là modulo 2, nghĩa là kết quả các phép toán luôn được chia lấy dư cho 2. Ví dụ

$$\begin{aligned}
\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} &= \begin{bmatrix} 1 \times 1 + 1 \times 1 \pmod{2} & 1 \times 0 + 1 \times 1 \pmod{2} \\ 1 \times 1 + 0 \times 1 \pmod{2} & 1 \times 0 + 0 \times 1 \pmod{2} \end{bmatrix} \\
&= \begin{bmatrix} 2 \pmod{2} & 1 \pmod{2} \\ 1 \pmod{2} & 0 \pmod{2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.
\end{aligned}$$

Thuật toán thông thường để giải bài toán trên là nhân 2 ma trận A, B rồi so sánh bằng với ma trận C . Tuy nhiên, phép nhân ma trận thông thường có độ phức tạp là $\Theta(n^3)$. Dùng các thuật toán nhân ma trận tinh vi (và phức tạp) hơn, có thể giảm độ phức tạp xuống khoảng $\Theta(n^{2.37})$.¹⁸

Nhận xét, nếu có vector nhị phân $r \in \mathbb{B}^n$ sao cho $ABr \neq Cr$ thì $AB \neq C$. Hơn nữa, phép nhân ma trận với vector có độ phức tạp chỉ là $\Theta(n^2)$. Lưu ý, nếu $ABr = Cr$ thì không bảo đảm $AB = C$. Từ đó, ta có Thuật toán 8 là một thuật toán Monte Carlo giải bài toán trên. Chọn k cố định thì Thuật toán 8 có thời gian chạy là $\Theta(n^2)$, hiệu quả hơn thuật toán thông thường. Xác suất lỗi của Thuật toán 8 là (Bài tập 2.26)

$$P(\text{lỗi}) \leq \left(\frac{1}{2}\right)^k,$$

rất nhỏ nếu k vừa phải. □

Thuật toán 8 Một thuật toán Monte Carlo kiểm tra phép nhân ma trận

Input: $k \in \mathbb{N}_{>0}$ và $A, B, C \in \mathbb{B}^{n \times n}$.

Output: $AB \stackrel{?}{=} C$.

```

1: for  $j = 1$  to  $k$  do
2:    $r \leftarrow \text{RAND}(\{0, 1\}, n)$  ▷ chọn ngẫu nhiên 1 vector từ tập  $\mathbb{B}^n$ 
3:   if  $A(Br) \neq Cr$  then return false
4:   end if
5: end for
6: return true

```

Các nội dung còn lại của phần này trình bày cách mở rộng tính toán tất định cho trường hợp bất định bằng các vector, ma trận xác suất và kí pháp Dirac để dễ dàng chuyển qua tính toán lượng tử.¹⁹ Các khái niệm và phép toán trên các vector, ma trận xác suất tương tự như trên các vector, ma trận phức ở Chương 1 với tập số thực \mathbb{R} thay cho tập số phức \mathbb{C} .

2.4.2 Trạng thái xác suất

Cho biến ngẫu nhiên X có tập giá trị S gồm n phần tử, để thuận tiện tính toán ta xem như²⁰

$$S = \{0, 1, \dots, n-1\}.$$

¹⁸https://en.wikipedia.org/wiki/Matrix_multiplication#Computational_complexity.

¹⁹cách trình bày này không thường thấy trong các tài liệu lý thuyết xác suất thông thường.

²⁰có thể nói, ta mã hóa n phần tử của S bằng các số nguyên không âm $0, 1, \dots, n-1$.

Hàm xác suất (hay phân phối) của X có thể được mô tả bằng **vector xác suất** (probability vector)

$$|X\rangle = \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_{n-1} \end{bmatrix} \in \mathbb{R}^n, \quad p_k = P(X = k), k = 0, \dots, n-1.$$

Nhận xét (do yêu cầu của các tiên đề xác suất)

1. $p_k \geq 0, \forall k = 0, \dots, n-1$,
2. $\sum_{k=0}^{n-1} p_k = 1$.

Ta đồng nhất tập giá trị $S = \{0, \dots, n-1\}$ với cơ sở chuẩn tắc của \mathbb{R}^n , tức là đồng nhất các giá trị $0, 1, \dots, n-1$ với lần lượt các vector

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \quad \dots, \quad |n-1\rangle = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}.$$

Khi đó, $|X\rangle$ có thể được viết là

$$|X\rangle = \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_{n-1} \end{bmatrix} = p_0|0\rangle + p_1|1\rangle + \dots + p_{n-1}|n-1\rangle = \sum_{k=0}^{n-1} p_k|k\rangle.$$

Các giá trị $k \equiv |k\rangle$ ($k = 0, \dots, n-1$) được gọi là các **trạng thái cổ điển** (classical state).

Như vậy, cho tập các trạng thái cổ điển S , một **trạng thái xác suất** (probabilistic state) là một phân phối trên S được xác định bằng một vector xác suất. Đặc biệt, các trạng thái tương ứng với các vector cơ sở chuẩn tắc là tất định vì

$$|X\rangle = |k\rangle \iff P(X = k) = 1,$$

nghĩa là “ X chắc chắn nhận giá trị k ”.

Ta nói việc tiến hành thí nghiệm ngẫu nhiên tương ứng với biến ngẫu nhiên X là việc **đo** (measure) X . Như vậy, sau khi đo (tức là biết kết quả của thí nghiệm), dù $|X\rangle = \sum_{k=0}^{n-1} p_k|k\rangle$ có phải là trạng thái tất định hay không thì X cũng chắc chắn nhận một giá trị k nào đó thuộc S (tức là một trạng thái cổ điển) với

$$P(X = k) = p_k.$$

Ví dụ 2.4.3. Xét các biến ngẫu nhiên trên tập bit $\mathbb{B} = \{0, 1\}$, các giá trị

$$0 \equiv |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, 1 \equiv |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

là các trạng thái cổ điển.

Gọi X là mặt ra khi tung một đồng xu đồng chất thì X có thể được mô tả bằng vector xác suất

$$|\frac{1}{2}\rangle = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle$$

mà sau khi đo (tung xong) thì X chỉ có thể hoặc là 0 (sấp) hoặc là 1 (ngửa). Xác suất để X là 0 hay là 1 đều là 50%.

Lưu ý, kí hiệu trong ket $|\cdot\rangle$ chỉ là nhãn. Chẳng hạn, $|\frac{1}{2}\rangle$ là trạng thái xác suất “được đặt tên” là $\frac{1}{2}$ để “gợi nhớ” đây là trạng thái “tổ hợp đều” hay “5 ăn 5 thua”. \square

Ví dụ 2.4.4. Với tập chuỗi 2 bit $\mathbb{B}^2 = \{00, 01, 10, 11\}$, các giá trị

$$00 \equiv |0\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, 01 \equiv |1\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, 10 \equiv |2\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, 11 \equiv |3\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

là các trạng thái cổ điển.

Gọi X là mặt ra khi tung 2 đồng xu đồng chất. Nếu 2 đồng xu độc lập thì phân phối của X có thể được mô tả bằng vector xác suất

$$|\frac{1}{4}\rangle = \frac{1}{4} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \frac{1}{4}|0\rangle + \frac{1}{4}|1\rangle + \frac{1}{4}|2\rangle + \frac{1}{4}|3\rangle$$

mà sau khi đo (tung xong) thì X chỉ có thể hoặc là $0 \equiv 00$ (sấp, sấp) hoặc là $1 \equiv 01$ (sấp, ngửa) hoặc là $2 \equiv 10$ (ngửa, sấp) hoặc là $3 \equiv 11$ (ngửa, ngửa). Xác suất để X là 0, 1, 2 hay 3 đều là 25%.

Nếu 2 đồng xu là 2 thỏi nam châm có 2 mặt là 2 cực từ cùng dấu (đẩy nhau) thì phân phối của X có thể được mô tả bằng vector xác suất

$$\frac{1}{2} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \frac{1}{2}|1\rangle + \frac{1}{2}|2\rangle = \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle.$$

Nếu 2 mặt là 2 cực từ trái dấu (hút nhau) thì phân phối của X có thể được mô tả bằng vector xác suất

$$\frac{1}{2} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{2}|0\rangle + \frac{1}{2}|3\rangle = \frac{1}{2}|00\rangle + \frac{1}{2}|11\rangle.$$

□

2.4.3 Thao tác xác suất

Hệ thống động (dynamical system) là hệ thống có trạng thái thay đổi theo thời gian.

Xét hệ thống động có trạng thái khi được quan sát (khi đo) là một giá trị trong tập hữu hạn $S = \{0, 1, \dots, n-1\}$. Khi không được quan sát, trạng thái của hệ tại thời điểm t có thể được mô tả bằng biến ngẫu nhiên X_t có tập giá trị là S . $|X_t\rangle$ cũng được gọi là trạng thái của hệ tại thời điểm t .

Nếu trạng thái $|Y\rangle$ của hệ ngay sau khi chịu tác động T được xác định hoàn toàn khi biết trạng thái $|X\rangle$ ngay trước khi thực hiện T (mà không cần biết các trạng thái trước đó nữa) thì ta gọi T là một **thao tác** (operation) và kí hiệu

$$|Y\rangle = T|X\rangle.$$

Cho T là một thao tác trên hệ, đặt $|a_k\rangle = T|k\rangle$ ($k = 0, \dots, n-1$) là kết quả của T trên các trạng thái cổ điển. Đặt A là ma trận gồm các cột là $|a_k\rangle$, tức là

$$A = \begin{bmatrix} |a_0\rangle & |a_1\rangle & \dots & |a_{n-1}\rangle \end{bmatrix} \in \mathbb{R}^{n \times n}.$$

A được gọi là ma trận biểu diễn của T . Nhận xét (do yêu cầu của các tiên đề xác suất)

1. Các phần tử của A là các số thực không âm,
2. Tổng các phần tử trên mỗi cột của A đều là 1.

Ma trận dạng này được gọi là **ma trận xác suất** (probability matrix).

Đặc biệt, nếu A chỉ gồm các số 0 hoặc 1 (A là ma trận nhị phân) thì T được gọi là **thao tác tất định** (deterministic operation) để phân biệt với **thao tác xác suất** (probabilistic operation) nói chung. Hơn nữa, nếu ma trận nhị phân A khả nghịch thì A là **ma trận hoán vị** (permutation matrix) với mỗi dòng và mỗi cột đều chỉ

có một số 1. Khi đó, A biểu diễn cho thao tác tắt định khả nghịch với A^{-1} , cũng là ma trận hoán vị, biểu diễn cho thao tác tắt định ngược với T (“undo” T).

Cho thao tác T có ma trận biểu diễn là A , nếu $|Y\rangle = T|X\rangle$ và $|X\rangle = \sum_{k=0}^{n-1} p_k |k\rangle$ thì theo công thức xác suất toàn phần ta có

$$|Y\rangle = \sum_{k=0}^{n-1} P(X=k)P(Y|X=k) = \sum_{k=0}^{n-1} p_k T|k\rangle = A|X\rangle.$$

Do đó A, T thường được dùng lẫn lộn. Hơn nữa, do tính chất tuyến tính của phép nhân ma trận nên T còn được gọi là toán tử (tuyến tính).

Cho các thao tác T_1, T_2 (có ma trận biểu diễn là T_1, T_2), nếu T_1 biến đổi trạng thái của hệ từ $|X\rangle$ thành $|Y\rangle$ rồi T_2 biến đổi từ $|Y\rangle$ thành $|Z\rangle$ thì

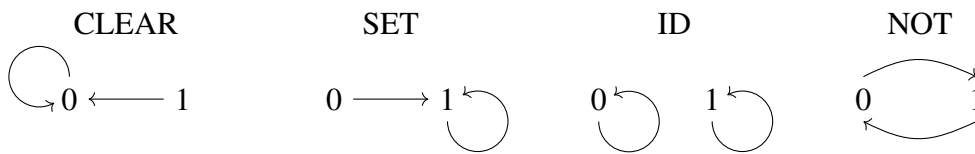
$$|Z\rangle = T_2|Y\rangle = T_2(T_1|X\rangle) = (T_2T_1)|X\rangle$$

nên ma trận biểu diễn của thao tác “ T_1 rồi T_2 ” là T_2T_1 .

Ví dụ 2.4.5. (tiếp Ví dụ 2.4.3) Chỉ có 4 thao tác tắt định sau trên tập bit \mathbb{B} (\clubsuit)

$$\text{CLEAR} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}, \quad \text{SET} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}, \quad \text{ID} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{NOT} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

CLEAR là xóa bit ($0 \rightarrow 0, 1 \rightarrow 0$), SET là đặt bit ($0 \rightarrow 1, 1 \rightarrow 1$), ID = I là “NOOP” ($0 \rightarrow 0, 1 \rightarrow 1$), NOT là lật bit ($0 \rightarrow 1, 1 \rightarrow 0$). Các thao tác này cũng có thể được mô tả bằng **sơ đồ chuyển trạng thái** (state transition diagram) sau



Trong sơ đồ này, các nút mô tả các trạng thái cổ điển (0, 1) còn các cạnh có hướng (mũi tên) mô tả việc chuyển trạng thái từ nút đầu đến nút đích của cạnh. Chẳng hạn với CLEAR, bắt đầu từ nút 0 hay nút 1 đều chuyển đến nút 0. Vì các thao tác này đều tắt định nên từ mỗi trạng thái đầu chỉ có duy nhất một trạng thái đích, tức là mỗi nút chỉ có một mũi tên đi ra.

Tác động của NOT lên trạng thái xác suất $|X\rangle = \begin{bmatrix} p_0 & p_1 \end{bmatrix}^T$ là

$$\text{NOT}|X\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \end{bmatrix} = \begin{bmatrix} p_1 \\ p_0 \end{bmatrix}$$

Ta cũng có thể tính

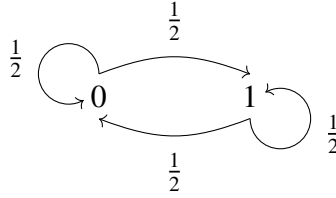
$$\begin{aligned}\text{NOT}|X\rangle &= \text{NOT}(p_0|0\rangle + p_1|1\rangle) = p_0\text{NOT}|0\rangle + p_1\text{NOT}|1\rangle \\ &= p_0|1\rangle + p_1|0\rangle = p_1|0\rangle + p_0|1\rangle.\end{aligned}$$

Có thể nói NOT là phiên bản mở rộng của cổng logic NOT cho tính toán bất định mặc dù bản thân NOT là một thao tác tất định.

Thao tác tung đồng xu đồng chất có thể được mô tả bằng ma trận

$$\text{TOSS} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

Thao tác này cũng có thể được mô tả bằng sơ đồ chuyển trạng thái sau



Khác với sơ đồ của thao tác tất định, trong sơ đồ này, từ một nút có thể có nhiều mũi tên đi ra. Chẳng hạn từ nút 0 có thể đến nút 0 hoặc nút 1 với xác suất đều là 50%. Các cạnh trong thao tác tất định có thể hiểu là chuyển với xác suất 100%.

Tác động của TOSS lên trạng thái xác suất $|X\rangle = \begin{bmatrix} p_0 & p_1 \end{bmatrix}^T$ là

$$\text{TOSS}|X\rangle = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} p_0 + p_1 \\ p_0 + p_1 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix} = |\frac{1}{2}\rangle.$$

Như vậy, bất kể trạng thái ban đầu của đồng xu là gì, khi tung ta đều được trạng thái “tổ hợp đều” mà khi đo (tung xong) thì được 0 (sấp) hoặc 1 (ngửa) với xác suất đều là 50%.

Từ ý nghĩa của các thao tác CLEAR và TOSS, ta đoán rằng “CLEAR rồi TOSS” cũng là TOSS, thật vậy

$$\text{TOSS} \times \text{CLEAR} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

“TOSS rồi CLEAR” là CLEAR

$$\text{CLEAR} \times \text{TOSS} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2 & 2 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}.$$

Tương tự ta có “NOT rồi TOSS” và “TOSS rồi NOT” đều là TOSS; thật ra với mọi T , thao tác “ T rồi TOSS” đều là TOSS. (♣) □

Ví dụ 2.4.6. (tiếp Ví dụ 2.4.4) Có $(2^2)^2 = 256$ thao tác tất định trên tập chuỗi 2 bit $\mathbb{B}^2 = \{00, 01, 10, 11\}$ (♣). Thao tác “nổi tiếng nhất” là

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

CNOT (Control-NOT) lật bit sau nếu bit đầu là 1

$$00 \rightarrow 00, 01 \rightarrow 01, 10 \rightarrow 11, 11 \rightarrow 10.$$

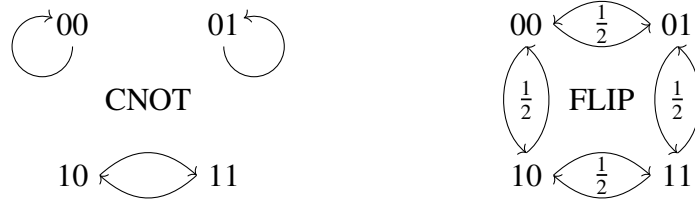
Có thể nói CNOT là phiên bản mở rộng của cổng logic XOR vì

$$\text{CNOT}|a, b\rangle = |a, a \oplus b\rangle, \quad a, b \in \{0, 1\}.$$

Ta thấy CNOT là ma trận hoán vị nên CNOT mô tả cho phép toán 2 bit khả nghịch, chính là cổng CNOT ở Ví dụ 2.2.4 mà nghịch đảo là chính nó (♣)

$$\text{CNOT}^{-1} = \text{CNOT}.$$

CNOT cũng có thể được mô tả bằng sơ đồ bên trái của hình sau



CNOT là tất định trong khi thao tác ứng với sơ đồ bên phải, gọi là FLIP, không tất định. Tưởng tượng ta có chuỗi 2 bit (hoặc 2 đồng xu), chọn ngẫu nhiên 1 trong 2 và lật bit. Từ 00 nếu bit đầu bị lật (xác suất 50%) thì thành 10 còn nếu bit sau bị lật (xác suất 50%) thì thành 01. Thao tác này có ma trận biểu diễn là

$$\text{FLIP} = \frac{1}{2} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}.$$

Tác động của FLIP lên trạng thái $|00\rangle$ (ứng với vector trạng thái $[1 \ 0 \ 0 \ 0]^T$) hay $|11\rangle$ (ứng với vector trạng thái $[0 \ 0 \ 0 \ 1]^T$) đều thành trạng thái 50% $|01\rangle$ và 50% $|10\rangle$ (ứng với vector trạng thái $[0 \ \frac{1}{2} \ \frac{1}{2} \ 0]^T$) mà nếu tiếp tục tác động FLIP thì thành 50% $|00\rangle$ và 50% $|11\rangle$ (ứng với vector trạng thái $[\frac{1}{2} \ 0 \ 0 \ \frac{1}{2}]^T$). \square

2.4.4 Hệ nhiều thành phần

Việc “ghép” nhiều hệ thống nhỏ đơn giản để tạo nên một hệ thống lớn phức tạp hơn thường được dùng trong biểu diễn và xử lý thông tin, chẳng hạn như việc ghép nhiều bit.

Cho A, B là các hệ có các tập trạng thái cổ điển tương ứng là

$$S_A = \{0, \dots, m-1\}, \quad S_B = \{0, \dots, n-1\}.$$

Khi xem xét “đồng thời” A, B , ta có **hệ thống ghép** (compound system) $C = A \times B$ với tập trạng thái cổ điển

$$S_C = S_A \times S_B = \{(a, b) : a \in S_A, b \in S_B\}.$$

Ta đã đồng nhất tập giá trị $S_A = \{0, \dots, m-1\}$ với cơ sở chuẩn tắc $\{|0\rangle, \dots, |m-1\rangle\}$ của \mathbb{R}^m và tập giá trị $S_B = \{0, \dots, n-1\}$ với cơ sở chuẩn tắc $\{|0\rangle, \dots, |n-1\rangle\}$ của \mathbb{R}^n . Dùng tích tensor ta đồng nhất các $(a, b) \in S_C$ với

$$|a\rangle \otimes |b\rangle = |a\rangle|b\rangle = |a, b\rangle = |ab\rangle$$

là các vector trong cơ sở chuẩn tắc của \mathbb{R}^{mn} .

Khi không được quan sát, trạng thái của các hệ A, B có thể được mô tả bằng các biến ngẫu nhiên X, Y với tập giá trị tương ứng là S_A, S_B . Do đó, trạng thái của hệ ghép $C = A \times B$ có thể được mô tả bằng phân phối đồng thời của X, Y mà có thể được biểu diễn bằng vector xác suất

$$|Z\rangle = \sum_{(a,b) \in S_A \times S_B} p_{ab} |ab\rangle = \sum_{a=0}^{m-1} \sum_{b=0}^{n-1} p_{ab} |ab\rangle$$

trong đó

$$p_{ab} = P((X, Y) = (a, b)) = P(X = a, Y = b).$$

Nếu X, Y độc lập thì

$$P(X = a, Y = b) = P(X = a)P(Y = b), \forall a \in S_A, b \in S_B$$

nên

$$\begin{aligned} |Z\rangle &= \sum_{a=0}^{m-1} \sum_{b=0}^{n-1} p_{ab} |ab\rangle = \sum_{a=0}^{m-1} \sum_{b=0}^{n-1} P(X = a) |a\rangle P(Y = b) |b\rangle \\ &= \left(\sum_{a=0}^{m-1} P(X = a) |a\rangle \right) \otimes \left(\sum_{b=0}^{n-1} P(Y = b) |b\rangle \right) \\ &= |X\rangle \otimes |Y\rangle = |X\rangle|Y\rangle. \end{aligned}$$

Do đó $|Z\rangle$ tách được. Ngược lại, nếu X, Y không độc lập thì $|Z\rangle$ không tách được.

Thao tác T trên hệ ghép $C = A \times B$ có ma trận biểu diễn gồm các cột là

$$T|ab\rangle, \quad a = 0, \dots, m-1, b = 0, \dots, n-1$$

là kết quả của T trên các trạng thái cổ điển. Lưu ý, $T \in \mathbb{R}^{mn \times mn}$.

Việc thực hiện “đồng thời” thao tác T_A, T_B một cách “độc lập” (hay “riêng rẽ”) trên A, B có thể được xem như thực hiện thao tác

$$T = T_A \otimes T_B$$

trên hệ ghép C . Khi đó ta cũng nói T tách được, ngược lại thì T không tách được.

Ví dụ 2.4.7. (tiếp Ví dụ 2.4.4, 2.4.6) Tập chuỗi 2 bit $\mathbb{B}^2 = \{00, 01, 10, 11\}$ có thể được xem là ghép 2 bit $\mathbb{B} = \{0, 1\}$, ta có

$$00 \equiv |0_2\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = |0\rangle \otimes |0\rangle = |0\rangle|0\rangle = |00\rangle.$$

Tương tự, $01 \equiv |1_2\rangle = |01\rangle, 10 \equiv |2_2\rangle = |10\rangle, 11 \equiv |3_2\rangle = |11\rangle$.²¹ Các trạng thái cổ điển này là tách được.

Trạng thái xác suất khi tung 2 đồng xu đồng chất độc lập là tách được

$$|\frac{1}{4}\rangle = \frac{1}{4} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes \frac{1}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = |\frac{1}{2}\rangle|\frac{1}{2}\rangle.$$

Ngược lại, các trạng thái xác suất sau không tách được (♣)

$$\frac{1}{2} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle, \quad \frac{1}{2} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{2}|00\rangle + \frac{1}{2}|11\rangle.$$

Thao tác tung 2 đồng xu đồng chất độc lập có ma trận biểu diễn

$$\text{TOSS}_2 = \text{TOSS} \otimes \text{TOSS} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \otimes \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

²¹kí hiệu $|3_2\rangle$ chỉ vector cơ sở chuẩn tắc thứ 3 (tính từ 0) của \mathbb{R}^{2^2} là không gian vector ứng với việc ghép 2 bit.

Do đó TOSS_2 là tách được. Ngược lại, CNOT không tách được (\clubsuit) vì là thao tác “đánh lúu” trên cả 2 bit. FLIP có tách được không? (\clubsuit) \square

2.5 Dẫn nhập tính toán lượng tử

Tính toán tất định và bất định với lý thuyết xác suất được gọi chung là **tính toán cổ điển** (classical computation) khi so với **tính toán lượng tử** (quantum computation). Nói một cách đơn giản, việc chuyển từ tính toán cổ điển sang tính toán lượng tử chính là chuyển từ vector và ma trận xác suất sang vector và ma trận phức mà nói nôm na là chuyển từ số thực \mathbb{R} sang số phức \mathbb{C} .

Chi tiết hơn, Bảng 2.12 so sánh trạng thái xác suất với trạng thái lượng tử, Bảng 2.13 so sánh thao tác xác suất với thao tác lượng tử và Bảng 2.14 so sánh hệ nhiều thành phần xác suất với lượng tử.

Tính toán cổ điển	Tính toán lượng tử
Tập các trạng thái cơ bản (còn gọi là các trạng thái cổ điển) $S = \{0, 1, \dots, n-1\} \equiv \{ 0\rangle, 1\rangle, \dots, n-1\rangle\}$	
Trạng thái xác suất $ X\rangle = \begin{bmatrix} p_0 \\ \vdots \\ p_{n-1} \end{bmatrix} = \sum_{k=0}^{n-1} p_k k\rangle \in \mathbb{R}^n$ p_k được gọi là xác suất	Trạng thái lượng tử $ \psi\rangle = \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_{n-1} \end{bmatrix} = \sum_{k=0}^{n-1} \alpha_k k\rangle \in \mathbb{C}^n$ α_k được gọi là amplitude
Yêu cầu cho vector trạng thái	
Vector xác suất $p_k \geq 0$ và $\sum_{k=0}^{n-1} p_k = 1$	Vector đơn vị $\sum_{k=0}^{n-1} \alpha_k ^2 = 1$
Khi đo sẽ được trạng thái cơ bản k với xác suất	
p_k	$ \alpha_k ^2$
Hệ 2 mức với tập trạng thái cơ bản là tập bit $S = \mathbb{B} = \{0, 1\}$ bit xác suất	bit lượng tử (qubit)

Bảng 2.12: Trạng thái xác suất với trạng thái lượng tử.

Ví dụ 2.5.1. (tiếp Ví dụ 2.4.5) Trên tập bit $\mathbb{B} = \{0, 1\}$ chỉ có một trạng thái xác suất “tổ hợp đều” là

$$|\tfrac{1}{2}\rangle = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Tính toán cổ điển	Tính toán lượng tử
Thao tác là toán tử tuyến tính $T\left(\sum_{k=0}^{n-1} x_k k\rangle\right) = \sum_{k=0}^{n-1} x_k T k\rangle$ $T = \begin{bmatrix} c_0\rangle & \dots & c_{n-1}\rangle \end{bmatrix}, \quad c_k\rangle = T k\rangle, k = 0, \dots, n-1$	
Thao tác xác suất $ Y\rangle = T X\rangle$ là vector xác suất $T \in \mathbb{R}^{n \times n}$ là ma trận xác suất T không nhất thiết khả nghịch $\{ c_0\rangle, \dots, c_{n-1}\rangle\}$ là các vector xác suất	Thao tác lượng tử $ \phi\rangle = T \psi\rangle$ là vector đơn vị $T \in \mathbb{C}^{n \times n}$ là ma trận unita T phải khả nghịch một cơ sở trực chuẩn của \mathbb{C}^n
Nếu $n = 2^N$ thì gọi là các cổng N -bit cổng logic	cổng lượng tử
Ma trận biểu diễn của thao tác “ T_1 rồi T_2 ” là $T_2 T_1$	

Bảng 2.13: Thao tác xác suất với thao tác lượng tử.

Tính toán cổ điển	Tính toán lượng tử
Hệ ghép $C = A \times B$ được mô tả bởi tích tensor	
Trạng thái $ Z\rangle = \sum_{a=0}^{m-1} \sum_{b=0}^{n-1} p_{ab} ab\rangle \in \mathbb{R}^{mn}$	
$ X\rangle = \sum_{a=0}^{m-1} \sum_{b=0}^{n-1} \alpha_{ab} ab\rangle \in \mathbb{C}^{mn}$ tách được $ Z\rangle = X\rangle \otimes Y\rangle$ khi X, Y độc lập	
$ X\rangle = \psi\rangle \otimes \phi\rangle$ khi $ X\rangle$ không rời Ma trận của thao tác T trên hệ ghép $C = A \times B$ có các cột là $T ab\rangle$ $T \in \mathbb{R}^{mn \times mn}$	
$T \in \mathbb{C}^{mn \times mn}$ $T = T_A \otimes T_B$ tách được nếu T được thực hiện “riêng rẽ” T_A trên A và T_B trên B	
Ghép nhiều hệ 2 mức	
Chuỗi bit	Thanh ghi lượng tử

Bảng 2.14: Hệ nhiều thành phần xác suất với lượng tử.

nhưng có nhiều trạng thái lượng tử “tổ hợp đều”, chẳng hạn

$$|+\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}, \quad |-\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}, \quad |i\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ i \end{bmatrix}, \quad |-i\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -i \end{bmatrix}.$$

Khi đo các trạng thái này đều được 0, 1 với xác suất như nhau (50%).

Cũng vậy, trên tập bit \mathbb{B} chỉ có một thao tác xác suất là

$$\text{TOSS} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

biến cả $|0\rangle$ lẫn $|1\rangle$ thành $|\frac{1}{2}\rangle$. Ngược lại, có nhiều thao tác lượng tử biến cả $|0\rangle$ lẫn $|1\rangle$ thành các trạng thái “tổ hợp đều” như

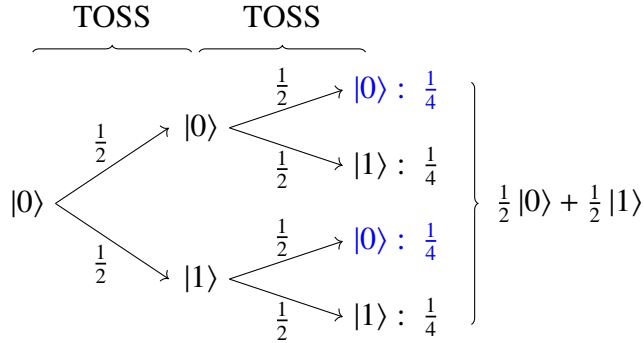
$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad B = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix}, \quad SH = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix}.$$

Ta thấy

- H biến $|0\rangle$ thành $|+\rangle$ và $|1\rangle$ thành $|-\rangle$,
- B biến $|0\rangle$ thành $|i\rangle$ và $|1\rangle$ thành $-i|i\rangle$,
- SH biến $|0\rangle$ thành $|i\rangle$ và $|1\rangle$ thành $-i|i\rangle$.

Như vậy, “nếu chỉ được thực hiện 1 lần thì các thao tác H , B hay SH đều giống thao tác tung đồng xu cổ điển”.

Ta đã biết thao tác “TOSS rồi TOSS” cũng là TOSS, tức là $\text{TOSS}^2 = \text{TOSS}$. Như vậy, “tung đồng xu cổ điển 2 lần thì cũng như tung 1 lần.” Từ trạng thái $|0\rangle$ (0, ngửa), sơ đồ sau cho thấy “hành vi” của “TOSS rồi TOSS”: có 2 “con đường” biến $|0\rangle$ thành $|0\rangle$, mỗi con đường có xác suất là $\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$ nên tổng xác suất là $\frac{1}{2}$. Tương tự, có 2 con đường biến $|0\rangle$ thành $|1\rangle$ với tổng xác suất cũng là $\frac{1}{2}$.



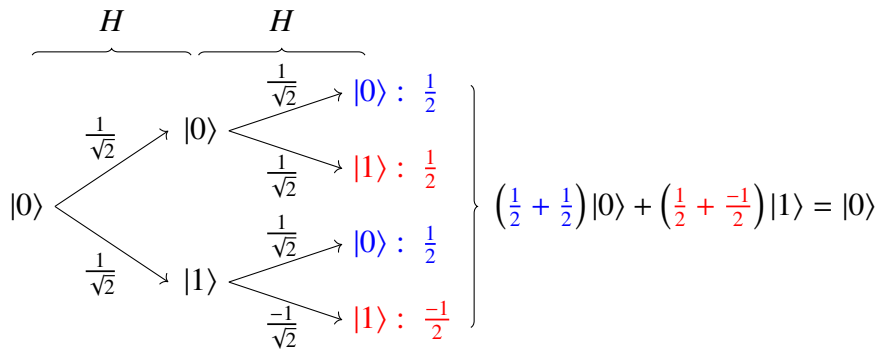
Tuy nhiên

$$\begin{aligned}
 H^2 &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \\
 B^2 &= \frac{i}{\sqrt{2}} \begin{bmatrix} 0 & i \\ i & 0 \end{bmatrix}, \\
 SH^2 &= \frac{1}{2} \begin{bmatrix} 1+i & 1-i \\ 1+i & i-1 \end{bmatrix}.
 \end{aligned}$$

Ta thấy

- “tung đồng xu lượng tử 2 lần theo kiểu H thì được lại mặt cũ!”
- “tung đồng xu lượng tử 2 lần theo kiểu B thì được mặt ngược mặt cũ!”
- “tung đồng xu lượng tử 2 lần theo kiểu SH thì giống tung đồng xu cổ điển!”

Chẳng hạn, từ trạng thái $|0\rangle$ sơ đồ sau cho thấy “hành vi” của “ H rồi H ”: có 2 con đường biến $|0\rangle$ thành $|0\rangle$, mỗi con đường có amplitude là $\frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}} = \frac{1}{2}$ nên tổng amplitude là $\frac{1}{2}$. Tuy nhiên, cũng có 2 con đường biến $|0\rangle$ thành $|1\rangle$ nhưng có amplitude là $\frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}} = \frac{1}{2}$ và $\frac{1}{\sqrt{2}} \frac{-1}{\sqrt{2}} = \frac{-1}{2}$ nên tổng amplitude là 0. Có thể nói 2 con đường của $|0\rangle$ là “tăng cường nhau” trong khi 2 con đường của $|1\rangle$ là “đập tắt nhau”. Hiện tượng này thường được gọi là **giao thoa** (interference) và là một đặc trưng của tính toán lượng tử mà tính toán xác suất cổ điển không có (vì các xác suất không âm nên khi gộp lại chỉ có tăng cường).



Với FLIP, từ $|00\rangle$ hay $|11\rangle$ ta đều được 50% $|01\rangle$ và 50% $|10\rangle$ mà nếu tiếp tục tác động FLIP thì được 50% $|00\rangle$ và 50% $|11\rangle$. FLIP không nhớ bắt đầu từ $|00\rangle$ hay $|11\rangle$. Ngược lại, xét thao tác lượng tử sau

$$QFLIP = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & -1 \\ 1 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 \end{bmatrix}.$$

Từ $|00\rangle$ áp dụng QFLIP được $[0 \ \frac{1}{\sqrt{2}} \ \frac{1}{\sqrt{2}} \ 0]^T$ cũng là 50% $|01\rangle$ và 50% $|10\rangle$ nhưng nếu tiếp tục áp dụng QFLIP thì được lại $|00\rangle$. Từ $|11\rangle$ áp dụng QFLIP được $[0 \ \frac{-1}{\sqrt{2}} \ \frac{1}{\sqrt{2}} \ 0]^T$ cũng là 50% $|01\rangle$ và 50% $|10\rangle$ nhưng nếu tiếp tục áp dụng QFLIP thì được lại $|11\rangle$. Như vậy QFLIP “nhớ” trạng thái bắt đầu.

Ví dụ này cho thấy các hệ quả “lạ lùng” của tính toán lượng tử so với tính toán cổ điển. \square

Ví dụ 2.5.2. Xét bài toán **kiểm tra chẵn-lẻ (parity check)**: cho chuỗi 2 bit

$$b = b_1 b_0 \in \mathbb{B}^2,$$

tính $b_0 \oplus b_1$.

Rõ ràng, các **thuật toán cổ điển** (classical algorithm) cần “truy cập” chuỗi b 2 lần để có b_0 và b_1 , mới tính được $b_0 \oplus b_1$. Tuy nhiên, **thuật toán lượng tử** (quantum algorithm) 9 chỉ cần 1 lần truy cập chuỗi b .

Thuật toán 9 Thuật toán Deutsch.

Input: $b = b_1 b_0 \in \mathbb{B}^2$.

Output: $b_0 \oplus b_1$.

- Chuẩn bị trạng thái lượng tử sau với 1 lần truy cập b (ta sẽ rõ cách làm trong Phần 5.3.2)

$$|\psi\rangle = \frac{1}{\sqrt{2}} \left((-1)^{b_0} |0\rangle + (-1)^{b_1} |1\rangle \right).$$

- Áp dụng thao tác Hadamard để được trạng thái lượng tử

$$|\phi\rangle = H|\psi\rangle.$$

- Đo $|\phi\rangle$ và trả về kết quả đo.
-

Thuật toán Deutsch trả về kết quả là $b_0 \oplus b_1$. Thật vậy, ta có (♣)

$$|\psi\rangle = \frac{1}{\sqrt{2}} \left((-1)^{b_0} |0\rangle + (-1)^{b_1} |1\rangle \right) = \frac{(-1)^{b_0}}{\sqrt{2}} \left(|0\rangle + (-1)^{b_0 \oplus b_1} |1\rangle \right).$$

Nếu $b_0 \oplus b_1 = 0$ thì

$$|\psi\rangle = \frac{(-1)^{b_0}}{\sqrt{2}} (|0\rangle + |1\rangle) = (-1)^{b_0} |+\rangle$$

nên

$$|\phi\rangle = H|\psi\rangle = H \left((-1)^{b_0} |+\rangle \right) = (-1)^{b_0} |0\rangle.$$

Khi đo $|\phi\rangle$ chắc chắn được 0. Ngược lại, nếu $b_0 \oplus b_1 = 1$ thì $|\psi\rangle = (-1)^{b_0} |-\rangle$ nên $|\phi\rangle = (-1)^{b_0} |1\rangle$. Khi đo $|\phi\rangle$ chắc chắn được 1.

Ví dụ này cho thấy, nếu các đặc trưng của tính toán lượng tử được tận dụng một cách khéo léo thì nó có thể tạo ra các “ưu thế” so với tính toán cổ điển. \square

Bài tập

2.1 Có bao nhiêu trạng thái có thể khi tung một đồng xu 2 lần? Tìm cách mã hóa các trạng thái này bằng chuỗi bit phù hợp. Làm tương tự với 5 lần và n lần.

2.2 Tương tự Bài tập 2.1 với xúc xắc thay cho đồng xu.

2.3 Cho biết giá trị mà các số nhị phân sau biểu diễn

- (a) 10100011. (b) 0.010110. (c) 1011.1101.

2.4 Tìm số nhị phân biểu diễn cho các giá trị sau

- (a) 2025. (b) 0.34375. (c) 18.01.

2.5 Tìm hiểu dạng **bù 2** (two's complement) là cách để biểu diễn (mã hóa) các số nguyên có dấu (signed integer) bao gồm cả các số nguyên âm. Giả sử dùng chuỗi 1 byte (8 bit), cho biết chuỗi bit trong mã bù 2 của các số nguyên sau đây

- (a) 0. (b) 1. (c) -1. (d) -100.

2.6 Tìm hiểu về mã **màu RGB** và cho biết mã của các màu trong 7 sắc cầu vồng: đỏ, cam, vàng, lục, lam, chàm, tím.

2.7 Tìm hiểu mã **Unicode** và lược đồ mã UTF-8, cho biết chuỗi bit mã hóa UTF-8 cho thông điệp: Chào!

2.8 Tìm hiểu cách mã số thực **IEEE FP32** và cho biết chuỗi bit mã hóa cho các giá trị trong Bài tập 2.4 theo cách mã hóa này.

2.9 Với A, B là 2 bit đầu vào, các cổng 2 bit sau cũng hay được dùng

- cổng negative-OR: $\overline{A} + \overline{B}$,
- cổng negative-AND: $\overline{A} \overline{B}$.

Lập bảng chân trị và tìm hiểu ký hiệu mạch hay được dùng cho 2 cổng này.

2.10 Thiết kế mạch điện tương tự Hình 2.3 cho mạch nửa cộng ở Ví dụ 2.2.1.

2.11 Thiết kế mạch logic thực hiện phép nhân hai số nhị phân

- (a) 1 bit. (b) 2 bit. (c) 4 bit.

và đánh giá độ phức tạp mạch.

2.12 Thiết kế mạch cài đặt cổng XOR bằng tập các cổng toàn năng sau

- (a) {NOT, AND, OR}. (b) {NOT, AND}. (c) {NAND}

2.13 Cho $f : \mathbb{B}^3 \rightarrow \mathbb{B}$ với bảng chân trị sau

A	B	C	$f(A, B, C)$
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

- (a) Tìm dạng DNF của f . Dạng này cần bao nhiêu cổng?
 (b) Đơn giản mạch ở Câu (a) bằng cách dùng các tương đương logic phù hợp. Biểu thức đã đơn giản cần bao nhiêu cổng?

2.14 Đơn giản mạch cho

- (a) $f(A, B, C) = (A \vee B)(\bar{A} \vee B \vee C)\bar{C}$.
 (b) $f(A, B, C) = (A \vee \bar{B})(A \vee B \vee \bar{C})$.

2.15 Chứng minh rằng, với mọi $f : \mathbb{B}^n \rightarrow \mathbb{B}$, cổng G ở Hình 2.5 khả nghịch và tìm nghịch đảo của G .

2.16 Mở rộng kĩ thuật xây dựng cổng khả nghịch như Hình 2.5 cho $f : \mathbb{B}^n \rightarrow \mathbb{B}^m$.

2.17 Xét cổng Toffoli ở Ví dụ 2.2.5.

- (a) Lập bảng chân trị cho cổng Toffoli.
 (b) Thiết kế mạch cài đặt cổng Toffoli dùng các cổng cơ bản.
 (c) Cổng Toffoli có khả nghịch không? Nếu có thì nghịch đảo của nó là gì?

2.18 Cổng SWAP là cổng biến đổi 2 bit đầu vào AB thành 2 bit đầu ra CD thực hiện thao tác hoán đổi A, B , nghĩa là $C = B, D = A$.

- (a) Lập bảng chân trị cho SWAP.
 (b) Thiết kế mạch cài đặt SWAP.
 (c) SWAP có khả nghịch không? Nếu có thì nghịch đảo của nó là gì?

2.19 Cổng Fredkin là cổng biến đổi 3 bit đầu vào ABC thành 3 bit đầu ra $A'B'C'$ thực hiện thao tác hoán đổi có điều khiển (controlled-SWAP) B, C theo A , nghĩa là $A' = A$ và nếu $A = 1$ thì hoán đổi B, C thành B', C' còn không thì giữ nguyên (B', C' là B, C).

- (a) Lập bảng chân trị cho cổng Fredkin.
- (b) Thiết kế mạch cài đặt cổng Fredkin dùng các cổng cơ bản.
- (c) Cổng Fredkin có khả nghịch không? Nếu có thì nghịch đảo của nó là gì?
- (d) Chứng tỏ cổng Fredkin là một cổng toàn năng.

2.20 Tất cả các cổng điều khiển đều có phiên bản “ngược” (anti) trong đó việc thực hiện được điều khiển bằng bit 0 thay vì bit 1. Lập bảng chân trị và thiết kế mạch cho phiên bản “ngược” của các cổng điều khiển sau

- (a) CNOT.
- (b) Toffoli.
- (c) Fredkin.

2.21 Mở rộng ý tưởng của Hình 2.5, thiết kế mạch khả nghịch cho phép toán $f : \mathbb{B}^n \rightarrow \mathbb{B}^m$ và áp dụng để thiết kế phiên bản khả nghịch cho

- (a) Mạch nửa cộng HA.
- (b) Mạch toàn cộng FA.

2.22 Tương tự Ví dụ 2.3.3, thiết kế và phân tích độ phức tạp của thuật toán giải bài toán chia (division): cho các số nguyên không âm A, B với $B \neq 0$, tìm thương q và số dư r khi chia A cho B , tức q, r là các số nguyên không âm thỏa

$$A = qB + r, \quad 0 \leq r < B.$$

2.23 Chọn ngẫu nhiên chuỗi n bit b từ tập \mathbb{B}^n và đưa b làm đầu vào cho Thuật toán 3. Gọi T là số phép so sánh bit của Thuật toán 3 khi đó. Chứng minh T là biến ngẫu nhiên có kì vọng

$$E(T) = \Theta(n).$$

$E(T)$ còn được gọi là **độ phức tạp trường hợp trung bình** (average-case complexity) của Thuật toán 3.

2.24 Gọi T là số phép so sánh bit cần dùng (cũng là số lần lặp của vòng lặp 1-5) của Thuật toán 6, chứng minh

- (a) T là biến ngẫu nhiên có phân phối hình học với tham số $p = \frac{1}{2}$. Từ đó,

$$E(T) = \frac{1}{p} = 2.$$

(b)

$$P(T > k) > 0, \forall k \in \mathbb{N}.$$

(c)

$$P(T < \infty) = 1.$$

2.25 Chọn $k \in \mathbb{N}_{>0}$ cố định, chứng minh xác suất lỗi của Thuật toán 7 là

$$P(\text{fail}) = \left(\frac{1}{2}\right)^k.$$

2.26 Xét bài toán “kiểm tra phép nhân ma trận” trong Ví dụ 2.4.2. Chứng minh: nếu $AB \neq C$ và r được chọn ngẫu nhiên từ tập \mathbb{B}^n thì

$$P(ABr = Cr) \leq \frac{1}{2}.$$

Từ đó, xác suất lỗi của Thuật toán 8 thỏa

$$P(\text{lỗi}) \leq \left(\frac{1}{2}\right)^k.$$

2.27 Ma trận B trong Ví dụ 2.5.1 mô hình cho giao thoa kế Mach–Zehnder (Mach–Zehnder interferometer). Tìm hiểu về thiết bị này và thí nghiệm lượng tử dùng nó.

Chương 3

Qubit

Chương này trình bày đối tượng cơ bản nhất của tính toán lượng tử, quantum bit, qubit. Cơ sở toán học ở Chương 1 được vận dụng đầy thú vị để mở rộng mô hình tính toán ở Chương 2. Các khái niệm như trạng thái, phép đo, cổng và mạch sẽ được lặp lại nhưng cho đối tượng mới là qubit thay vì bit. Một công cụ trực quan để mô tả trạng thái và các biến đổi trên qubit là mặt cầu Bloch cũng được trình bày cùng với một ví dụ về hiện thực vật lý của qubit.

3.1 Qubit

3.1.1 Chồng chất

Một **bit lượng tử** (qubit, quantum bit) là một hệ thống lượng tử có trạng thái là sự **chồng chất** (superposition) của 2 trạng thái cơ bản. Hệ thống này còn được gọi là **hệ thống lượng tử 2 mức** (two-level quantum system).

Nếu kí hiệu 2 trạng thái cơ bản của hệ là 0, 1 và đồng nhất với 2 ket

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \in \mathbb{C}^2$$

thì trạng thái ψ của qubit được mô tả là một tổ hợp tuyến tính của $|0\rangle, |1\rangle$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \in \mathbb{C}^2. \quad (3.1)$$

Hơn nữa, để thuận tiện tính toán, ta yêu cầu $|\psi\rangle$ phải được chuẩn hóa ($|\psi\rangle$ là vector đơn vị)

$$\| |\psi\rangle \| = 1 \Leftrightarrow |\alpha|^2 + |\beta|^2 = 1. \quad (3.2)$$

Lưu ý phân biệt các kí hiệu: 0, 1 là 2 trạng thái cơ bản còn $|0\rangle, |1\rangle$ là 2 ket đặc biệt của \mathbb{C}^2 (2 vector cơ sở chuẩn tắc). Cũng lưu ý, nếu ψ kí hiệu trạng thái lượng tử thì ket $|\psi\rangle$ là vector đơn vị của \mathbb{C}^2 biểu diễn ψ . Hơn nữa, trong tính toán lượng tử, *trạng thái lượng tử và ket biểu diễn nó được dùng lẫn lộn!*

Các trạng thái sau đây của qubit rất hay được dùng (kiểm tra chúng là các vector đơn vị của \mathbb{C}^2)

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

Nhớ rằng các nhãn bên trong dấu ket $|\cdot\rangle$ (0, 1, +, -, i , $-i$) là tùy ý nhưng đã được chọn phổ biến. Nói cách khác, đây là các trạng thái lượng tử “nổi tiếng và đã được đặt tên”. Danh sách các kí hiệu này cùng với các kí hiệu nổi tiếng khác của tính toán lượng tử cũng được cho ở Phụ lục C để tiện tham khảo.

3.1.2 Phép đo

Ta đã biết $B_Z = \{|0\rangle, |1\rangle\}$ là một cơ sở trực chuẩn của \mathbb{C}^2 , được gọi là cơ sở chuẩn tắc hay cơ sở Z. Trong tính toán lượng tử, cơ sở này còn được gọi là **cơ sở tính toán** (computational basis).

Một qubit ở trạng thái $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ khi được thực hiện **phép đo** (measurement) theo cơ sở tính toán sẽ được 1 trong 2 kết quả

- được 0 với xác suất $|\alpha|^2$ và ngay sau khi đo thì qubit chuyển sang trạng thái $|0\rangle$,
- được 1 với xác suất $|\beta|^2$ và ngay sau khi đo thì qubit chuyển sang trạng thái $|1\rangle$.

Việc chuyển trạng thái khi đo còn được gọi là sự **sụp đổ** (collapse) trạng thái.

Để phân biệt, trong trạng thái lượng tử $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, các hệ số α, β được gọi là các **amplitude**, còn $|\alpha|^2, |\beta|^2$ là các xác suất. Nhận xét, $|\alpha|^2 + |\beta|^2 = 1$ như yêu cầu của xác suất vì việc đo qubit chỉ cho ra 1 trong 2 kết quả.

Tổng quát, cho $B = \{|a\rangle, |b\rangle\}$ là một cơ sở trực chuẩn của \mathbb{C}^2 , trạng thái $|\psi\rangle$ của qubit có thể được viết theo cơ sở B là

$$|\psi\rangle = \langle a|\psi\rangle|a\rangle + \langle b|\psi\rangle|b\rangle.$$

Khi đó, phép đo theo cơ sở B sẽ cho 1 trong 2 kết quả

- được a với xác suất $|\langle a|\psi\rangle|^2$ và sụp đổ thành $|a\rangle$,
- được b với xác suất $|\langle b|\psi\rangle|^2$ và sụp đổ thành $|b\rangle$.

Lưu ý, $(\langle a|\psi\rangle, \langle b|\psi\rangle)$ là tọa độ của $|\psi\rangle$ theo cơ sở B . Ta cũng phân biệt “kết quả báo ra” của phép đo là các “nhãn” a, b và trạng thái sụp đổ thành $|a\rangle, |b\rangle$. Nhãn báo a, b có thể được chọn tùy ý (tùy cách thông báo của thiết bị đo) còn trạng thái $|a\rangle, |b\rangle$ chính là các vector cơ sở gắn với phép đo. Thông thường nếu các vector cơ sở được sắp theo thứ tự thì nhãn tương ứng là số thứ tự 0, 1.

Nhận xét (♣)

- $|\langle a|\psi\rangle|^2 + |\langle b|\psi\rangle|^2 = 1$,
- phép đo theo cơ sở tính toán chỉ là một trường hợp của định nghĩa này.

Ví dụ 3.1.1. Qubit ở trạng thái $|0\rangle$ khi đo (theo cơ sở tính toán) sẽ chắc chắn được 0 (xác suất được 0 là 100%) và vẫn giữ trạng thái $|0\rangle$ sau khi đo. Qubit ở trạng thái $|1\rangle$ khi đo sẽ chắc chắn được 1 và vẫn giữ trạng thái $|1\rangle$. Do đó, $|0\rangle, |1\rangle$ tương tự như 0, 1 của **bit cổ điển** (classical bit).

Các trạng thái $|+\rangle, |-\rangle, |i\rangle, |-i\rangle$ là các trạng thái “tổ hợp đều” vì khi đo sẽ được 0 hoặc 1 với xác suất như nhau (50% được 0, 50% được 1). Chẳng hạn, xác suất được 0, 1 khi đo $|i\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{i}{\sqrt{2}}|1\rangle$ lần lượt là $\left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2}, \left|\frac{i}{\sqrt{2}}\right|^2 = \frac{1}{2}$.

Trạng thái $|\psi\rangle = \frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle$ thì “thiên về 0” vì khi đo, xác suất được 0 là $\left|\frac{\sqrt{3}}{2}\right|^2 = \frac{3}{4}$ (xác suất được 1 là $\frac{1}{4}$).

Trạng thái $|0\rangle$ viết theo cơ sở $B_X = \{|+\rangle, |-\rangle\}$ là

$$|0\rangle = \langle +|0\rangle|+\rangle + \langle -|0\rangle|-\rangle = \frac{1}{\sqrt{2}}|+\rangle + \frac{1}{\sqrt{2}}|-\rangle$$

nên khi đo $|0\rangle$ theo cơ sở X sẽ được $|+\rangle$ với xác suất 50% (được $|-\rangle$ xác suất 50%). Ngược lại, khi đo $|+\rangle$ theo X sẽ chắc chắn được $|+\rangle$.

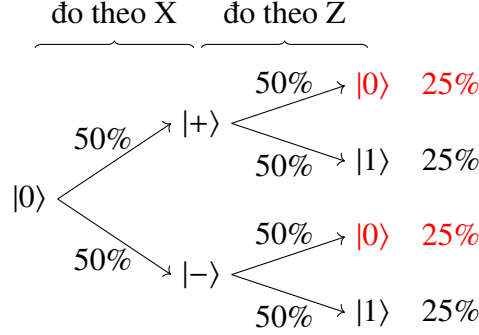
Khi đo $|\psi\rangle = \frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle$ theo X sẽ được $|+\rangle$ với xác suất là

$$|\langle +|\psi\rangle|^2 = \left| \frac{1}{\sqrt{2}} \frac{\sqrt{3}}{2} + \frac{1}{\sqrt{2}} \frac{1}{2} \right|^2 = \frac{(1 + \sqrt{3})^2}{8} \approx 0.93.$$

(xác suất được $|-\rangle$ là khoảng 7%.)

Hỏi. Từ trạng thái $|0\rangle$ nếu đo liên tiếp theo cơ sở X rồi Z thì được 0 với xác suất bao nhiêu?

Trả lời. $25\% + 25\% = 50\%$ như sơ đồ sau cho thấy



□

3.2 Pha chung, pha tương đối và mặt cầu Bloch

3.2.1 Pha chung

Dùng dạng mũ của số phức, trạng thái qubit (3.1) có thể được mô tả bằng dạng

$$|\psi\rangle = \rho_0 e^{i\theta_0} |0\rangle + \rho_1 e^{i\theta_1} |1\rangle$$

với $\rho_0, \rho_1 \in \mathbb{R}_{\geq 0}$ và $\theta_0, \theta_1 \in \mathbb{R}$ lần lượt là độ lớn và pha của các amplitude. Khi đó, yêu cầu chuẩn hóa (3.2) có nghĩa là

$$\| |\psi\rangle \| = 1 \iff |\rho_0 e^{i\theta_0}|^2 + |\rho_1 e^{i\theta_1}|^2 = 1 \iff \rho_0^2 + \rho_1^2 = 1.$$

Cho $\theta \in \mathbb{R}$, xét ket $|\phi\rangle = e^{i\theta} |\psi\rangle$, ta có

$$|\phi\rangle = e^{i\theta} |\psi\rangle = \rho_0 e^{i(\theta_0+\theta)} |0\rangle + \rho_1 e^{i(\theta_1+\theta)} |1\rangle$$

nên θ được gọi là **pha chung** (global phase) của các amplitude của $|\phi\rangle$.

Nhận xét

- $\| |\phi\rangle \| = \| e^{i\theta} |\psi\rangle \| = |e^{i\theta}| \| |\psi\rangle \| = 1$.
- Nếu đo $|\phi\rangle$ theo cơ sở trực chuẩn $B = \{|a\rangle, |b\rangle\}$ bất kì của \mathbb{C}^2 thì được $|a\rangle$ với xác suất

$$|\langle a|\phi\rangle|^2 = |\langle a|(e^{i\theta} |\psi\rangle)|^2 = |e^{i\theta} \langle a|\psi\rangle|^2 = |\langle a|\psi\rangle|^2$$

chính là xác suất được $|a\rangle$ khi đo $|\psi\rangle$ theo B (do đó xác suất được $|b\rangle$ cũng như nhau khi đo $|\phi\rangle$ hay $|\psi\rangle$).

- Ta sẽ thấy (Phần 3.3.1), biến đổi T trên qubit là toán tử tuyến tính của \mathbb{C}^2 nên

$$T|\phi\rangle = T(e^{i\theta}|\psi\rangle) = e^{i\theta}T|\psi\rangle.$$

Như vậy, $|\psi\rangle$ và $|\phi\rangle = e^{i\theta}|\psi\rangle$ đều mô tả cùng một trạng thái lượng tử nên các ket $|\psi\rangle, |\phi\rangle$ được gọi là **tương đương** (equivalent) nhau, kí hiệu $|\psi\rangle \equiv |\phi\rangle$.¹ Ta cũng nói, *pha chung không có ý nghĩa Vật lý!* Khi tính toán, ta “được phép” chọn pha chung tùy ý.

3.2.2 Pha tương đối

Vì pha chung không có ý nghĩa Vật lý nên ta có thể viết lại ket

$$|\psi\rangle = \rho_0 e^{i\theta_0} |0\rangle + \rho_1 e^{i\theta_1} |1\rangle$$

bằng ket tương đương

$$|\phi\rangle \equiv e^{i(-\theta_0)} |\psi\rangle = \rho_0 e^{i(\theta_0 - \theta_0)} |0\rangle + \rho_1 e^{i(\theta_1 - \theta_0)} |1\rangle = \rho_0 |0\rangle + \rho_1 e^{i(\theta_1 - \theta_0)} |1\rangle.$$

Như vậy, trạng thái lượng tử của một qubit có thể được viết ở dạng

$$|\psi\rangle = \rho_0 |0\rangle + \rho_1 e^{i\phi} |1\rangle$$

với $\rho_0, \rho_1 \in \mathbb{R}_{\geq 0}$ và $\phi \in [0, 2\pi)$. Số thực ϕ được gọi là **pha tương đối** của $|\psi\rangle$ (rõ hơn là pha tương đối của amplitude $\rho_1 e^{i\phi}$ với amplitude ρ_0).

Pha tương đối có ý nghĩa Vật lý vì các trạng thái lượng tử

$$|\psi\rangle = \rho_0 |0\rangle + \rho_1 e^{i\phi} |1\rangle, \quad |\psi'\rangle = \rho_0 |0\rangle + \rho_1 e^{i\phi'} |1\rangle$$

với $\phi \neq \phi'$ có thể được phân biệt bằng phép đo theo một cơ sở trực chuẩn nào đó.

Ví dụ 3.2.1. Hai ket $|a\rangle = -\frac{1}{\sqrt{2}}|0\rangle - \frac{i}{\sqrt{2}}|1\rangle$ và $|b\rangle = \frac{i}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$ tương đương nhau ($|a\rangle \equiv |b\rangle$) vì

$$\begin{aligned} |b\rangle &= \frac{i}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle = -i \left(\frac{-1}{\sqrt{2}}|0\rangle + \frac{-i}{\sqrt{2}}|1\rangle \right) \\ &= e^{i\frac{3\pi}{2}} \left(-\frac{1}{\sqrt{2}}|0\rangle - \frac{i}{\sqrt{2}}|1\rangle \right) = e^{i\frac{3\pi}{2}} |a\rangle, \end{aligned}$$

tức là $|a\rangle, |b\rangle$ chỉ khác pha chung là $\frac{3\pi}{2}$.

¹pha chung của các amplitude cũng tương tự như thừa số chung của tử mẫu phân số, ví dụ $\frac{2}{3}$ và $\frac{4}{6}$ được xem là các phân số bằng nhau, dù chúng là các cặp tử, mẫu khác nhau ($(2, 3) \neq (4, 6)$).

Ta cũng thấy $|a\rangle, |b\rangle$ mô tả cùng một trạng thái lượng tử mà thường được mô tả “thuận tiện” hơn là

$$|i\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{i}{\sqrt{2}}|1\rangle.$$

Ta có

$$\begin{aligned} |i\rangle &= \frac{1}{\sqrt{2}}|0\rangle + \frac{i}{\sqrt{2}}|1\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}e^{i\frac{\pi}{2}}|1\rangle, \\ |+\rangle &= \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}e^{i0}|1\rangle. \end{aligned}$$

Như vậy, $|i\rangle$ và $|+\rangle$ chỉ khác nhau pha tương đối là $\frac{\pi}{2}$ và 0.

Vì pha tương đối có ý nghĩa Vật lý nên $|i\rangle$ và $|+\rangle$ là 2 trạng thái lượng tử thật sự khác nhau. Chẳng hạn, nếu đo theo cơ sở $B_X = \{|+\rangle, |-\rangle\}$ thì xác suất để được $|+\rangle$ khi đo $|i\rangle$ và $|+\rangle$ lần lượt là

$$|\langle +|i\rangle|^2 = \left| \frac{1}{2} + \frac{i}{2} \right|^2 = \frac{1}{2}, \quad |\langle +|+\rangle|^2 = 1.$$

□

3.2.3 Mặt cầu Bloch

Ta đã thấy trạng thái lượng tử của một qubit có thể được viết ở dạng

$$|\psi\rangle = \rho_0|0\rangle + \rho_1 e^{i\phi}|1\rangle$$

với $\rho_0, \rho_1 \in \mathbb{R}_{\geq 0}$ và $\phi \in [0, 2\pi)$. Vì $\| |\psi\rangle \| = 1$ nên

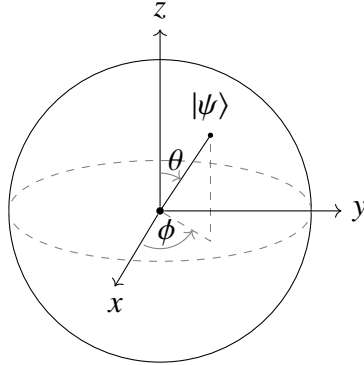
$$|\rho_0|^2 + |\rho_1 e^{i\phi}|^2 = \rho_0^2 + \rho_1^2 = 1.$$

Do đó $\rho_0 = \cos\left(\frac{\theta}{2}\right), \rho_1 = \sin\left(\frac{\theta}{2}\right)$ với duy nhất θ nào đó trong $[0, \pi)$.

Như vậy, trạng thái lượng tử của một qubit được xác định bằng ket có dạng (gọi là **dạng Bloch** hoặc **dạng chuẩn tắc**)

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi} \sin\left(\frac{\theta}{2}\right)|1\rangle, \quad \theta \in [0, \pi), \phi \in [0, 2\pi). \quad (3.3)$$

Vì mỗi điểm trên mặt cầu đơn vị có thể được xác định bằng **tọa độ cầu** (spherical coordinate) (θ, ϕ) với θ là “**vĩ độ**” (colatitude) và ϕ là “**kinh độ**” (longitude) nên mỗi trạng thái lượng tử (3.3) có thể được biểu diễn bằng một điểm tương ứng trên



Hình 3.1: Mặt cầu Bloch.

mặt cầu đơn vị còn được gọi là **mặt cầu Bloch** (Bloch sphere surface) như minh họa ở Hình 3.1

Khi đo qubit có trạng thái (viết theo dạng Bloch)

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

theo cơ sở tính toán sẽ được 0 với xác suất $\cos^2\left(\frac{\theta}{2}\right)$ (xác suất được 1 là $\sin^2\left(\frac{\theta}{2}\right) = 1 - \cos^2\left(\frac{\theta}{2}\right)$). Đặc biệt, các “tổ hợp đều” của $|0\rangle, |1\rangle$ là các điểm nằm trên **đường xích đạo** (equator) của mặt cầu Bloch vì

$$\cos^2\left(\frac{\theta}{2}\right) = \frac{1}{2} \iff \frac{\theta}{2} = \frac{\pi}{4} \iff \theta = \frac{\pi}{2}.$$

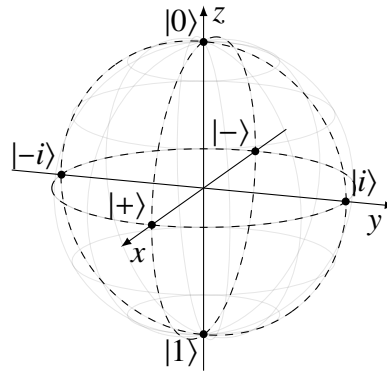
Mệnh đề 3.2.1. Nếu $\{|a\rangle, |b\rangle\}$ trực chuẩn trong \mathbb{C}^2 thì $|a\rangle, |b\rangle$ được biểu diễn bằng 2 điểm **đối qua tâm** (antipode) trên mặt cầu Bloch. \triangle

Các trạng thái lượng tử thông dụng có biểu diễn trên mặt cầu Bloch như Hình 3.2. Nhận xét $|0\rangle, |1\rangle$ đối qua tâm do $\{|0\rangle, |1\rangle\}$ trực chuẩn, hơn nữa $|0\rangle, |1\rangle$ nằm trên trục X nên $\{|0\rangle, |1\rangle\}$ được gọi là cơ sở X. Tương tự với $\{|+\rangle, |-\rangle\}$ và $\{|i\rangle, |-i\rangle\}$.

Ví dụ 3.2.2. Xét các ket $|a\rangle = \frac{\sqrt{3}}{2}|0\rangle + \frac{i}{2}|1\rangle, |b\rangle = \frac{i}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle$

- $\| |a\rangle \|^2 = \frac{3}{4} + \frac{1}{4} = 1, \| |b\rangle \|^2 = 1,$
- $\langle a|b\rangle = \frac{\sqrt{3}}{2} \frac{i}{2} - \frac{i}{2} \frac{\sqrt{3}}{2} = 0$

nên $\{|a\rangle, |b\rangle\}$ trực chuẩn.



Hình 3.2: Biểu diễn trên mặt cầu Bloch của các trạng thái lượng tử thông dụng.

Viết theo dạng Bloch

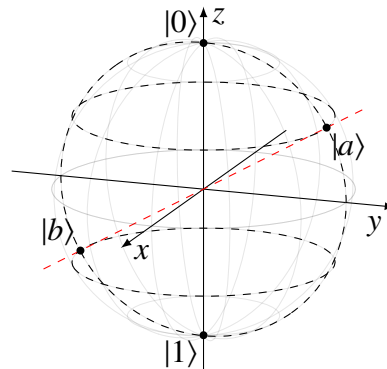
$$|a\rangle = \frac{\sqrt{3}}{2}|0\rangle + \frac{i}{2}|1\rangle = \frac{\sqrt{3}}{2}|0\rangle + i\frac{1}{2}|1\rangle = \cos\frac{\pi}{6}|0\rangle + e^{i\frac{\pi}{2}}\sin\frac{\pi}{6}|1\rangle,$$

$$|b\rangle \equiv -i\left(\frac{i}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle\right) = \frac{1}{2}|0\rangle + (-i)\frac{\sqrt{3}}{2}|1\rangle = \cos\frac{\pi}{3}|0\rangle + e^{i\frac{3\pi}{2}}\sin\frac{\pi}{3}|1\rangle.$$

Do đó, trên mặt cầu Bloch

- $|a\rangle$ có vĩ độ $\theta = \frac{\pi}{3}$, kinh độ $\phi = \frac{\pi}{2}$.
- $|b\rangle$ có vĩ độ $\theta = \frac{2\pi}{3}$, kinh độ $\phi = \frac{3\pi}{2}$.

Nhận xét $|a\rangle, |b\rangle$ đối qua tâm trên mặt cầu Bloch. Điều này là bắt buộc vì $\{|a\rangle, |b\rangle\}$ trực chuẩn.



□

3.3 Phép toán, công và mạch lượng tử

3.3.1 Phép toán lượng tử

Một **phép toán lượng tử (1 qubit)** (single-qubit quantum operation) là một thao tác biến đổi trạng thái của qubit $T : \mathbb{C}^2 \longrightarrow \mathbb{C}^2$ thỏa

- $T(\alpha|\phi\rangle + \beta|\psi\rangle) = \alpha T(|\phi\rangle) + \beta T(|\psi\rangle)$,
- $\|T(|\psi\rangle)\| = 1$ nếu $\| |\psi\rangle \| = 1$,

với mọi số phức α, β và mọi trạng thái $|\phi\rangle, |\psi\rangle$ của qubit.

Như vậy, T là một toán tử tuyến tính bảo toàn chuẩn trên \mathbb{C}^2 . Do đó T có thể được biểu diễn bằng một ma trận unita $U \in \mathbb{C}^{2 \times 2}$ thỏa

$$T(|\psi\rangle) = U|\psi\rangle, \quad |\psi\rangle \in \mathbb{C}^2.$$

Vì T tuyến tính nên T có thể được xác định bằng kết quả biến đổi trên các vector của một cơ sở, chẳng hạn trên $|0\rangle, |1\rangle$ của cơ sở tính toán. Nếu

$$|a\rangle = T|0\rangle, \quad |b\rangle = T|1\rangle$$

thì ma trận biểu diễn cho T trong cơ sở tính toán là

$$U = \begin{bmatrix} |a\rangle & |b\rangle \end{bmatrix}.$$

Hơn nữa, $\{|a\rangle, |b\rangle\}$ cũng lập thành một cơ sở trực chuẩn nên U là ma trận chuyển cơ sở tính toán sang cơ sở $\{|a\rangle, |b\rangle\}$.

Vì U unita nên U khả nghịch với $U^{-1} = U^\dagger$, do đó, T khả nghịch và ma trận biểu diễn cho biến đổi ngược của T chính là U^\dagger .

Vì U unita nên luôn tìm được 2 vector riêng $|u_1\rangle, |u_2\rangle$ lập thành một cơ sở trực chuẩn với các trị riêng tương ứng là $e^{i\theta_1}, e^{i\theta_2}$, khi đó

$$U|u_1\rangle = e^{i\theta_1}|u_1\rangle \equiv |u_1\rangle, \quad U|u_2\rangle = e^{i\theta_2}|u_2\rangle \equiv |u_2\rangle.$$

Do đó U biểu diễn cho phép xoay quanh trục tạo bởi $|u_1\rangle, |u_2\rangle$ một góc $\theta_2 - \theta_1$ trên mặt cầu Bloch. U cũng có thể được mô tả bằng tích ngoài

$$U = e^{i\theta_1}|u_1\rangle\langle u_1| + e^{i\theta_2}|u_2\rangle\langle u_2|.$$

Trong tính toán lượng tử, *phép toán lượng tử và ma trận unita biểu diễn nó được dùng lẫn lộn!*

3.3.2 Cổng lượng tử

Cổng lượng tử (quantum gate) mô hình các phép toán lượng tử cơ bản được dùng để xây dựng nên các phép toán khác. Sau đây là danh sách các cổng lượng tử 1-qubit thông dụng ²

- **Cổng đơn vị** (identity gate) tương ứng với toán tử đơn vị

$$I|0\rangle = |0\rangle, \quad I|1\rangle = |1\rangle.$$

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

I mô tả cho biến đổi “không làm gì cả” (NOOP) và chính là “phiên bản lượng tử” của cổng logic ID. Lưu ý, điều này không có nghĩa cổng logic ID là cổng lượng tử I vì ID chỉ xử lý 2 trạng thái cổ điển (bit) là 0, 1 còn I xử lý trạng thái lượng tử; chỉ là hành vi của I trên các trạng thái lượng tử đặc biệt $|0\rangle, |1\rangle$ (các vector cơ sở tính toán) thì tương tự hành vi của ID trên 0, 1.

I có ma trận biểu diễn

$$I = \begin{bmatrix} |0\rangle & |1\rangle \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = |0\rangle\langle 0| + |1\rangle\langle 1|.$$

Vì

$$I|\psi\rangle = |\psi\rangle, \quad \forall |\psi\rangle \in \mathbb{C}^2$$

nên mọi vector đều là vector riêng của I với trị riêng $1 = e^{i0}$. Do đó, I là phép quay góc 0° quanh trục bất kỳ của mặt cầu Bloch!

- **Cổng NOT** (NOT gate) hay **cổng Pauli X** (Pali X gate)

$$X|0\rangle = |1\rangle, \quad X|1\rangle = |0\rangle.$$

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

X mô tả cho biến đổi “**lật bit**” (bit flip) là phiên bản lượng tử của cổng logic NOT.

X có ma trận biểu diễn

$$X = \begin{bmatrix} |1\rangle & |0\rangle \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = |0\rangle\langle 1| + |1\rangle\langle 0|.$$

Vì (♣)

$$X|+\rangle = |+\rangle, \quad X|-\rangle = -|-\rangle,$$

nên X có 2 vector riêng trực chuẩn là $\{|+\rangle, |-\rangle\}$ với trị riêng tương ứng là $1 = e^{i0}, -1 = e^{i\pi}$. Do đó, X là phép quay góc 180° quanh trục x của mặt cầu Bloch.

²danh sách này cũng được tóm lược ở Phụ lục C để tiện tham khảo.

- **Cổng Pauli Z** (Pali Z gate)

$$Z|0\rangle = |0\rangle, \quad Z|1\rangle = -|1\rangle = e^{i\pi}|1\rangle.$$

$$Z(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle - \beta|1\rangle.$$

Nếu $\beta = \rho e^{i\theta}$ thì $-\beta = e^{i\pi}\rho e^{i\theta} = \rho e^{i(\theta+\pi)}$ nên Z mô tả cho biến đổi “**lật pha**” (phase flip) vì Z “đón thêm” pha tương đối π .

Z có ma trận biểu diễn

$$Z = \begin{bmatrix} |0\rangle & -|1\rangle \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = |0\rangle\langle 0| - |1\rangle\langle 1|.$$

Vì

$$Z|0\rangle = |0\rangle, \quad Z|1\rangle = -|1\rangle.$$

nên Z có 2 vector riêng trực chuẩn là $\{|0\rangle, |1\rangle\}$ với trị riêng tương ứng là $1, -1$. Do đó, Z là phép quay góc 180° quanh trục z của mặt cầu Bloch.

- **Cổng Pauli Y** (Pali Y gate)

$$Y|0\rangle = i|1\rangle, \quad Y|1\rangle = -i|0\rangle.$$

$$Y(\alpha|0\rangle + \beta|1\rangle) = i\alpha|1\rangle - i\beta|0\rangle = -i(\beta|0\rangle - \alpha|1\rangle) \equiv \beta|0\rangle - \alpha|1\rangle.$$

Y mô tả cho biến đổi “lật cả bit lẫn pha”.

Y có ma trận biểu diễn

$$Y = \begin{bmatrix} i|1\rangle & -i|0\rangle \end{bmatrix} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = -i|0\rangle\langle 1| + i|1\rangle\langle 0| \equiv |0\rangle\langle 1| - |1\rangle\langle 0|.$$

Vì (♣)

$$Y|i\rangle = |i\rangle, \quad Y|-i\rangle = -|-i\rangle,$$

nên Y có 2 vector riêng trực chuẩn là $\{|i\rangle, |-i\rangle\}$ với trị riêng tương ứng là $1, -1$. Do đó, Y là phép quay góc 180° quanh trục y của mặt cầu Bloch.

- **Cổng Hadamard** (Hadamard gate)

$$H|0\rangle = |+\rangle, \quad H|1\rangle = |-\rangle.$$

$$H(\alpha|0\rangle + \beta|1\rangle) = \alpha|+\rangle + \beta|-\rangle.$$

Vì $H|0\rangle = |+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ nên H mô tả cho biến đổi “tổ hợp đều”.

H có ma trận biểu diễn

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

Ta cũng có (♣)

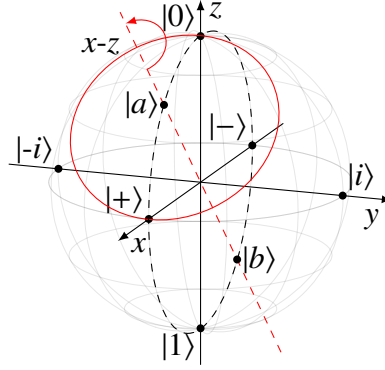
$$H|+\rangle = |0\rangle, \quad H|-\rangle = |1\rangle.$$

Do đó, ta “đoán” H là phép quay góc 180° quanh trục x-z của mặt cầu Bloch.

Đặt $|a\rangle = |0\rangle + |+\rangle$, $|b\rangle = |1\rangle - |-\rangle$, ta có (♣)

- $|a\rangle, |b\rangle$ là 2 vector riêng của H với trị riêng tương ứng là $1, -1$,
- $\left\{ \frac{|a\rangle}{\|a\rangle}, \frac{|b\rangle}{\|b\rangle} \right\}$ là một họ trực chuẩn,
- $|a\rangle$ có vĩ độ $\theta = \frac{\pi}{4}$, kinh độ $\phi = 0$.

Do đó, H là phép quay góc 180° quanh trục x-z.



• **Cổng pha** (phase gate)

$$S|0\rangle = |0\rangle, \quad S|1\rangle = i|1\rangle = e^{i\frac{\pi}{2}}|1\rangle.$$

$$S(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle + i\beta|1\rangle.$$

S mô tả cho biến đổi “đôn thêm” pha tương đối $\frac{\pi}{2}$.

S có ma trận biểu diễn

$$S = \begin{bmatrix} |0\rangle & i|1\rangle \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} = |0\rangle\langle 0| + i|1\rangle\langle 1|.$$

S là phép quay góc 90° quanh trục z của mặt cầu Bloch. (♣)

Nhận xét, $S^2 = SS$ mô tả cho phép biến đổi “ S rồi S nữa”, tức là làm 2 lần S , do đó là phép quay góc 180° quanh trục z, đó chính là biến đổi Z , như vậy $S^2 = Z$ nên ta thường viết $S = \sqrt{Z}$.

- **Cổng T** (T gate)

$$T|0\rangle = |0\rangle, \quad T|1\rangle = e^{i\frac{\pi}{4}}|1\rangle.$$

$$T(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle + e^{i\frac{\pi}{4}}\beta|1\rangle.$$

T mô tả cho biến đổi “đơn thêm” pha tương đối $\frac{\pi}{4}$.

T có ma trận biểu diễn

$$T = \begin{bmatrix} |0\rangle & e^{i\frac{\pi}{4}}|1\rangle \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix} = |0\rangle\langle 0| + e^{i\frac{\pi}{4}}|1\rangle\langle 1|.$$

T là phép quay góc 45° quanh trục z của mặt cầu Bloch. (♣)

Nhận xét, $T^2 = S$ và $T^4 = Z$ nên ta thường viết $T = \sqrt{S}$.

- **Cổng dịch pha**. Cho $\theta \in [0, 2\pi)$, **cổng dịch pha** (phase shift gate)

$$R(\theta)|0\rangle = |0\rangle, \quad R(\theta)|1\rangle = e^{i\theta}|1\rangle.$$

$$R(\theta)(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle + e^{i\theta}\beta|1\rangle.$$

$R(\theta)$ mô tả cho biến đổi “đơn thêm” pha tương đối θ .

$R(\theta)$ có ma trận biểu diễn

$$R(\theta) = \begin{bmatrix} |0\rangle & e^{i\theta}|1\rangle \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix} = |0\rangle\langle 0| + e^{i\theta}|1\rangle\langle 1|.$$

$R(\theta)$ là phép quay góc θ quanh trục z của mặt cầu Bloch. (♣)

$R(\theta)$ là **cổng được tham số hóa** (parameterized gate) theo **tham số** (parameter) θ vì tùy giá trị cụ thể của θ ta có $R(\theta)$ cụ thể, chẳng hạn

$$Z = R(\pi), \quad S = R\left(\frac{\pi}{2}\right), \quad T = R\left(\frac{\pi}{4}\right).$$

Nghịch đảo của cổng I là I . Các cổng X, Y, Z, H Hermite (♣) nên là nghịch đảo của chính nó, chẳng hạn

$$H^{-1} = H^\dagger = H.$$

Điều này là dễ hiểu vì các cổng này đều là các phép quay 180° quanh một trục của mặt cầu Bloch (tương ứng là các trục $x, y, z, x-z$) nên “quay thêm lần nữa là huề”.

Các cổng S, T (quay tương ứng $90^\circ, 45^\circ$ quanh trục z) không là nghịch đảo của chính nó. Dễ thấy (♣)

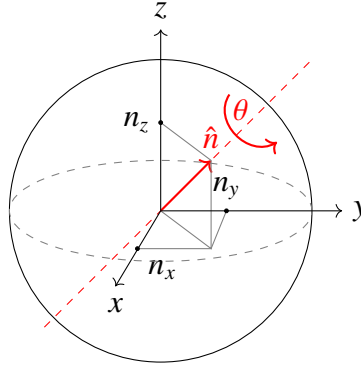
$$S^{-1} = S^3, \quad T^{-1} = T^7.$$

Cũng dễ thấy (♣)

$$R^{-1}(\theta) = R(2\pi - \theta) = R(-\theta).$$

Mệnh đề 3.3.1. Mọi phép toán lượng tử (1 qubit) đều là phép quay trên mặt cầu Bloch. Nếu $\hat{n} = (n_x, n_y, n_z) \in \mathbb{R}^3$ là vector đơn vị mô tả (bằng tọa độ Descartes) hướng của trục quay thì phép toán lượng tử thực hiện phép quay góc θ quanh trục \hat{n} của mặt cầu Bloch có ma trận biểu diễn là

$$U = \cos\left(\frac{\theta}{2}\right)I - i \sin\left(\frac{\theta}{2}\right)(n_x X + n_y Y + n_z Z) \quad (3.4)$$



△

Ví dụ 3.3.1. Gọi $R_z(\theta)$ là toán tử quay quanh trục z góc $\theta \in [0, 2\pi)$, vì trục z được xác định bởi vector đơn vị $\hat{n} = (0, 0, 1)$ nên từ (3.4) ta có

$$\begin{aligned} R_z(\theta) &= \cos\left(\frac{\theta}{2}\right)I - i \sin\left(\frac{\theta}{2}\right)Z \\ &= \cos\left(\frac{\theta}{2}\right)\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - i \sin\left(\frac{\theta}{2}\right)\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix}. \end{aligned}$$

Đây chính là cổng dịch pha $R(\theta)$ vì (nhớ là pha chung không có ý nghĩa Vật lý)

$$R_z(\theta) = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} = e^{-i\frac{\theta}{2}} \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix} \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix} = R(\theta).$$

□

3.3.3 Mạch lượng tử

Tương tự mạch cổ điển, **mạch lượng tử** (quantum circuit) là một mô hình của tính toán lượng tử, trong đó việc tính toán được mô tả bằng một dãy các cổng lượng tử (và/hoặc các thao tác khác như phép đo) trên các qubit. Các qubit được biểu diễn bằng các đường ngang và các cổng lượng tử được biểu diễn bằng các hộp nằm trên đường của qubit mà cổng tác động. Thứ tự tác động của các cổng là từ trái qua phải. Chẳng hạn, **sơ đồ mạch** (circuit diagram) sau

$$|\psi\rangle \text{ --- } \boxed{U_1} \text{ --- } \boxed{U_2} \text{ --- } \boxed{U_3} \text{ --- } |\phi\rangle$$

mô tả các “bước tính toán” để từ trạng thái đầu vào $|\psi\rangle$ được trạng thái đầu ra $|\phi\rangle$

$$|\phi\rangle = U_3 U_2 U_1 |\psi\rangle.$$

Các cổng lượng tử thông dụng được mô tả bằng các hộp có nhãn là kí hiệu của cổng. Chẳng hạn, sơ đồ mạch sau

$$|0\rangle \text{ --- } \boxed{H} \text{ --- } \boxed{Z} \text{ --- } \boxed{H} \text{ --- } |\phi\rangle$$

mô tả các tính toán: từ đầu vào $|0\rangle$ thực hiện cổng Hadamard H rồi cổng Pauli Z rồi lại cổng Hadamard để được đầu ra

$$\begin{aligned} |\phi\rangle &= HZH|0\rangle = HZ|+\rangle = HZ \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = H \frac{1}{\sqrt{2}}(Z|0\rangle + Z|1\rangle) \\ &= H \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = H|-\rangle = |1\rangle. \end{aligned}$$

Lưu ý, phép đo không phải là toán tử (nó thậm chí **không tất định** (nondeterministic) vì kết quả mang tính ngẫu nhiên). Tuy nhiên, phép đo vẫn được xem là một thao tác trên qubit (thường được thực hiện sau cùng để có kết quả là các bit cổ điển 0, 1).

Trong sơ đồ mạch sau

$$|0\rangle \text{ --- } \boxed{H} \text{ --- } \boxed{Z} \text{ --- } \boxed{H} \text{ --- } \boxed{\text{đo}} \text{ --- } \text{==}$$

kí hiệu “đồng hồ đo” mô tả phép đo và đường nét kép mô tả bit cổ điển (đường nét đơn mô tả qubit). Mạch này chắc chắn cho kết quả 1 như đã phân tích. Mạch sau

$$|0\rangle \text{ --- } \boxed{H} \text{ --- } \boxed{T} \text{ --- } \boxed{S} \text{ --- } \boxed{H} \text{ --- } \boxed{\text{đo}} \text{ --- } \text{==}$$

cho 1 với xác suất khoảng 85% (cho 0 với xác suất khoảng 15%). (♣)

Ví dụ 3.3.2. Mạch lượng tử sau

$$|0\rangle \text{ --- } \boxed{H} \text{ --- } \boxed{\text{đo}} \text{ --- } \text{==}$$

trông đơn giản nhưng là “ước mơ khó thành” của tính toán cổ điển!

Thật vậy, vì

$$H|0\rangle = |+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

nên mạch trên cho ra các bit cổ điển 0, 1 với xác suất đều là 50%.

Ngẫu nhiên là nguồn tài nguyên cực kỳ quan trọng trong tính toán cổ điển (như ta đã thấy trong Phần 2.4.1). Tuy nhiên, do bản chất **tất định** (deterministic), tính toán cổ điển chỉ có “**giả ngẫu nhiên**” (pseudorandom). Ngược lại, tính toán lượng tử tạo ra “**ngẫu nhiên thực sự**” (truly random). \square

3.4 Qubit vật lý

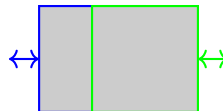
Các hệ thống lượng tử 2 mức trong thực tế khi được dùng làm qubit được gọi là **qubit vật lý** (physical qubit). Ngược lại, qubit trừu tượng bằng mô hình Toán học được gọi là **qubit logic** (logical qubit). Không giống các qubit logic, qubit vật lý chịu nhiều ràng buộc thực tế và công nghệ như tính ổn định, bị lỗi, mở rộng, ... Thông thường, cần nhiều qubit vật lý để có một qubit logic. Vài qubit vật lý hay dùng trong thực tế là ³

- **Photon** với 2 trạng thái **phân cực** (polarization) cơ bản là ngang (Horizontal) và dọc (Vertical).
- **Electron** với 2 trạng thái **spin** cơ bản là Up và Down.
- **Bẫy ion** (trapped ion) với 2 **mức năng lượng** (energy level) cơ bản là “trạng thái nghỉ” (ground state) và “trạng thái kích thích” (excited state).

Ví dụ 3.4.1. Xem thí nghiệm 3 kính lọc phân cực tại <https://www.youtube.com/watch?v=5SIxEiL8ujA>.

Kính lọc phân cực (polarizing filters) cho phép các hạt photon (ánh sáng) có trạng thái phân cực cùng hướng với kính đi qua và hấp thụ các photon khác.

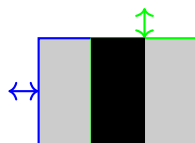
TN-1: đặt 2 kính lọc cùng hướng ngang chồng lên nhau.



Quan sát: lượng photon đi qua cả 2 kính giống như đi qua mỗi kính.

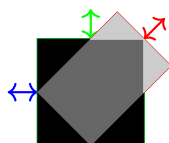
TN-2: đặt kính lọc hướng dọc chồng lên kính lọc hướng ngang.

³xem thêm https://en.wikipedia.org/wiki/Qubit#Physical_implementations.



Quan sát: không còn photon đi qua cả 2 kính.

TN-3: đặt kính lọc hướng chéo xen giữa hướng ngang và dọc.



Quan sát: có photon đi qua cả 3 kính!

Kết quả quan sát ở TN-3 là “không bình thường”: “lọc 2 lần thì hết mà 3 lần thì còn!”

Giải thích. Mỗi photon là một hệ thống lượng tử 2 mức với trạng thái phân cực là sự chồng chất của 2 trạng thái cơ bản. Theo mô hình tính toán lượng tử, mỗi photon là một qubit với trạng thái là ket đơn vị $|\psi\rangle \in \mathbb{C}^2$ là tổ hợp tuyến tính

$$|\psi\rangle = \alpha|a\rangle + \beta|b\rangle$$

của 2 vector trực chuẩn $|a\rangle, |b\rangle$ trong \mathbb{C}^2 .

Mỗi kính lọc tương ứng với một phép đo theo cơ sở trực chuẩn $\{|a\rangle, |b\rangle\}$, mà mỗi photon khi được đo sẽ sụp đổ trạng thái thành $|a\rangle$ với xác suất $|\alpha|^2$ và thành $|b\rangle$ với xác suất $|\beta|^2$. Hơn nữa, kính sẽ để photon $|a\rangle$ đi qua và hấp thụ photon $|b\rangle$.

Kí hiệu hướng phân cực ngang là $|0\rangle$, phân cực dọc là $|1\rangle$ thì $\{|0\rangle, |1\rangle\}$ trực chuẩn. Kính ngang có cơ sở trực chuẩn tương ứng là $\{|0\rangle, |1\rangle\}$ (để ngang đi qua và hấp thụ dọc) còn kính dọc có cơ sở trực chuẩn tương ứng là $\{|1\rangle, |0\rangle\}$ (để dọc đi qua và hấp thụ ngang).

Trong TN-1 và TN-2, photon đến kính ngang sẽ sụp đổ thành $|0\rangle$ hoặc $|1\rangle$ và photon $|0\rangle$ được đi qua còn $|1\rangle$ bị hấp thụ. Nếu ánh sáng môi trường là “phân cực ngẫu nhiên” thì có 50% lượng photon đi qua.

Sau đó, TN-1 đặt thêm kính ngang 2 thì photon qua kính ngang 1 có trạng thái $|0\rangle$ nên chắc chắn qua kính ngang 2. Do đó toàn bộ photon qua kính 1 cũng qua kính 2. Ngược lại, TN-2 đặt thêm kính dọc 2 thì photon $|0\rangle$ chắc chắn bị kính dọc hấp thụ nên không còn photon qua kính 2.

Hướng phân cực chéo là tổ hợp đều của $|0\rangle$ (ngang) và $|1\rangle$ (dọc) nên kính chéo có

cơ sở trực chuẩn tương ứng là $\{|+\rangle, |-\rangle\}$ với

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

Lưu ý, ta cũng có $|0\rangle = \frac{1}{\sqrt{2}}|+\rangle + \frac{1}{\sqrt{2}}|-\rangle$, $|1\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|-\rangle$. Như vậy, 3 kính trong TN-3 là

1. kính ngang 1 với cơ sở trực chuẩn tương ứng là $\{|0\rangle, |1\rangle\}$,
2. kính chéo 2 với cơ sở trực chuẩn tương ứng là $\{|+\rangle, |-\rangle\}$,
3. kính dọc 3 với cơ sở trực chuẩn tương ứng là $\{|1\rangle, |0\rangle\}$.

Trong TN-3

1. photon qua kính ngang 1 có trạng thái $|0\rangle$,
2. photon $|0\rangle = \frac{1}{\sqrt{2}}|+\rangle + \frac{1}{\sqrt{2}}|-\rangle$ khi gặp kính chéo 2 sẽ sụp đổ thành $|+\rangle$ với xác suất 50% và được đi qua (50% thành $|-\rangle$ bị hấp thụ),
3. photon $|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ khi gặp kính dọc 3 sẽ sụp đổ thành $|1\rangle$ với xác suất 50% và được đi qua (50% thành $|0\rangle$ bị hấp thụ).

Như vậy, trong các photon qua kính 1 thì có 50% qua được kính 2 và 25% qua được tiếp kính 3. Chung cuộc, nếu ánh sáng môi trường là “phân cực ngẫu nhiên” thì có 1/8 lượng photon đi qua cả 3 kính.

Lưu ý, nếu để kính chéo trước kính ngang hoặc sau kính dọc thì vẫn không có photon nào đi qua cả 3 kính. (♣) □

Bài tập

3.1 Khảo sát phép đo theo các cơ sở $B_Z = \{|0\rangle, |1\rangle\}$, $B_X = \{|+\rangle, |-\rangle\}$, $B_Y = \{|i\rangle, |-i\rangle\}$ của các trạng thái lượng tử sau

(a) $|\psi_1\rangle = \frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle$.

(b) $|\psi_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\frac{\pi}{6}}|1\rangle)$.

(c) $|\psi_3\rangle = \frac{2}{3}|0\rangle + \frac{1-2i}{3}|1\rangle$.

3.2 Cho $|a\rangle = \frac{\sqrt{3}}{2}|0\rangle - \frac{i}{2}|1\rangle$, $|b\rangle = \frac{i}{2}|0\rangle - \frac{\sqrt{3}}{2}|1\rangle$.

(a) Chứng minh $B = \{|a\rangle, |b\rangle\}$ là một cơ sở trực chuẩn của \mathbb{C}^2 .

(b) Khảo sát phép đo theo B của các trạng thái lượng tử ở Bài tập 3.1.

3.3 Với từng trạng thái lượng tử ở Bài tập 3.1, tính xác suất được 0 khi thực hiện liên tiếp các phép đo theo các cơ sở sau đây

- (a) X, Z . (b) Z, X, Z . (c) X, Y, Z . (d) Y, X, Z .

3.4 Viết dạng Bloch và mô tả trên mặt cầu Bloch các trạng thái ở Bài tập 3.1.

3.5 Cho thấy rằng biểu diễn trên mặt cầu Bloch của 2 ket $|a\rangle, |b\rangle$ ở Bài tập 3.2 là 2 điểm đối nhau qua tâm.

3.6 Cho $|\phi\rangle, |\psi\rangle$ là 2 trạng thái lượng tử, chứng minh $\{|\phi\rangle, |\psi\rangle\}$ trực chuẩn khi và chỉ khi biểu diễn trên mặt cầu Bloch của chúng là 2 điểm đối nhau qua tâm.

3.7 Cho $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ với $\| |\psi\rangle \| = 1$ và $B = \{|a\rangle, |b\rangle\}$ là một cơ sở trực chuẩn của \mathbb{C}^2 . Chứng minh

$$|\langle a|\psi\rangle|^2 + |\langle b|\psi\rangle|^2 = 1.$$

3.8 Cho các trạng thái lượng tử $|\psi\rangle = \rho_0|0\rangle + \rho_1 e^{i\phi}|1\rangle$ và $|\psi'\rangle = \rho_0|0\rangle + \rho_1 e^{i\phi'}|1\rangle$ với $\phi \neq \phi'$ (ρ_0, ρ_1 là các số thực không âm). Chứng minh, tồn tại một cơ sở trực chuẩn $B = \{|a\rangle, |b\rangle\}$ để phép đo theo B phân biệt được $|\psi\rangle$ và $|\psi'\rangle$, tức là xác suất được $|a\rangle$ khi đo $|\psi\rangle$ và $|\psi'\rangle$ theo B là khác nhau.

3.9 Cho biết kết quả biến đổi của các cổng I, X, Y, Z, H, S, T trên các trạng thái

$$|0\rangle, |1\rangle, |+\rangle, |-\rangle, |i\rangle, |-i\rangle$$

bằng cách

- (a) Biến đổi trực quan trên mặt cầu Bloch.
(b) Thực hiện phép nhân ma trận.

3.10 Xét tập các trạng thái lượng tử thông dụng

$$|0\rangle, |1\rangle, |+\rangle, |-\rangle, |i\rangle, |-i\rangle.$$

Cho biết phép toán lượng tử tương ứng (mô tả bằng các cổng lượng tử thông dụng) để biến đổi một trạng thái này thành một trạng thái khác trong tập trên.

3.11 Đặt $|\psi\rangle = HSTH|0\rangle$,

- (a) Vẽ $|\psi\rangle$ trên mặt cầu Bloch.
(b) Khảo sát phép đo $|\psi\rangle$ theo các cơ sở X, Y, Z .

3.12 Làm lại Bài tập 3.11 cho $|\psi\rangle = HYTHX|0\rangle$.

3.13 Cho U là một toán tử tuyến tính trên \mathbb{C}^2 , biết

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

- (a) Chứng minh U là một cổng lượng tử.
- (b) Cho biết kết quả biến đổi U trên các trạng thái $|+\rangle, |-\rangle, |i\rangle, |-i\rangle$.
- (c) Cho biết kết quả biến đổi U trên các trạng thái của Câu 1.
- (d) U tương ứng với phép quay quanh trục nào với góc bao nhiêu trên mặt cầu Bloch?

3.14 Tương tự Bài tập 3.13 cho toán tử tuyến tính U với

$$\begin{aligned} U|0\rangle &= \frac{\sqrt{3}}{2}|0\rangle + \frac{\sqrt{3}+i}{4}|1\rangle, \\ U|1\rangle &= \frac{\sqrt{3}+i}{4}|0\rangle - \frac{\sqrt{3}+3i}{4}|1\rangle. \end{aligned}$$

3.15 Tương tự Bài tập 3.13 cho toán tử tuyến tính U với

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

3.16 Chứng minh các khẳng định sau (các đẳng thức trên cổng lượng tử)

- (a) $X^2 = Y^2 = Z^2 = I$. (c) $X = HZH$. (e) $HYH = XYX = -Y$.
- (b) $H = \frac{1}{\sqrt{2}}(X + Z)$. (d) $Z = HXH$. (f) $XZXZ = ZXZX = -I$.

3.17 Ta đã biết $S = \sqrt{Z}$ vì $S^2 = Z$. Tương tự, tìm căn của các cổng X, Y, S, T, H .

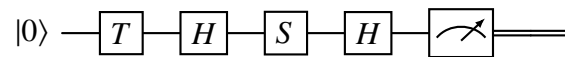
3.18 Gọi $R_x(\theta), R_y(\theta), R_z(\theta)$ là các toán tử quay góc $\theta \in [0, 2\pi)$ lần lượt quanh các trục x, y, z của mặt cầu Bloch. Xác định ma trận biểu diễn chúng.

3.19 Gọi $R_{xy}(\theta), R_{xz}(\theta), R_{yz}(\theta)$ là các toán tử quay góc $\theta \in [0, 2\pi)$ lần lượt quanh các trục $x-y, x-z, y-z$ của mặt cầu Bloch. Xác định ma trận biểu diễn chúng.

3.20 Gọi T là phép quay 180° quanh trục $x-y-z$ của mặt cầu Bloch.

- (a) Tìm $T|0\rangle, T|1\rangle$ bằng cách quay trực quan trên mặt cầu Bloch rồi từ đó xác định ma trận biểu diễn của T .
- (b) Dùng công thức công tổng quát xác định ma trận biểu diễn của T và so với kết quả của Câu (a).

3.21 Cho biết kết quả của mạch lượng tử sau



3.22 Từ trạng thái đầu vào $|0\rangle$,

- (a) Vẽ mạch mô tả tính toán $HYTHX$.
- (b) Tính đầu ra của Câu (a).
- (c) Thêm phép đo ở cuối mạch của Câu (a) và tính xác suất được 1.

Chương 4

Hệ nhiều qubit

Tương tự như việc ghép nhiều bit để mã hóa thông tin trong tính toán cổ điển, việc ghép nhiều qubit sẽ mở rộng khả năng biểu diễn và tính toán trong tính toán lượng tử. Tuy nhiên, nếu như việc ghép n bit chỉ được tập các chuỗi bit \mathbb{B}^n thì việc ghép n qubit sẽ có không gian “khổng lồ” \mathbb{C}^{2^n} . Việc tính toán trên không gian này cũng mang lại nhiều tiềm năng và lợi thế cho tính toán lượng tử nhờ các đặc trưng như chồng chất, giao thoa hay vướng.

4.1 Hệ nhiều qubit

4.1.1 Trạng thái lượng tử

Cho hệ n qubit, để thuận tiện, ta đánh số các qubit lần lượt là $0, 1, \dots, n-1$. Vì mỗi qubit có 2 trạng thái cơ bản là 0, 1 nên hệ n qubit có 2^n trạng thái cơ bản. Trạng thái cơ bản của hệ ứng với việc qubit i có trạng thái b_i ($i = 0, \dots, n-1$) được mô tả bằng chuỗi n bit¹

$$b = b_{n-1}b_{n-2} \dots b_1b_0 \in \mathbb{B}^n.$$

Ta nhận thấy qui ước này giống với qui ước số nhị phân ở Phần 2.1.3 nên để thuận tiện ta mã trạng thái cơ bản b bằng số nguyên $[b]$. Ta cũng sẽ dùng lẫn lộn trạng thái b (xem như chuỗi bit b), số nhị phân b và số nguyên $[b]$.

Tương tự như trường hợp 1 qubit, trạng thái của hệ n qubit là sự chồng chất của 2^n trạng thái cơ bản. Cụ thể, nếu đồng nhất trạng thái cơ bản $b = b_{n-1} \dots b_1b_0$ với ket thứ $[b]$ (tính từ 0) của cơ sở chuẩn tắc của \mathbb{C}^{2^n} thì trạng thái $|\psi\rangle$ của hệ n qubit

¹thứ tự này được gọi là **big-endian**, ngược với một lựa chọn khác là little-endian.

được mô tả là một tổ hợp tuyến tính

$$|\psi\rangle = \sum_{b \in \{0,1\}^n} \alpha_b |b\rangle = \sum_{k=0}^{2^n-1} \alpha_k |k\rangle \in \mathbb{C}^{2^n}.$$

Để thuận tiện tính toán, ta cũng yêu cầu $|\psi\rangle$ phải được chuẩn hóa

$$\| |\psi\rangle \| = 1 \iff \sum_{k=0}^{2^n-1} |\alpha_k|^2 = 1.$$

Cơ sở chuẩn tắc của \mathbb{C}^{2^n} cũng được gọi là cơ sở tính toán.

Nếu ta đồng nhất 2 trạng thái cơ bản 0, 1 của từng qubit với 2 ket

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \in \mathbb{C}^2$$

thì với từng trạng thái cơ bản của hệ $b = b_{n-1} \dots b_1 b_0$, ta có (♣)

$$|b_{n-1}\rangle \otimes \dots \otimes |b_1\rangle \otimes |b_0\rangle$$

chính là vector thứ $[b]$ của cơ sở chuẩn tắc của \mathbb{C}^{2^n} . Hơn nữa, theo cách viết gọn của kí pháp Dirac thì

$$|b_{n-1}\rangle \otimes \dots \otimes |b_1\rangle \otimes |b_0\rangle = |b_{n-1} \dots b_1 b_0\rangle = |b\rangle.$$

Như vậy, với $b = b_{n-1} \dots b_1 b_0, k = [b]$, ta có thể dùng lẫn lộn các kí hiệu

$$|k\rangle, \quad |b_{n-1} \dots b_1 b_0\rangle, \quad |b_{n-1}\rangle \otimes \dots \otimes |b_1\rangle \otimes |b_0\rangle.$$

Cho hệ n qubit, nếu tất cả các qubit của hệ đều ở trạng thái $|\psi\rangle \in \mathbb{C}^2$ thì trạng thái của cả hệ được kí hiệu là

$$|\psi\rangle^{\otimes n} = \underbrace{|\psi\rangle \otimes |\psi\rangle \otimes \dots \otimes |\psi\rangle}_{n \text{ lần}} = \underbrace{|\psi\rangle |\psi\rangle \dots |\psi\rangle}_{n \text{ lần}} \in \mathbb{C}^{2^n}.$$

Chẳng hạn, trạng thái cơ bản ứng với vector đầu tiên của cơ sở chuẩn tắc là

$$|0\rangle^{\otimes n} = \underbrace{|0\rangle |0\rangle \dots |0\rangle}_{n \text{ lần}} = \underbrace{|00 \dots 0\rangle}_{n \text{ lần}} = |0\rangle_n.$$

Đặc biệt (♣)

$$|+\rangle^{\otimes n} = \sum_{k=0}^{2^n-1} \frac{1}{\sqrt{2^n}} |k\rangle$$

mô tả trạng thái “tổ hợp đều” các trạng thái cơ bản.

Ví dụ 4.1.1. Xét hệ $n = 2$ qubit được đánh số là qubit 0, qubit 1. Hệ này có $2^n = 4$ trạng thái cơ bản ứng với các cặp trạng thái cơ bản của qubit 1, qubit 0 là $(0, 0)$, $(0, 1)$, $(1, 0)$, $(1, 1)$ mà được kí hiệu bằng các chuỗi 2 bit là 00, 01, 10, 11. Các chuỗi này cũng được đánh số bằng số nguyên mà chúng biểu diễn là 0, 1, 2, 3.

Các trạng thái cơ bản 00, 01, 10, 11 được đồng nhất lần lượt với các vector của cơ sở chuẩn tắc của $\mathbb{C}^{2^n} = \mathbb{C}^4$ là

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

mà các vector này cũng lần lượt là các tích tensor $|0\rangle|0\rangle, |0\rangle|1\rangle, |1\rangle|0\rangle, |1\rangle|1\rangle$. Dùng số thứ tự, các vector này cũng lần lượt được kí hiệu là $|0\rangle_2, |1\rangle_2, |2\rangle_2, |3\rangle_2$.²

Trạng thái của hệ 2 qubit được mô tả là

$$\begin{aligned} |\psi\rangle &= \sum_{b \in \{0,1\}^2} \alpha_b |b\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \\ &= \sum_{k=0}^3 \alpha_k |k\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle + \alpha_2|2\rangle + \alpha_3|3\rangle \\ &= \alpha_0 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \alpha_1 \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \alpha_3 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} \in \mathbb{C}^4. \end{aligned}$$

Hơn nữa $|\psi\rangle$ là vector đơn vị: $\sum_{k=0}^3 |\alpha_k|^2 = 1$.

Xét trạng thái sau của hệ 2 qubit

$$|\psi\rangle = \sum_{k=0}^3 \frac{1}{2} |k\rangle = \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Nhận xét, $|\psi\rangle = |+\rangle|+\rangle$ (\clubsuit). Ket $|\psi\rangle$ mô tả trạng thái “tổ hợp đều” 4 trạng thái cơ bản. Ngược lại, ket

$$|\phi\rangle = \frac{\sqrt{3}}{2}|01\rangle + \frac{1}{2}|10\rangle = \begin{bmatrix} 0 & \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \end{bmatrix}^T$$

mô tả trạng thái “thiên về” 01. □

²chỉ số 2 ở dưới kí hiệu cho số qubit $n = 2$.

Với hệ 2 qubit, ngoài các trạng thái cơ bản ứng với cơ sở chuẩn tắc là $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ thì các trạng thái sau cũng hay được dùng, gọi là các **trạng thái Bell** (Bell state)

$$\begin{aligned} |\Phi^+\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}, & |\Phi^-\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ -\frac{1}{\sqrt{2}} \end{bmatrix}, \\ |\Psi^+\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) = \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}, & |\Psi^-\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) = \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}. \end{aligned}$$

Các trạng thái này tạo thành một cơ sở trực chuẩn của \mathbb{C}^4 (\clubsuit) được gọi là **cơ sở Bell** (Bell basis).

4.1.2 Vướng lượng tử

Nếu trạng thái $|\psi\rangle \in \mathbb{C}^{2^n}$ của hệ n qubit có thể được viết thành tích tensor của các trạng thái $|\psi_i\rangle \in \mathbb{C}^2$ của qubit i ($i = 0, \dots, n-1$)

$$|\psi\rangle = |\psi_{n-1}\rangle |\psi_{n-2}\rangle \cdots |\psi_0\rangle$$

thì $|\psi\rangle$ được gọi là **trạng thái tích** (product state) hay **trạng thái tách được** (separable state). Ngược lại, $|\psi\rangle$ được gọi là **trạng thái vướng (lượng tử)** (entangled state).

Khi hệ ở trạng thái vướng, các qubit “vướng víu nhau” một cách “thật sự rối rắm” nên không thể phân tách ra thành các trạng thái riêng lẻ của từng qubit. Các trạng thái Bell $|\Phi^+\rangle, |\Phi^-\rangle, |\Psi^+\rangle, |\Psi^-\rangle$ là các trạng thái vướng điển hình của hệ 2 qubit.

Ví dụ 4.1.2. (tiếp Ví dụ 4.1.1) Rõ ràng các trạng thái cơ bản $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ là các trạng thái tách được của hệ 2 qubit. Hơn nữa, trạng thái “tổ hợp đều”

$$|\psi\rangle = \sum_{k=0}^3 \frac{1}{2} |k\rangle = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

cũng tách được vì $|\psi\rangle = |+\rangle|+\rangle$.

Khó thấy hơn, trạng thái sau cũng tách được

$$|\phi\rangle = \frac{1}{2\sqrt{2}} (\sqrt{3}|00\rangle - \sqrt{3}|01\rangle + |10\rangle - |11\rangle).$$

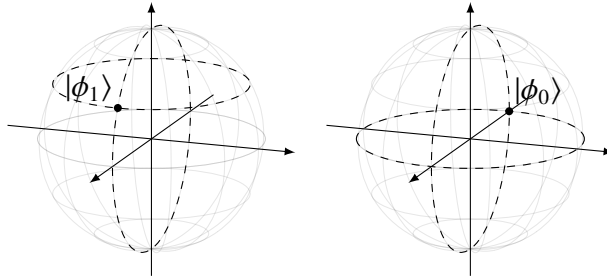
Thật vậy

$$\begin{aligned}
 |\phi\rangle &= \frac{1}{2\sqrt{2}} (\sqrt{3}|00\rangle - \sqrt{3}|01\rangle + |10\rangle - |11\rangle) \\
 &= \frac{1}{2\sqrt{2}} (\sqrt{3}|0\rangle|0\rangle - \sqrt{3}|0\rangle|1\rangle + |1\rangle|0\rangle - |1\rangle|1\rangle) \\
 &= \frac{1}{2\sqrt{2}} (\sqrt{3}|0\rangle(|0\rangle - |1\rangle) + |1\rangle(|0\rangle - |1\rangle)) = \frac{1}{2\sqrt{2}} (\sqrt{3}|0\rangle + |1\rangle)(|0\rangle - |1\rangle) \\
 &= \left(\frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle \right) \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \right) = |\phi_1\rangle|\phi_0\rangle.
 \end{aligned}$$

Như vậy $|\phi\rangle$ có thể được tách thành các trạng thái riêng lẻ của qubit 0 là $|\phi_0\rangle$ và qubit 1 là $|\phi_1\rangle$. Hơn nữa, có thể biểu diễn $|\phi_0\rangle, |\phi_1\rangle$ trên mặt cầu Bloch (♣)

$$|\phi_0\rangle = \cos \frac{\pi}{4}|0\rangle + e^{-i\pi} \sin \frac{\pi}{4}|1\rangle, \quad |\phi_1\rangle = \cos \frac{\pi}{6}|0\rangle + e^{0i} \sin \frac{\pi}{6}|1\rangle.$$

□



Với hệ 3 qubit, các trạng thái sau hay được dùng

- **Trạng thái GHZ** (Greenberger–Horne–Zeilinger state)

$$|\text{GHZ}\rangle = \frac{1}{\sqrt{2}}|000\rangle + \frac{1}{\sqrt{2}}|111\rangle.$$

- **Trạng thái W** (Wolfgang Dür state)

$$|\text{W}\rangle = \frac{1}{\sqrt{3}}|001\rangle + \frac{1}{\sqrt{3}}|010\rangle + \frac{1}{\sqrt{3}}|100\rangle.$$

Các trạng thái này đều là các trạng thái vướng. (♣)

4.1.3 Phép đo theo một cơ sở trực chuẩn

Tương tự trường hợp 1 qubit, một hệ n qubit ở trạng thái

$$|\psi\rangle = \sum_{b \in \{0,1\}^n} \alpha_b |b\rangle$$

khi được thực hiện phép đo theo cơ sở tính toán sẽ được một chuỗi n bit b với xác suất $|\alpha_b|^2$ và ngay sau khi đo thì hệ chuyển sang trạng thái $|b\rangle$. Các hệ số α_b được gọi là các amplitude để phân biệt với $|\alpha_b|^2$ là các xác suất. Nhận xét, $\sum_{b \in \{0,1\}^n} |\alpha_b|^2 = 1$ như yêu cầu của xác suất vì việc đo chỉ cho ra một chuỗi n bit.

Kết quả đo $b = b_{n-1} \dots b_1 b_0$ nghĩa là qubit 0 được bit b_0 , qubit 1 được bit b_1 , ... và trạng thái của hệ sụp đổ thành $|b\rangle = |b_{n-1}\rangle \dots |b_1\rangle |b_0\rangle$ nghĩa là qubit 0 sụp đổ thành $|b_0\rangle$, qubit 1 sụp đổ thành $|b_1\rangle$, ... Như vậy, phép đo này đo đồng thời tất cả các qubit.

Tổng quát, cho $B = \{|a_k\rangle, k = 0, \dots, 2^n - 1\}$ là một cơ sở trực chuẩn của \mathbb{C}^{2^n} , trạng thái $|\psi\rangle$ của hệ n qubit có thể được viết theo cơ sở B là

$$|\psi\rangle = \sum_{k=0}^{2^n-1} \langle a_k | \psi \rangle |a_k\rangle.$$

Khi đó, phép đo (đồng thời) theo cơ sở B sẽ cho kết quả a_k (có thể gọi là kết quả k) với xác suất $|\langle a_k | \psi \rangle|^2$ và trạng thái của hệ sụp đổ thành $|a_k\rangle$. Lưu ý, $(\langle a_k | \psi \rangle, k = 0, \dots, 2^n - 1)$ là tọa độ của $|\psi\rangle$ theo cơ sở B .

Ví dụ 4.1.3. (tiếp Ví dụ 4.1.2) Xét hệ 2 qubit với trạng thái

$$|\psi\rangle = \sum_{k=0}^3 \frac{1}{2} |k\rangle = \frac{1}{2} |00\rangle + \frac{1}{2} |01\rangle + \frac{1}{2} |10\rangle + \frac{1}{2} |11\rangle.$$

Khi đo $|\psi\rangle$ theo cơ sở tính toán ta được 00, 01, 10, 11 với xác suất đều là $\left|\frac{1}{2}\right|^2 = 25\%$ nên $|\psi\rangle$ được gọi là “tổ hợp đều”. Nếu dùng số thứ tự ta cũng có thể nói là được 0, 1, 2, 3 với xác suất đều là 25%.

Ngược lại, theo cơ sở Bell, $|\psi\rangle$ được viết là

$$\begin{aligned} |\psi\rangle &= \langle \Phi^+ | \psi \rangle |\Phi^+\rangle + \langle \Phi^- | \psi \rangle |\Phi^-\rangle + \langle \Psi^+ | \psi \rangle |\Psi^+\rangle + \langle \Psi^- | \psi \rangle |\Psi^-\rangle \\ &= \frac{1}{\sqrt{2}} |\Phi^+\rangle + \frac{1}{\sqrt{2}} |\Psi^+\rangle \end{aligned}$$

nên khi đo theo cơ sở Bell ta được chỉ 1 trong 2 kết quả Φ^+, Ψ^+ với xác suất đều là 50%. Nếu dùng số thứ tự ($|\Phi^+\rangle$ là 0, $|\Phi^-\rangle$ là 1, $|\Psi^+\rangle$ là 2, $|\Psi^-\rangle$ là 3) ta cũng có thể nói là được 0, 2 với xác suất đều là 50% (không thể được 1, 3).

Khi đo trạng thái

$$|\phi\rangle = \frac{1}{2\sqrt{2}} (\sqrt{3}|00\rangle - \sqrt{3}|01\rangle + |10\rangle - |11\rangle)$$

theo cơ sở tính toán ta được: 00, 01 với cùng xác suất $\left|\frac{\sqrt{3}}{2\sqrt{2}}\right|^2 = \frac{3}{8}$ và 10, 11 với cùng xác suất $\left|\frac{1}{2\sqrt{2}}\right|^2 = \frac{1}{8}$.

Xét riêng qubit 0, ta được 0 với xác suất $\frac{3}{8} + \frac{1}{8} = 50\%$ (được 1 với xác suất 50%). Xét riêng qubit 1, ta được 0 với xác suất $\frac{3}{8} + \frac{3}{8} = 75\%$ (được 1 với xác suất 25%). \square

Cho hệ n qubit ở trạng thái $|\psi\rangle \in \mathbb{C}^{2^n}$, ta có thể tiến hành phép đo riêng lẻ trên các qubit, gọi là **phép đo riêng** (partial measurement). Chẳng hạn, để đo qubit 0 theo cơ sở tính toán, $|\psi\rangle$ có thể được viết lại theo dạng sau một cách duy nhất (\clubsuit)

$$|\psi\rangle = |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle, \quad |\psi_0\rangle, |\psi_1\rangle \in \mathbb{C}^{2^{(n-1)}}.$$

Khi đó, phép đo qubit 0 theo cơ sở tính toán sẽ cho 1 trong 2 kết quả

- qubit 0 được 0 với xác suất $\| |\psi_0\rangle \|^2$ và hệ sụp đổ trạng thái thành

$$\frac{1}{\| |\psi_0\rangle \|} |\psi_0\rangle |0\rangle,$$

- qubit 0 được 1 với xác suất $\| |\psi_1\rangle \|^2$ và hệ sụp đổ trạng thái thành

$$\frac{1}{\| |\psi_1\rangle \|} |\psi_1\rangle |1\rangle.$$

Lưu ý, $\| |\psi_0\rangle \|^2 + \| |\psi_1\rangle \|^2 = 1$ (\clubsuit). Từ trên, ta cũng thấy, qubit 0 phân tách với các qubit khác trong hệ khi và chỉ khi dù đo qubit 0 ra gì (0 hay 1) thì trạng thái của các qubit còn lại là như nhau ($|\psi_0\rangle = |\psi_1\rangle$), nói cách khác, việc đo qubit 0 không ảnh hưởng đến trạng thái của các qubit còn lại.

Ví dụ 4.1.4. (tiếp Ví dụ 4.1.3) Xét hệ 2 qubit với trạng thái

$$|\phi\rangle = \frac{1}{2\sqrt{2}} (\sqrt{3}|00\rangle - \sqrt{3}|01\rangle + |10\rangle - |11\rangle).$$

Để khảo sát phép đo riêng qubit 0 theo cơ sở tính toán, ta viết lại

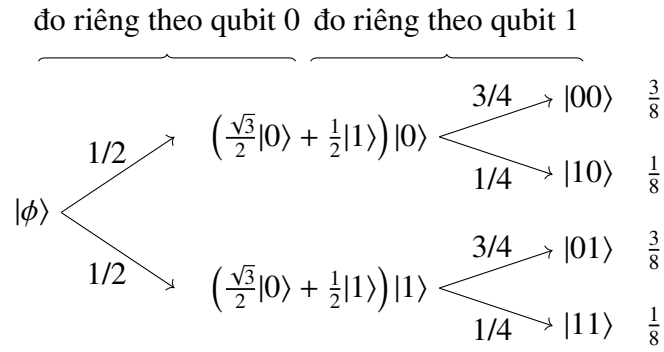
$$|\phi\rangle = \frac{1}{2\sqrt{2}} (\sqrt{3}|0\rangle + |1\rangle)|0\rangle + \frac{1}{2\sqrt{2}} (-\sqrt{3}|0\rangle - |1\rangle)|1\rangle.$$

Do đó ta có qubit 0 được 0 với xác suất $\left| \frac{1}{2\sqrt{2}}(\sqrt{3}|0\rangle + |1\rangle) \right|^2 = 50\%$ và sụp đổ thành $\left(\frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle \right)|0\rangle$; được 1 với xác suất 50% và sụp đổ thành $\left(-\frac{\sqrt{3}}{2}|0\rangle - \frac{1}{2}|1\rangle \right)|1\rangle \equiv \left(\frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle \right)|1\rangle$.

Xác suất được 0 (hay 1) của qubit 0 khi đo riêng qubit 0 khớp với kết quả đo đồng thời cả 2 qubit ở Ví dụ 4.1.3. Hơn nữa, trong trường hợp này, không phụ thuộc kết quả đo qubit 0, qubit 1 luôn có trạng thái $\frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle$. Như vậy, qubit 0 phân tách với qubit 1 trong trạng thái $|\phi\rangle$ như ở Ví dụ 4.1.2 ta đã thấy

$$|\phi\rangle = \left(\frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle \right) \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \right) = |\phi_1\rangle|\phi_0\rangle.$$

Sơ đồ sau khảo sát phép đo $|\phi\rangle$ riêng theo qubit 0 rồi đến qubit 1



Ta thấy kết quả chung cuộc giống với kết quả đo đồng thời cả 2 qubit. Tương tự, nếu đo riêng theo qubit 1 rồi đến qubit 0 cũng cho cùng kết quả. \square

4.1.4 Phép đo chiếu *

Ma trận vuông $P \in \mathbb{C}^{N \times N}$ được gọi là **ma trận chiếu** (projection matrix)³ nếu

$$\begin{cases} P^2 = P, \\ P^\dagger = P. \end{cases}$$

Trường hợp đơn giản nhất chính là ma trận chiếu (1.6) lên vector đơn vị u

$$P_u = |u\rangle\langle u|.$$

³kĩ hơn nên gọi là chiếu trực giao (orthogonal projection), tuy nhiên, thông thường chiếu được hiểu là chiếu trực giao (vuông góc).

Ta có

$$P_u^2 = (|u\rangle\langle u|)(|u\rangle\langle u|) = |u\rangle \underbrace{\langle u|u\rangle}_1 \langle u| = |u\rangle\langle u| = P_u,$$

$$P_u^\dagger = (|u\rangle\langle u|)^\dagger = (\langle u|)^\dagger (|u\rangle)^\dagger = |u\rangle\langle u| = P_u.$$

Tổng quát hơn, cho $\{|\psi_1\rangle, \dots, |\psi_m\rangle\}$ là một họ trực chuẩn trong \mathbb{C}^N (tập các vector đơn vị đôi một trực giao) thì ma trận sau là ma trận chiếu (Bài tập 4.7)

$$P = \sum_{k=1}^m |\psi_k\rangle\langle\psi_k|. \quad (4.1)$$

Hơn nữa, mọi ma trận chiếu $P \in \mathbb{C}^{N \times N}$ đều có thể được viết ở dạng (4.1) với họ $\{|\psi_1\rangle, \dots, |\psi_m\rangle\}$ trực chuẩn nào đó của \mathbb{C}^N .

Phép đo theo cơ sở trực chuẩn có thể được mở rộng thành **phép đo chiếu** (projective measurement). Cụ thể, cho họ các ma trận chiếu $\{P_0, \dots, P_{m-1}\}$ “đầy đủ” (complete) trong \mathbb{C}^N , nghĩa là

$$\sum_{k=0}^{m-1} P_k = I_N.$$

Khi hệ lượng tử có trạng thái $|\psi\rangle \in \mathbb{C}^N$ được đo theo họ ma trận chiếu trên thì

- kết quả đo được $k \in \{0, \dots, m-1\}$ với xác suất

$$\|P_k |\psi\rangle\|^2 = \langle\psi|P_k^\dagger P_k|\psi\rangle = \langle\psi|P_k P_k|\psi\rangle = \langle\psi|P_k|\psi\rangle,$$

- và hệ sụp đổ thành trạng thái tương ứng là

$$\frac{P_k |\psi\rangle}{\|P_k |\psi\rangle\|}.$$

Chẳng hạn, với hệ n qubit, $N = 2^n$, phép đo theo cơ sở trực chuẩn $\{|\psi_0\rangle, \dots, |\psi_{N-1}\rangle\}$ chính là phép đo chiếu theo họ ma trận chiếu $\{|\psi_k\rangle\langle\psi_k| : k = 0, \dots, N-1\}$.

Phép đo riêng cũng là một trường hợp của phép đo chiếu. Cụ thể, cho hệ n qubit ở trạng thái $|\psi\rangle \in \mathbb{C}^{2^n}$, phép đo riêng qubit 0 theo cơ sở tính toán có thể được xem là phép đo chiếu theo họ ma trận chiếu sau (Bài tập 4.8)

$$\{I_{2^{n-1}} \otimes (|0\rangle\langle 0|), I_{2^{n-1}} \otimes (|1\rangle\langle 1|)\}.$$

Cho 2 trạng thái $|\phi\rangle, |\psi\rangle$ trực giao, tức là $\langle\phi|\psi\rangle = 0$, dùng phép đo chiếu theo họ 2 ma trận chiếu $\{|\phi\rangle\langle\phi|, I - |\phi\rangle\langle\phi|\}$ ta có thể phân biệt hoàn hảo 2 trạng thái này vì khi đo $|\phi\rangle$ ta chắc chắn được 0 còn đo $|\psi\rangle$ ta chắc chắn được 1. Thật vậy

$$\begin{cases} \| |\phi\rangle\langle\phi| |\phi\rangle \|^2 = \| \underbrace{\langle\phi|\phi\rangle}_1 |\phi\rangle \|^2 = \| |\phi\rangle \|^2 = 1, \\ \| (I - |\phi\rangle\langle\phi|) |\psi\rangle \|^2 = \| |\psi\rangle - \underbrace{\langle\phi|\psi\rangle}_{0} |\phi\rangle \|^2 = \| |\psi\rangle \|^2 = 1. \end{cases}$$

Với phép đo chiếu theo họ các ma trận chiếu $\{P_0, \dots, P_{m-1}\}$, nếu ta muốn kết quả đo ứng với P_k là số thực α_k thì phép đo chiếu có thể được đặc tả bằng ma trận

$$M = \sum_{k=0}^{m-1} \alpha_k P_k.$$

Lưu ý, M là ma trận Hermite. Ngược lại mọi ma trận Hermite M đều có thể mô tả một phép đo chiếu như trên bằng cách dùng phân rã phổ của M (Bài tập 4.9). Như vậy phép đo chiếu được xác định bởi ma trận Hermite và ngược lại. Do đó, các ma trận Hermite thường được gọi là các **quan sát** (observable).

Ví dụ 4.1.5. Với hệ 1 qubit, các ma trận Pauli X, Y, Z vừa là các cổng lượng tử (ma trận unita) vừa là các quan sát (ma trận Hermite). Chẳng hạn

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = |0\rangle\langle 0| - |1\rangle\langle 1| = 1|0\rangle\langle 0| + (-1)|1\rangle\langle 1|.$$

Như vậy Z xác định phép đo chiếu mà $|0\rangle$ đo được giá trị 1 còn $|1\rangle$ đo được giá trị -1. Đây chính là phép đo theo cơ sở tính toán nhưng có giá trị đo khác (với phép đo theo cơ sở tính toán, $|0\rangle$ đo được giá trị 0 còn $|1\rangle$ đo được giá trị 1). \square

Ví dụ 4.1.6. Xét hệ 2 qubit và phép đo chiếu “chẵn-lẻ” dùng họ gồm 2 ma trận chiếu sau

$$P_0 = |00\rangle\langle 00| + |11\rangle\langle 11|, \quad P_1 = |01\rangle\langle 01| + |10\rangle\langle 10|.$$

Phép đo này giúp phân biệt hoàn hảo các chuỗi 2 bit có parity 0 (số lượng bit 1 là chẵn) là 00 và 11 với các chuỗi 2 bit có parity 1 (số lượng bit 1 là lẻ) là 01 và 10.

Nếu muốn kết quả đo ứng với P_0 là 0 và P_1 là 1 (chẳng hạn, qui ước 0 là chẵn, 1 là lẻ) thì phép đo này có thể được đặc tả bằng ma trận Hermite sau

$$M = 0P_0 + 1P_1 = P_1 = |01\rangle\langle 01| + |10\rangle\langle 10| = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

\square

4.2 Phép toán và mạch lượng tử

Tương tự trường hợp 1 qubit, phép toán lượng tử trên hệ n qubit là toán tử tuyến tính bảo toàn chuẩn trên \mathbb{C}^{2^n} . Phép toán thường được đồng nhất với ma trận biểu diễn nó là ma trận unita kích thước $2^n \times 2^n$.

Trường hợp đơn giản, phép toán lượng tử trên hệ n qubit là tích tensor các phép toán lượng tử 1 qubit. Chẳng hạn, xét hệ 2 qubit đang có trạng thái $|\psi\rangle$, nếu ta tác động U_0 lên qubit 0 và U_1 lên qubit 1 thì hệ sẽ chuyển sang trạng thái (lưu ý qui ước big-endian)

$$|\phi\rangle = (U_1 \otimes U_0)|\psi\rangle.$$

Đơn giản hơn nữa, nếu $|\psi\rangle$ tách được $|\psi\rangle = |\psi_1\rangle|\psi_0\rangle$ thì

$$|\phi\rangle = (U_1 \otimes U_0)(|\psi_1\rangle \otimes |\psi_0\rangle) = U_1|\psi_1\rangle \otimes U_0|\psi_0\rangle = U_1|\psi_1\rangle U_0|\psi_0\rangle.$$

Khi đó, $|\phi\rangle$ cũng tách được.

Biến đổi $|\phi\rangle = (U_1 \otimes U_0)|\psi\rangle$ được mô tả bằng sơ đồ mạch

$$|\psi\rangle \left\{ \begin{array}{c} \text{---} \boxed{U_0} \text{---} \\ \text{---} \boxed{U_1} \text{---} \end{array} \right\} |\phi\rangle$$

Ta qui ước các đường ứng với các qubit được vẽ từ trên xuống (đường trên cùng ứng với qubit 0, ..., đường dưới cùng ứng với qubit $n - 1$ của hệ). Cặp ngoặc nhọn nối các đường để chỉ các qubit có thể ở trạng thái vướng. Trường hợp đầu vào $|\psi\rangle$ tách được, $|\psi\rangle = |\psi_1\rangle|\psi_0\rangle$, sơ đồ mạch sau mô tả rõ ràng hơn

$$\begin{array}{ccc} |\psi_0\rangle & \text{---} \boxed{U_0} \text{---} & |\phi_0\rangle \\ |\psi_1\rangle & \text{---} \boxed{U_1} \text{---} & |\phi_1\rangle \end{array}$$

Đầu ra $|\phi\rangle = |\phi_1\rangle|\phi_0\rangle$.

Nếu một qubit nào đó trong hệ không bị tác động, ta có thể xem như nó bị tác động bởi cổng đơn vị I (cổng NOOP). Chẳng hạn, mạch sau

$$|\psi\rangle \left\{ \begin{array}{c} \text{---} \boxed{U_0} \text{---} \\ \text{---} \boxed{I} \text{---} \end{array} \right\} |\phi\rangle$$

mô tả tác động U_0 lên qubit 0 mà không làm gì với qubit 1. Dĩ nhiên, mạch trên có thể được mô tả đơn giản hơn là

$$|\psi\rangle \left\{ \begin{array}{c} \boxed{U_0} \\ \hline \end{array} \right\} |\phi\rangle$$

Trường hợp ta tác động U lên tất cả n qubit của hệ thì tác động lên hệ là

$$U^{\otimes n} = \underbrace{U \otimes U \otimes \dots \otimes U}_{n \text{ lần}} \in \mathbb{C}^{2^n \times 2^n}.$$

Ví dụ 4.2.1. Mạch 2 qubit

$$\begin{array}{c} |0\rangle \\ |0\rangle \end{array} \begin{array}{c} \boxed{H} \\ \boxed{H} \end{array} \left\{ \begin{array}{c} \\ \\ \end{array} \right\} |\phi\rangle$$

cho đầu ra là trạng thái (tách được)

$$|\phi\rangle = H|0\rangle H|0\rangle = |+\rangle|+\rangle = \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle.$$

Tổng quát, với hệ n qubit, đầu ra là trạng thái “tổ hợp đều”

$$H^{\otimes n}|0\rangle^{\otimes n} = (H|0\rangle)^{\otimes n} = |+\rangle^{\otimes n} = \sum_{k=0}^{2^n-1} \frac{1}{\sqrt{2^n}} |k\rangle.$$

□

Ví dụ 4.2.2. Xét mạch 2 qubit

$$|\Phi^+\rangle \left\{ \begin{array}{c} \boxed{X} \\ \hline \end{array} \right\} |\chi\rangle$$

Mạch này mô tả hệ 2 qubit ở trạng thái vướng $|\Phi^+\rangle$, nhận tác động cổng X lên qubit 0 (không tác động gì đến qubit 1), chuyển sang trạng thái

$$\begin{aligned} |\chi\rangle &= (I \otimes X)|\Phi^+\rangle = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}^T \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = |\Psi^+\rangle. \end{aligned}$$

□

4.3 Các cổng nhiều qubit thông dụng

Tương tự trường hợp 1 qubit, các phép toán lượng tử cơ bản thường được gọi là cổng lượng tử. Phần này trình bày các cổng 2 và 3 qubit thông dụng.

4.3.1 Cổng CNOT

Cổng CNOT (CNOT gate, controlled-NOT, CX, controlled-X) là cổng 2 qubit được xác định bởi

$$\begin{aligned} \text{CNOT}|00\rangle &= |00\rangle, & \text{CNOT}|01\rangle &= |01\rangle, \\ \text{CNOT}|10\rangle &= |11\rangle, & \text{CNOT}|11\rangle &= |10\rangle, \end{aligned}$$

mà có thể được mô tả gọn hơn là

$$\text{CNOT}|a\rangle|b\rangle = |a\rangle|a \oplus b\rangle, \quad a, b \in \{0, 1\}.$$

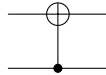
CNOT là phiên bản lượng tử của cổng XOR cổ điển.

Ta cũng thấy

- nếu qubit 1 là $|0\rangle$ thì $\text{CNOT}|0\rangle|b\rangle = |0\rangle|b\rangle$,
- nếu qubit 1 là $|1\rangle$ thì $\text{CNOT}|1\rangle|b\rangle = |1\rangle|1 \oplus b\rangle = |1\rangle|\text{NOT}b\rangle$.

Như vậy, qubit 1 điều khiển thao tác NOT trên qubit 0.

Cổng CNOT thường được vẽ trên mạch là



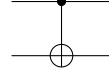
Dấu chấm cho biết **qubit điều khiển** (control qubit), ở đây là qubit 1; dấu cộng trong vòng tròn cho biết **qubit mục tiêu** (target qubit), ở đây là qubit 0.

CNOT có ma trận biểu diễn và tác động lên $|\psi\rangle = \sum_{k=0}^3 \alpha_k |k\rangle$ là

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \text{CNOT} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_3 \\ \alpha_2 \end{bmatrix}.$$

Tổng quát, ta kí hiệu CNOT_{ij} là cổng CNOT với qubit điều khiển i và qubit mục tiêu j . Như vậy, CNOT chính là CNOT_{10} .

Cổng CNOT_{01}



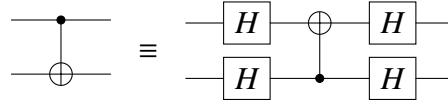
có ma trận biểu diễn và tác động lên $|\psi\rangle = \sum_{k=0}^3 \alpha_k |k\rangle$ là

$$\text{CNOT}_{01} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \text{CNOT}_{01} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \alpha_3 \\ \alpha_2 \\ \alpha_1 \end{bmatrix}.$$

Nhận xét (♣)

$$\text{CNOT}_{01} = (H \otimes H) \text{CNOT} (H \otimes H).$$

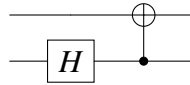
Do đó



tức là, ta có thể “cài đặt” hay “thi công” hay “hiện thực” CNOT_{01} bằng các cổng H và CNOT .

(Lưu ý, nếu chỉ được phép dùng các cổng H và CNOT thì mạch trên có thể được coi là một “thuật toán lượng tử” giúp thực hiện tính toán ta cần là CNOT_{01} bằng các tính toán ta có là H và CNOT . Hơn nữa, ý tưởng của thuật toán này là “ CNOT_{01} trong cơ sở tính toán (cơ sở Z) chính là CNOT_{10} trong cơ sở X !”)

Ví dụ 4.3.1. Mạch sau có vai trò rất quan trọng trong tính toán lượng tử



Nếu đầu vào là $|00\rangle$ thì đầu ra là

$$\begin{aligned} \text{CNOT}(H|0\rangle \otimes |0\rangle) &= \text{CNOT}(|+\rangle|0\rangle) = \text{CNOT} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle \\ &= \frac{1}{\sqrt{2}} \text{CNOT}(|00\rangle + |10\rangle) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = |\Phi^+\rangle. \end{aligned}$$

Mạch này chuyển cơ sở tính toán gồm các trạng thái tách được sang cơ sở Bell gồm các trạng thái vướng. (♣) \square

4.3.2 Cổng điều khiển

Mở rộng cổng CNOT, với U là cổng 1 qubit bất kỳ, **cổng điều khiển- U** (controlled- U gate) là cổng 2 qubit được xác định bởi

$$CU|00\rangle = |00\rangle, \quad CU|01\rangle = |01\rangle,$$

$$CU|10\rangle = |1\rangle U|0\rangle, \quad CU|11\rangle = |1\rangle U|1\rangle.$$

CU tác động U lên qubit 0 nếu qubit 1 là 1 (không làm gì nếu qubit 1 là 0). Nhận xét, CNOT chính là CX .

Nếu U có ma trận biểu diễn là

$$U = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \in \mathbb{C}^{2 \times 2}$$

thì

$$U|0\rangle = a|0\rangle + c|1\rangle, \quad U|1\rangle = b|0\rangle + d|1\rangle.$$

Khi đó

$$CU|00\rangle = |00\rangle, \quad CU|01\rangle = |01\rangle,$$

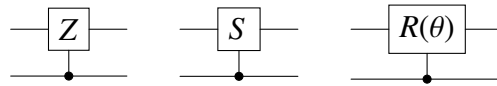
$$CU|10\rangle = |1\rangle U|0\rangle = |1\rangle(a|0\rangle + c|1\rangle) = a|10\rangle + c|11\rangle.$$

$$CU|11\rangle = |1\rangle U|1\rangle = |1\rangle(b|0\rangle + d|1\rangle) = b|10\rangle + d|11\rangle.$$

Do đó CU có ma trận biểu diễn là

$$CU = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & c & d \end{bmatrix}.$$

Ngoài CX thì các cổng điều khiển khác cũng hay được dùng là CZ , CS và $CR(\theta)$



4.3.3 Cổng SWAP

Cổng SWAP (SWAP gate) là cổng 2 qubit được xác định bởi

$$\text{SWAP}|00\rangle = |00\rangle, \quad \text{SWAP}|01\rangle = |10\rangle,$$

$$\text{SWAP}|10\rangle = |01\rangle, \quad \text{SWAP}|11\rangle = |11\rangle,$$

mà có thể được mô tả gọn hơn là

$$\text{SWAP}|a\rangle|b\rangle = |b\rangle|a\rangle, a, b \in \{0, 1\}.$$

SWAP hoán đổi trạng thái 2 qubit vì với

$$|\psi\rangle = a|0\rangle + b|1\rangle, \quad |\phi\rangle = c|0\rangle + d|1\rangle \in \mathbb{C}^2$$

ta có

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

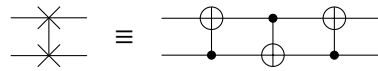
Cổng SWAP thường được vẽ trên mạch là



SWAP có ma trận biểu diễn và tác động lên $|\psi\rangle = \sum_{k=0}^3 \alpha_k |k\rangle$ là

$$\text{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{SWAP} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \alpha_2 \\ \alpha_1 \\ \alpha_3 \end{bmatrix}.$$

SWAP có thể được “thi công” bằng 3 cổng CNOT qua đẳng thức mạch sau (♣)



4.3.4 Cổng Toffoli

Cổng Toffoli (Toffoli gate) là cổng 3 qubit được xác định bởi

$$\text{Toffoli}|a\rangle|b\rangle|c\rangle = |a\rangle|b\rangle|ab \oplus c\rangle, a, b, c \in \{0, 1\}.$$

Toffoli là phiên bản lượng tử của cổng AND hoặc NAND cổ điển vì

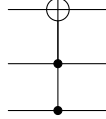
$$\begin{aligned} \text{Toffoli}|a\rangle|b\rangle|0\rangle &= |a\rangle|b\rangle|ab\rangle, a, b \in \{0, 1\}, \\ \text{Toffoli}|a\rangle|b\rangle|1\rangle &= |a\rangle|b\rangle|\text{NOT}ab\rangle, a, b \in \{0, 1\}. \end{aligned}$$

Toffoli còn được gọi là cổng CCNOT hoặc CCX vì

- $\text{Toffoli}|1\rangle|1\rangle|c\rangle = |1\rangle|1\rangle|\text{NOT}c\rangle$,
- $\text{Toffoli}|a\rangle|b\rangle|c\rangle = |a\rangle|b\rangle|c\rangle$, $a, b, c \in \{0, 1\}$, $ab \neq 1$.

Lưu ý, qubit 0 là qubit mục tiêu còn qubit 1, 2 là qubit điều khiển.

Cổng Toffoli thường được vẽ trên mạch là



Toffoli có ma trận biểu diễn và tác động lên $|\psi\rangle = \sum_{k=0}^7 \alpha_k |k\rangle$ là

$$\text{Toffoli} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad \text{Toffoli} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_7 \\ \alpha_6 \end{bmatrix}.$$

Cổng Toffoli được dùng khá phổ biến trong tính toán lượng tử và có thể được cài đặt bằng các cổng cơ bản hơn (Bài tập 4.24).

4.3.5 Tập cổng lượng tử toàn năng

Một tập (“nhỏ”) các cổng lượng tử cho phép “xấp xỉ” mọi phép toán lượng tử trên hệ có số lượng qubit bất kỳ với độ chính xác tùy ý được gọi là một **tập cổng lượng tử toàn năng** (universal quantum gate set). Các tập cổng lượng tử toàn năng điển hình là⁴

- $\{\text{CNOT và tất cả các cổng một qubit}\}$,
- $\{\text{CNOT}, H, T\}$,
- $\{\text{CNOT}, R_{\frac{\pi}{8}}, S\}$,
- $\{\text{Toffoli}, H, S\}$,
- $\{\text{Toffoli}, H\}$,
- $\{CH\}$.

⁴định nghĩa chính xác và các chứng minh nằm ngoài phạm vi của tài liệu này, xem thêm [4].

Nói chung, ta xem các cổng lượng tử sau đây là cơ bản

$$I, X, Y, Z, H, S, S^\dagger, T, T^\dagger, \text{CNOT}.$$

CNOT là cổng 2 qubit, các cổng còn lại là 1 qubit. H cho phép tạo tổ hợp “chồng chất bit” ($H|0\rangle = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$). CNOT giúp tạo trạng thái vướng (CNOT $|+\rangle|0\rangle = |\Phi^+\rangle$). S, T giúp có “amplitude phức”. Lưu ý, ngoại trừ S, T , các cổng còn lại đều có nghịch đảo là chính nó.

Các phép toán lượng tử khác cần được “cài đặt” bằng mạch chỉ dùng các cổng cơ bản trên. Hơn nữa, các qubit được “khởi động” với trạng thái $|0\rangle$ và chỉ được phép đo riêng lẻ từng qubit theo cơ sở tính toán.

4.4 Chuyển mạch logic thành mạch lượng tử

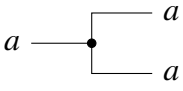
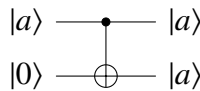
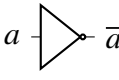
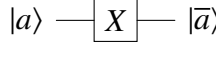
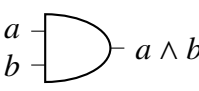
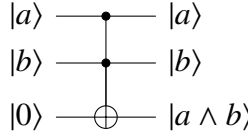
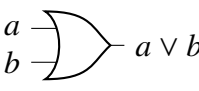
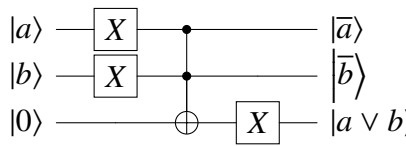
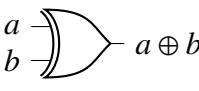
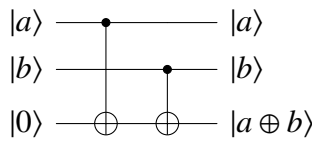
Tính toán cổ điển đã được nghiên cứu phát triển và triển khai thực tế từ lâu. Để khai thác nguồn di sản khổng lồ này, trong nhiều trường hợp, ta cần chuyển các mạch logic (là mạch thực hiện các tính toán cổ điển) thành mạch lượng tử tương ứng (là mạch thực hiện các tính toán lượng tử). Rất may, tính toán lượng tử là một mở rộng tự nhiên của tính toán cổ điển còn tính toán cổ điển là một trường hợp đặc biệt của tính toán lượng tử nên việc này không khó.

4.4.1 Phiên bản lượng tử của các cổng logic thông dụng

Ta đã biết cổng Toffoli là cổng toàn năng của mạch logic, nghĩa là mọi mạch logic đều có thể được cài đặt bằng cách chỉ dùng các cổng Toffoli. Hơn nữa, cổng Toffoli cũng có phiên bản lượng tử. Như vậy, cách đơn giản nhất để “chuyển” mọi mạch cổ điển thành mạch lượng tử là dùng cổng Toffoli. Mặc dù về lý thuyết một mạch logic có s cổng có thể được biến đổi để chỉ dùng $O(s)$ cổng Toffoli và mỗi cổng Toffoli lượng tử cũng chỉ cần số lượng cố định các cổng lượng tử cơ bản (16 cổng, Bài tập 4.24) nên mạch lượng tử tương ứng cũng chỉ cần $O(s)$ cổng lượng tử cơ bản, nghĩa là độ phức tạp mạch “không tăng”. Tuy nhiên trong thực hành, cách thiết kế mạch này là tốn kém do số lượng cổng cơ bản cần dùng khá nhiều.

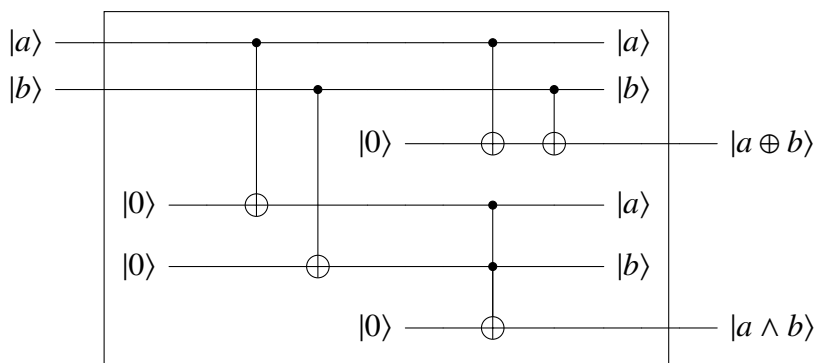
Một cách khác hiệu quả hơn là dùng các phiên bản lượng tử của các cổng logic thông dụng NOT, AND, OR, XOR như Bảng 4.1. Trong bảng này, $a, b \in \mathbb{B}$ là các bit. Lưu ý là, ngoài các qubit ứng với các bit đầu vào ta có thể phải dùng thêm các **qubit phụ trợ** (ancilla, ancillary qubit). Chẳng hạn phiên bản lượng tử của cổng AND là cổng Toffoli có dùng thêm 1 qubit phụ trợ khởi động với trạng thái $|0\rangle$.

Ví dụ 4.4.1. Bộ nửa cộng HA trong Ví dụ 2.2.1 dùng 1 cổng XOR và 1 cổng AND (cùng với 2 cổng FANOUT). Bằng cách dùng các phiên bản lượng tử tương ứng

Cổng logic	Cổng lượng tử
FANOUT 	CNOT 
NOT 	X 
AND 	Toffoli 
OR 	Toffoli, 3 X 
XOR 	2 CNOT 

Bảng 4.1: Phiên bản lượng tử của các cổng logic thông dụng.

của các cổng FANOUT, XOR, AND trong Bảng 4.1 ta có phiên bản lượng tử cho HA như Hình 4.1. Các qubit phụ trợ được cố ý đưa vào bên trong (“nội bộ”) hộp. Các qubit đầu ra không cần thiết cũng được đưa vào bên trong hộp. Về logic, mạch nhận 2 qubit đầu vào $|a\rangle, |b\rangle$, thực hiện các công việc nội bộ, rồi cho ra tổng là $|a \oplus b\rangle$ và nhớ $|a \wedge b\rangle$. \square



Hình 4.1: Phiên bản lượng tử của mạch nửa cộng HA.

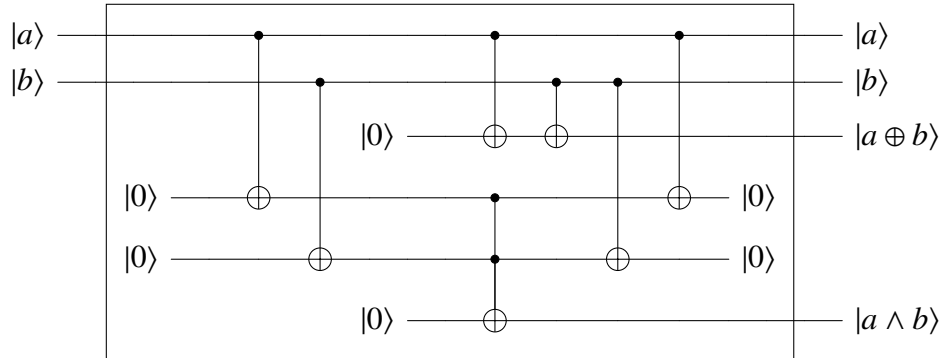
4.4.2 “Gỡ tính toán”

Trong phần trước, khi chuyển mạch logic thành mạch lượng tử, ta thấy ngoài việc dùng thêm các qubit phụ trợ thì mạch lượng tử có thể có các qubit đầu ra không cần thiết. Chẳng hạn khi dùng cổng Toffoli cho cổng AND ta có 3 đầu ra, trong đó chỉ có đầu ra $|a \wedge b\rangle$ là cần thiết. Các đầu ra không cần thiết có thể được bỏ hoặc xóa (đặt lại trạng thái 0) dễ dàng với tính toán cổ điển nhưng với tính toán lượng tử ta cần cẩn thận hơn vì các qubit này có thể ở trạng thái vướng với các qubit khác mà khi “xóa” không đúng ta có thể làm thay đổi trạng thái các qubit khác.

Nói chung, ta cần giữ các đầu ra để đảm bảo mạch khả nghịch và các đầu ra “không cần thiết” có thể được “xóa” về $|0\rangle$ bằng cách “undo” các cổng trước đó, nhờ rằng để undo $U_1 U_2$ ta dùng $U_2^\dagger U_1^\dagger$. Kỹ thuật này thường được gọi là **“gỡ tính toán”** (uncomputation) và là kỹ thuật chuẩn giúp ta làm “sạch” (clean) các qubit không cần thiết.

Cũng lưu ý, trong nhiều trường hợp, thay vì chuyển mạch logic thành mạch lượng tử một cách “cơ học” như trên, việc thiết kế mạch trực tiếp từ đầu thường cho mạch hiệu quả hơn (dùng ít cổng cơ bản hơn). Một ví dụ được cho trong Bài tập 4.31.

Ví dụ 4.4.2. (tiếp Ví dụ 4.4.1) Trong phiên bản lượng tử của bộ nửa cộng HA ở mạch 4.1 ta đã dùng thêm 4 qubit phụ trợ, nếu muốn đặt lại các trạng thái $|0\rangle$ cho các qubit này (và do đó có thể dùng làm qubit phụ trợ cho các mạch khác) thì ta có thể dùng kỹ thuật “gỡ tính toán” như Hình 4.2. Ta đã “xóa” 2 qubit phụ trợ không cần thiết về $|0\rangle$ bằng cách dùng cổng nghịch đảo của CNOT.



Hình 4.2: Mạch lượng tử HA với các qubit phụ trợ được “xóa”.

Trong mạch 4.2 ta có 4 đầu ra, trong đó có đầu vào $|a\rangle, |b\rangle$. Hai qubit này có thể được dùng tiếp để làm đầu vào cho các mạch sau đó. Lưu ý, ta không thể “xóa” 2 qubit này để chỉ giữ đúng 2 đầu ra là tổng $|a \oplus b\rangle$ và nhớ $|a \wedge b\rangle$ vì phép toán

$f : \mathbb{B}^2 \rightarrow \mathbb{B}^2$ với

$$f(a, b) = (a \oplus b, a \wedge b)$$

không khả nghịch như bảng chân trị 2.7 cho thấy. Có thể nói mạch 4.2 chính là phiên bản lượng tử của mạch khả nghịch HA trong Bài tập 2.21.

Cũng lưu ý, mạch 4.2 là mạch lượng tử “bình thường” như bao mạch lượng tử khác. Chẳng hạn, nếu gọi 2 qubit đầu vào là $|b\rangle|a\rangle$ (đúng ra là 4 qubit đầu vào $|0\rangle|0\rangle|b\rangle|a\rangle$) và 4 qubit đầu ra là $|a \wedge b\rangle|a \oplus b\rangle|b\rangle|a\rangle$ thì với trạng thái đầu vào $\frac{1}{\sqrt{2}}(|10\rangle + |11\rangle)$ là tổ hợp đều của 10 và 11, ta có đầu ra là

$$\frac{1}{\sqrt{2}}(|01\rangle|10\rangle + |10\rangle|11\rangle).$$

Như vậy ta được tổ hợp đều của các phép cộng 1 qubit $1 + 0$ và $1 + 1$ tương ứng là 01 và 10. Có thể nói, mạch đã giúp ta thực hiện cùng lúc việc cộng $1 + 0$ và $1 + 1$. Tuy nhiên, các kết quả được để trong trạng thái chồng chất mà nếu đo ta chỉ được 1 trong 2 kết quả. Đặc trưng này thường được gọi là “**song song lượng tử**” (quantum parallelism). \square

Kết quả phần này cho thấy mọi mạch logic đều có thể được chuyển đổi thành mạch lượng tử với cùng bậc lớn tiệm cận về số cổng. Như vậy, mọi tính toán cổ điển đều có thể được “**mô phỏng**” (simulation) hiệu quả bằng tính toán lượng tử. Điều đó cũng có nghĩa là mọi thuật toán cổ điển đều có thể được mô phỏng bằng thuật toán lượng tử với cùng độ phức tạp tiệm cận.

4.5 Định lý không nhân bản

Trong tính toán cổ điển, việc sao chép các bit thường là vấn đề đơn giản về kỹ thuật.⁵ Tuy nhiên, với tính toán lượng tử, việc sao chép trạng thái các qubit là điều rất khó, thậm chí là không thể. Sự bất khả thi này thường được gọi là **định lý không nhân bản** (no-cloning theorem).

Trước hết, xét trường hợp đơn giản, giả sử có 2 qubit A, B có trạng thái lần lượt là $|\psi_A\rangle = \alpha_A|0\rangle + \beta_A|1\rangle$ và $|\psi_B\rangle = \alpha_B|0\rangle + \beta_B|1\rangle$. Giả sử tồn tại phép toán U sao cho khi tác động U lên A, B thì cả 2 qubit đều có cùng trạng thái $|\psi_A\rangle$, tức là

$$U(|\psi_A\rangle \otimes |\psi_B\rangle) = |\psi_A\rangle \otimes |\psi_A\rangle.$$

⁵việc này còn được xem là “không có gì để nói” trong tính toán cổ điển nên cổng fan-out thường không được xem là cổng.

Điều này có nghĩa là

$$U \begin{pmatrix} \alpha_A \alpha_B \\ \alpha_A \beta_B \\ \beta_A \alpha_B \\ \beta_A \beta_B \end{pmatrix} = \begin{pmatrix} \alpha_A^2 \\ \alpha_A \beta_A \\ \alpha_A \beta_A \\ \beta_A^2 \end{pmatrix}.$$

Lưu ý U cố định (không phụ thuộc trạng thái của A, B) và đẳng thức trên đúng với mọi $|\psi\rangle_A$ và $|\psi\rangle_B$. Xét các trường hợp

- $\alpha_A = 1, \beta_A = 0$ (tức $|\psi_A\rangle = |0\rangle$), ta có

$$U \begin{pmatrix} \alpha_B \\ \beta_B \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (4.2)$$

- $\alpha_A = 0, \beta_A = 1$ (tức $|\psi_A\rangle = |1\rangle$), ta có

$$U \begin{pmatrix} 0 \\ 0 \\ \alpha_B \\ \beta_B \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (4.3)$$

- $\alpha_A = \frac{1}{\sqrt{2}}, \beta_A = \frac{1}{\sqrt{2}}$ (tức $|\psi_A\rangle = |+\rangle$), ta có

$$\frac{1}{\sqrt{2}} U \begin{pmatrix} \alpha_B \\ \beta_B \\ \alpha_B \\ \beta_B \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad (4.4)$$

Bấy giờ, từ (4.2) và (4.3) ta có

$$\frac{1}{\sqrt{2}} U \begin{pmatrix} \alpha_B \\ \beta_B \\ \alpha_B \\ \beta_B \end{pmatrix} = \frac{1}{\sqrt{2}} \left(U \begin{pmatrix} \alpha_B \\ \beta_B \\ 0 \\ 0 \end{pmatrix} + U \begin{pmatrix} 0 \\ 0 \\ \alpha_B \\ \beta_B \end{pmatrix} \right) = \frac{1}{\sqrt{2}} \left(\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \right) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

mâu thuẫn với (4.4). Như vậy, không tồn tại U thỏa yêu cầu. Tổng quát, ta có thể chứng minh mệnh đề sau (Bài tập 4.33).

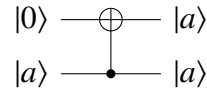
Mệnh đề 4.5.1. (Định lý không nhân bản) Không tồn tại phép toán lượng tử U trên hệ $2n$ qubit và trạng thái lượng tử $|\phi\rangle$ sao cho

$$U(|\psi\rangle \otimes |\phi\rangle) = |\psi\rangle \otimes |\psi\rangle$$

với mọi trạng thái $|\psi\rangle$ trên hệ n qubit.

△

Lưu ý, định lý không nhân bản cho thấy không thể tạo ra bản sao của một trạng thái lượng tử chưa biết. Việc sao chép một trạng thái lượng tử đã biết là hoàn toàn có thể. Chẳng hạn, dùng cổng CNOT



ta có thể sao chép trạng thái $|a\rangle$ với $a \in \{0, 1\}$. Cổng này, ngược lại không thể giúp sao chép các trạng thái khác như $|+\rangle$.

4.6 Bất đẳng thức Bell *

Phần này chạm đến “bản chất”, “nguyên lý”, “cơ chế” hay “triết học” của “thuyết lượng tử”.

Xét hai tình huống:

1. TH1: một hộp kín chứa một đồng xu đồng chất được xóc lên. Khi mở hộp (quan sát), ta sẽ thấy đồng xu ngửa hoặc sấp với xác suất đều là 50%.
2. TH2: một qubit ở trạng thái $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Khi đo (quan sát), ta sẽ được 0 hoặc 1 với xác suất đều là 50%.

Hay phức tạp hơn một chút, xét hai tình huống:

1. TH1: 2 hộp chứa Eva và Adam được hoán đổi ngẫu nhiên rồi gửi đến 2 thiên hà T và H. Trước khi mở hộp (quan sát), không ai biết được trong mỗi hộp là Eva hay Adam (xác suất 50%). Tuy nhiên, nếu người trên thiên hà T mở hộp và thấy Eva bên trong thì không cần mở hộp ở thiên hà H cũng biết hộp đó chứa Adam và ngược lại.
2. TH2: 2 qubit ở trạng thái vướng $\Psi^+ = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$ được gửi đến 2 thiên hà T và H. Trước khi đo (quan sát), không ai biết được khi đo sẽ được 0 hay 1 (xác suất 50%). Tuy nhiên, nếu người trên thiên hà T đo qubit và thấy kết quả là 0 thì không cần đo qubit ở thiên hà H cũng biết qubit đó phải là 1 và ngược lại.

Câu hỏi là: “tính ngẫu nhiên” trong TH1 và TH2 có gì khác nhau không?

Dưới góc nhìn của lý thuyết xác suất, cả 2 trường hợp đều như nhau và không phân biệt được. Ở ví dụ đầu ta có biến ngẫu nhiên X có thể nhận giá trị 0 hoặc 1 với xác suất đều là 50%. Ở ví dụ sau ta có cặp biến ngẫu nhiên X, Y có thể nhận các giá trị 0 hoặc 1 với xác suất đều là 50% và có tương quan nghịch, $\text{cor}(X, Y) = -1$. (♣)

Tuy nhiên, “cơ chế” bên dưới của 2 trường hợp ngẫu nhiên này lại khác nhau. Có thể nói, trong TH1, các giá trị đã được xác định trước khi mở (ngủ/sấp hay Eva/Adam đã được định trước), tính ngẫu nhiên đến từ việc “không biết”. Trong TH2, ngẫu nhiên là “thực sự”, kết quả chỉ được xác định khi mở. Liệu có cách nào phân biệt được 2 cơ chế ngẫu nhiên khác nhau này?

Rất thú vị, John Stewart Bell (và nhiều người khác) đã nghĩ ra được cách phân biệt 2 trường hợp này dựa trên các bất đẳng thức được gọi chung là **bất đẳng thức Bell** (Bell inequality). Các thí nghiệm giúp phân biệt 2 cơ chế ngẫu nhiên này bằng cách dùng các bất đẳng thức Bell thường được gọi là thử nghiệm Bell hay **kiểm định Bell** (Bell test).

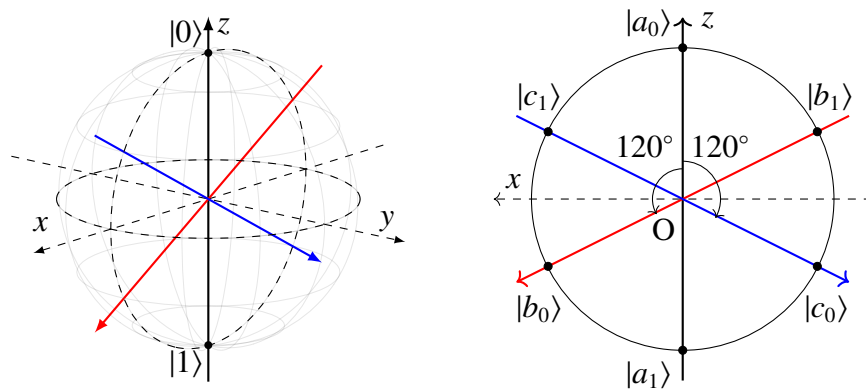
Phần này trình bày một thử nghiệm Bell cụ thể phỏng theo phần trình bày rất dễ hiểu trong [1]. Thử nghiệm này dùng trạng thái vướng 2 qubit là

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

và 3 cơ sở để đo qubit là

$$\begin{aligned} A &= \{|a_0\rangle, |a_1\rangle\}, & |a_0\rangle &= |0\rangle, |a_1\rangle = |1\rangle, \\ B &= \{|b_0\rangle, |b_1\rangle\}, & |b_0\rangle &= \frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle, |b_1\rangle = \frac{\sqrt{3}}{2}|0\rangle - \frac{1}{2}|1\rangle, \\ C &= \{|c_0\rangle, |c_1\rangle\}, & |c_0\rangle &= \frac{1}{2}|0\rangle - \frac{\sqrt{3}}{2}|1\rangle, |c_1\rangle = \frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle. \end{aligned}$$

Trên mặt cầu Bloch, các cơ sở này ứng với các trục trên mặt phẳng Oxz với góc so với Oz lần lượt là $0^\circ, 120^\circ, 240^\circ$ như minh họa trong Hình 4.3. (♣)



Hình 4.3: Các trục ứng với các cơ sở dùng trong thử nghiệm Bell.

Lưu ý, với cơ sở trực chuẩn bất kì $\{|u_0\rangle, |u_1\rangle\}$ của \mathbb{C}^2 ta luôn có (Bài tập 4.36)

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}} |b_0\rangle |b_0\rangle + \frac{1}{\sqrt{2}} |b_1\rangle |b_1\rangle.$$

Bây giờ, ta thiết lập một thử nghiệm như sau:

1. Victor chuẩn bị n cặp qubit ở trạng thái $|\Phi^+\rangle$ và chia ra 2 phần, mỗi phần có một qubit trong mỗi cặp, rồi gửi một phần cho Alice và một phần cho Bob.
2. Sau khi nhận n qubit từ Victor, với mỗi qubit, Alice chọn ngẫu nhiên một trong 3 cơ sở A, B, C để đo và ghi nhận kết quả (0 hoặc 1), rồi gửi chuỗi n bit đo được cho Victor.
3. Bob làm tương tự Alice.
4. Victor so sánh 2 chuỗi n bit Alice và Bob gửi và tính tỉ lệ những bit giống nhau của 2 chuỗi, gọi là f .

Với mỗi qubit, nếu Alice và Bob chọn cùng cơ sở thì cả 2 sẽ đo ra cùng giá trị (không quan trọng ai đo trước đo sau). Xác suất Alice và Bob chọn cùng cơ sở là $1/3$. Nếu chọn khác cơ sở (xác suất $2/3$) thì xác suất Alice và Bob đo ra giá trị giống nhau là $1/4$. Chẳng hạn, nếu Alice chọn A để đo và được 0 (là a_0) thì

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}} |a_0\rangle |a_0\rangle + \frac{1}{\sqrt{2}} |a_1\rangle |a_1\rangle$$

sẽ sụp đổ thành trạng thái $|a_0\rangle |a_0\rangle$ nên qubit mà Bob giữ sẽ có trạng thái (\clubsuit)

$$|a_0\rangle = |0\rangle = \frac{1}{2} |b_0\rangle + \frac{\sqrt{3}}{2} |b_1\rangle$$

nên Bob đo theo B sẽ được 0 (b_0) với xác suất $1/4$. Các trường hợp khác (Bob đo trước Alice hoặc Alice và Bob chọn các cơ sở khác) tương tự. Như vậy, xác suất Alice và Bob đo ra 2 bit giống nhau là

$$\frac{1}{3} + \frac{2}{3} \frac{1}{4} = \frac{1}{2}.$$

Do đó, khi n đủ lớn, ta có

$$f \approx 1/2 \quad (4.5)$$

Bây giờ, nếu việc tương quan được qui định trước (tương tự TH1 ở trên) thì ta có thể nói rằng có 3 “**biến ẩn**” (hidden variable) quyết định trước kết quả sẽ đo ra 0 hay 1 cho mỗi cơ sở A, B, C mà ta kí hiệu là V_A, V_B, V_C . Khi đó $V = v$ có nghĩa là sẽ đo ra v (mặc dù chưa đo thì ta không biết v), chẳng hạn nếu $V_A = 0$ thì khi đo

V_A	V_B	V_C	A, A	A, B	A, C	B, A	B, B	B, C	C, A	C, B	C, C
0	0	0	✓	✓	✓	✓	✓	✓	✓	✓	✓
0	0	1	✓	✓		✓	✓				✓
0	1	0	✓		✓		✓		✓		✓
0	1	1	✓				✓	✓		✓	✓
1	0	0	✓				✓	✓		✓	✓
1	0	1	✓		✓		✓		✓		✓
1	1	0	✓	✓		✓	✓				✓
1	1	1	✓	✓	✓	✓	✓	✓	✓	✓	✓

Bảng 4.2: Các trường hợp định trước và kết quả đồng thuận tương ứng.

theo cơ sở A sẽ được 0. Bảng 4.2 xét 8 trường hợp có thể của V_A, V_B, V_C và kết quả có đồng thuận hay không tương ứng.

Mỗi dòng của Bảng 4.2 ứng với một trường hợp định trước, mỗi cột tương ứng với một cặp cơ sở mà Alice và Bob có thể chọn, ô tương ứng được đánh dấu ✓ nếu khi đo qubit theo các cơ sở đã chọn, Alice và Bob được hai bit giống nhau. Chẳng hạn, dòng 2 ứng với trường hợp $V_A = 0, V_B = 0, V_C = 1$, qui định trước rằng khi dùng cơ sở A, B, C để đo thì được 0, 0, 1 tương ứng. Do đó, ở cột A, B , khi Alice chọn cơ sở A còn Bob chọn cơ sở B để đo thì Alice và Bob đều cùng được 0 nên giống nhau.

Ta thấy các dòng của Bảng 4.2 đều có ít nhất 5 lần đồng thuận trên 9. Do đó xác suất đồng thuận (khi thay đổi các cấu hình và/hoặc các lựa chọn cặp cơ sở) ít nhất là $5/9$. Khi n đủ lớn, ta có tỉ lệ đồng thuận

$$f \geq 5/9. \quad (4.6)$$

Bất đẳng thức (4.6) này được gọi là một bất đẳng thức Bell và nó cung cấp một cách để có thể kiểm tra giả thuyết về việc có biến ẩn xác định trước kết quả đo hay không? Nếu bất đẳng thức (4.6) bị vi phạm thì không thể có biến ẩn. Ta thấy (4.5) vi phạm bất đẳng thức (4.6), như vậy, hiện tượng vướng lượng tử không giải thích được bằng thuyết biến ẩn. Đã có nhiều thử nghiệm Bell tinh vi hơn, dựa trên các bất đẳng thức tương tự (4.6) và kết quả của tất cả các thử nghiệm này đều cho thấy tính ngẫu nhiên lượng tử không giải thích được bằng thuyết biến ẩn.⁶

Dưới góc độ Toán học, ta có thể hiểu rất đơn giản là trước khi đo, hệ lượng tử có

⁶John Clauser, Alain Aspect và Anton Zeilinger được trao Giải Nobel Vật lý năm 2022 cho các công trình thử nghiệm như vậy.

trạng thái chồng chất (tổ hợp tuyến tính) các trạng thái “tương thích” với phép đo

$$|\psi\rangle = \sum_{k \in \mathcal{M}} \alpha_k |k\rangle.$$

Ngay khi đo, hệ sẽ sụp đổ thành một trạng thái $|k\rangle$ tương ứng với k được chọn ngẫu nhiên từ \mathcal{M} với xác suất $|\alpha_k|^2$. Sự sụp đổ trạng thái là tức thời và không cần quá trình tương tác hay truyền tín hiệu nào cả. Cách hiểu này được gọi là **diễn giải Copenhagen** (Copenhagen interpretation).⁷ Lưu ý, cũng có những cách diễn giải khác, nhưng cách này có lẽ là “đơn giản” nhất về mặt Toán học.

Bài tập

4.1 Cho biết các trạng thái sau là tách được hay vướng, nếu tách được thì biểu diễn trên mặt cầu Bloch

- (a) $\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$. (c) $\frac{1}{4}(3|00\rangle - \sqrt{3}|01\rangle + \sqrt{3}|10\rangle - |11\rangle)$.
 (b) $\frac{1}{\sqrt{2}}(|10\rangle + i|11\rangle)$. (d) $\frac{1}{\sqrt{3}}|0\rangle|+\rangle + \sqrt{\frac{2}{3}}|1\rangle|-\rangle$.

4.2 Cho hệ n qubit, viết lại tường minh các trạng thái sau theo cơ sở tính toán

- (a) $|0\rangle^{\otimes n}, |1\rangle^{\otimes n}$. (b) $|+\rangle^{\otimes n}, |-\rangle^{\otimes n}$. (c) $|i\rangle^{\otimes n}, |-i\rangle^{\otimes n}$.

4.3 Cho hệ 2 qubit với trạng thái

$$|\psi\rangle = \frac{1}{2}|00\rangle - \frac{i}{2}|10\rangle + \frac{1}{\sqrt{2}}|11\rangle.$$

Khảo sát các phép đo sau

- (a) Đo đồng thời 2 qubit.
 (b) Đo qubit 0.
 (c) Đo qubit 1.
 (d) Đo qubit 0 rồi đo qubit 1 và so kết quả với Câu (a).
 (e) Đo qubit 1 rồi đo qubit 0 và so kết quả với Câu (a).

4.4 Tương tự Bài tập 4.3 cho các trạng thái 2 qubit sau

⁷được đặt tên theo thủ đô Copenhagen của Đan Mạch, là nơi làm việc của các nhà Vật lý lượng tử nổi tiếng là Niels Bohr và Werner Heisenberg.

$$(a) |\psi_1\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{2}|01\rangle + \frac{\sqrt{3}}{4}|10\rangle + \frac{1}{4}|11\rangle.$$

$$(b) |\psi_2\rangle = \frac{i}{\sqrt{10}}|00\rangle + \frac{1-2i}{\sqrt{10}}|01\rangle + \frac{e^{i\pi/100}}{\sqrt{10}}|10\rangle + \frac{\sqrt{3}}{\sqrt{10}}|11\rangle.$$

$$(c) |\psi_3\rangle = \frac{1}{4}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle + \frac{\sqrt{3}}{4}|11\rangle.$$

4.5 Cho hệ 3 qubit với trạng thái

$$|\psi\rangle = \frac{1}{2}|000\rangle + \frac{1}{2}|100\rangle + \frac{1}{2}|101\rangle - \frac{1}{2}|111\rangle.$$

Khảo sát các phép đo sau

- | | |
|---------------------------|----------------------|
| (a) Đo đồng thời 3 qubit. | (d) Đo qubit 2. |
| (b) Đo qubit 0. | (e) Đo qubit 0 và 1. |
| (c) Đo qubit 1. | (f) Đo qubit 0 và 2. |

4.6 Tương tự Bài tập 4.5 cho trạng thái 3 qubit sau

$$|\psi\rangle = \frac{1}{6}|000\rangle + \frac{1}{3\sqrt{2}}|001\rangle + \frac{1}{\sqrt{6}}|010\rangle + \frac{1}{2}|011\rangle + \frac{1}{6}|100\rangle + \frac{1}{3}|101\rangle + \frac{1}{6}|110\rangle + \frac{1}{\sqrt{3}}|111\rangle.$$

4.7 Cho $\{|\psi_1\rangle, \dots, |\psi_m\rangle\}$ là họ các vector trực chuẩn trong \mathbb{C}^N , chứng minh ma trận sau là ma trận chiếu

$$P = \sum_{k=1}^m |\psi_k\rangle \langle \psi_k|.$$

Hơn nữa, mọi ma trận chiếu $P \in \mathbb{C}^{N \times N}$ đều có thể được viết ở dạng trên với họ $\{|\psi_1\rangle, \dots, |\psi_m\rangle\}$ trực chuẩn nào đó của \mathbb{C}^N .

4.8 Cho hệ n qubit ở trạng thái $|\psi\rangle \in \mathbb{C}^{2^n}$, chứng minh phép đo riêng qubit 0 theo cơ sở tính toán chính là phép đo chiếu theo họ ma trận chiếu sau

$$\{I_{2^{n-1}} \otimes (|0\rangle\langle 0|), I_{2^{n-1}} \otimes (|1\rangle\langle 1|)\}.$$

4.9 Phép đo chiếu được xác định bằng ma trận Hermite.

- (a) Cho họ các ma trận chiếu $\{P_0, \dots, P_{m-1}\}$ đầy đủ ($\sum_{k=0}^{m-1} P_k = I$) và danh sách các số thực tương ứng $\alpha_0, \dots, \alpha_{m-1}$, chứng minh

$$M = \sum_{k=0}^{m-1} \alpha_k P_k$$

là một ma trận Hermite.

(b) Cho $M \in \mathbb{C}^{N \times N}$ là một ma trận Hermite có phân rã phổ là

$$M = \sum_{i=1}^N \lambda_i |v_i\rangle\langle v_i|$$

với $\lambda_0, \dots, \lambda_N$ là các trị riêng thực của M và $|v_0\rangle, \dots, |v_N\rangle$ là các vector riêng tương ứng. Giả sử danh sách các trị riêng này gồm m giá trị phân biệt là $\alpha_0, \dots, \alpha_{m-1}$, đặt

$$P_k = \sum_{i=1}^N \delta_{\lambda_i=\alpha_k} |v_i\rangle\langle v_i|, \quad \text{với } \delta_{\lambda_i=\alpha_k} = \begin{cases} 1 & \lambda_i = \alpha_k, \\ 0 & \lambda_i \neq \alpha_k. \end{cases}$$

Chứng minh tập $\{P_0, \dots, P_{m-1}\}$ lập thành một phép đo chiếu.

4.10 Tương tự Ví dụ 4.1.5, khảo sát các quan sát X, Y trên hệ 1 qubit.

4.11 Cho biết đầu ra $|\phi\rangle$ của mạch 2 qubit sau

$$|\psi\rangle \left\{ \begin{array}{c} \text{---} \boxed{H} \text{---} \\ \text{---} \boxed{I} \text{---} \end{array} \right\} |\phi\rangle$$

khi biết đầu vào $|\psi\rangle$ là

- | | | | |
|----------------------------|----------------------------|------------------------|------------------------|
| (a) $ 0\rangle +\rangle$. | (c) $ +\rangle +\rangle$. | (e) $ \Phi^+\rangle$. | (g) $ \Psi^+\rangle$. |
| (b) $ 1\rangle -\rangle$. | (d) $ -\rangle -\rangle$. | (f) $ \Phi^-\rangle$. | (h) $ \Psi^-\rangle$. |

4.12 Vẽ mạch và làm tương tự Bài tập 4.11 cho các phép toán sau trên hệ 2 qubit

- | | | |
|---------------------|---------------------|------------------------------------|
| (a) $H \otimes I$. | (c) $X \otimes X$. | (e) $(X \otimes Z)(Z \otimes X)$. |
| (b) $H \otimes X$. | (d) $X \otimes Y$. | (f) $(H \otimes S)(S \otimes H)$. |

4.13 Cho ma trận

$$U = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} & 0 & e^{i\frac{\pi}{4}} \\ 1 & 0 & -1 & 0 \\ 0 & e^{i\frac{\pi}{4}} & 0 & -e^{i\frac{\pi}{4}} \end{bmatrix}.$$

(a) Chứng minh U unita và do đó U biểu diễn một phép toán lượng tử 2 qubit.

(b) Chứng minh rằng

$$|v\rangle = \begin{bmatrix} 2+i \\ \sqrt{2}+1 \\ 1 \\ 1 \end{bmatrix}$$

là một vector riêng của U với trị riêng là $\lambda = e^{i\frac{\pi}{4}}$.

(c) U còn có các trị riêng với vector riêng tương ứng nào khác?

4.14 Khảo sát phép toán 2 qubit $U = H \otimes X$.

(a) Cho biết tác động của U lên các vector của cơ sở tính toán.

(b) Xác định ma trận biểu diễn của U từ Câu (a).

(c) Xác định ma trận biểu diễn của U bằng phép tích tensor.

(d) Cho biết tác động của U lên trạng thái

$$|\psi\rangle = \frac{1}{4}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle + \frac{\sqrt{3}}{4}|11\rangle.$$

4.15 Ví dụ 4.2.2 cho thấy mạch 2 qubit biến $|\Phi^+\rangle$ thành $|\Psi^+\rangle$. Tương tự, thiết kế mạch để biến đổi $|\Phi^+\rangle$ thành

(a) $|\Phi^-\rangle$.

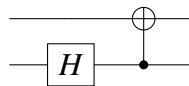
(b) $|\Psi^-\rangle$.

4.16 Cho biết đầu ra của mạch sau

$$|\psi\rangle \left\{ \begin{array}{c} \text{---} \\ \text{---} \boxed{X} \text{---} \end{array} \right\} |\phi\rangle$$

khi đầu vào là các trạng thái Bell.

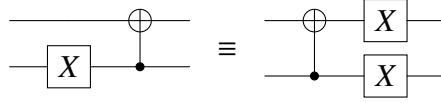
4.17 Cho biết đầu ra của mạch sau



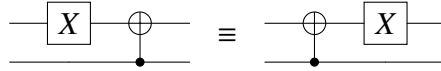
khi đầu vào là các trạng thái của cơ sở tính toán.

4.18 Chứng minh các đẳng thức mạch sau

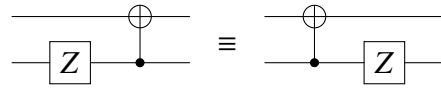
(a)



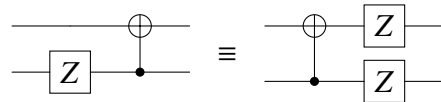
(b)



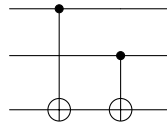
(c)



(d)

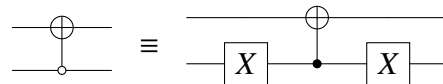


4.19 Cho mạch 3 qubit sau



- Cho biết biến đổi của mạch trên các vector của cơ sở tính toán $|abc\rangle$, $a, b, c \in \{0, 1\}$.
- Từ Câu (a) nhận diện phiên bản mạch cổ điển tương ứng.
- Xác định ma trận biểu diễn mạch.
- Xác định biến đổi của mạch lên trạng thái $|\psi\rangle = \sum_{k=0}^7 \alpha_k |k\rangle \in \mathbb{C}^8$.

4.20 Cổng CNOT thực hiện thao tác “lật qubit mục tiêu khi qubit điều khiển là 1”. **Cổng phản CNOT** (anti-CNOT gate), ngược lại, thực hiện thao tác “lật qubit mục tiêu khi qubit điều khiển là 0”. Cổng phản CNOT được vẽ bằng kí hiệu chấm rỗng trên qubit điều khiển (thay vì chấm đặc như CNOT). Chứng minh



4.21 Chứng minh, biến đổi CNOT trong cơ sở Z $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ chính là biến đổi CNOT_{01} trong cơ sở X $\{|+\rangle|+\rangle, |+\rangle|-\rangle, |-\rangle|+\rangle, |-\rangle|-\rangle\}$.

4.22 Cho 4 trạng thái 2 qubit sau

$$\begin{aligned} |\omega_0\rangle &= \frac{1}{2}(-|00\rangle + |01\rangle + |10\rangle + |11\rangle), \\ |\omega_1\rangle &= \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle + |11\rangle), \\ |\omega_2\rangle &= \frac{1}{2}(|00\rangle + |01\rangle - |10\rangle + |11\rangle), \\ |\omega_3\rangle &= \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle - |11\rangle). \end{aligned}$$

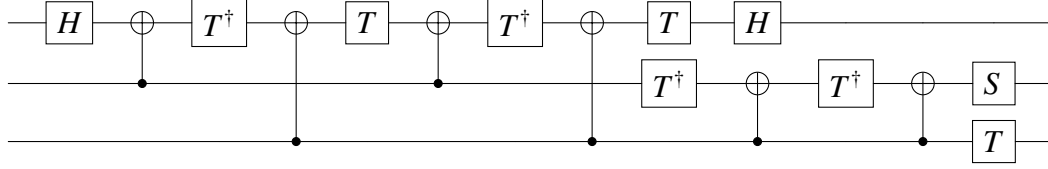
- (a) Chứng minh $\Omega = \{|\omega_0\rangle, |\omega_1\rangle, |\omega_2\rangle, |\omega_3\rangle\}$ là một cơ sở trực chuẩn của \mathbb{C}^4 .
- (b) Tìm ma trận $U \in \mathbb{C}^{4 \times 4}$ chuyển cơ sở tính toán thành cơ sở Ω . Chứng minh U unita và do đó U là một cổng 2 qubit.
- (c) Tìm U^{-1} .
- (d) Cho biết tác động của U trên các vector cơ sở Bell.
- (e) Cho biết tác động của U trên các vector cơ sở X $\{|+\rangle|+\rangle, |+\rangle|-\rangle, |-\rangle|+\rangle, |-\rangle|-\rangle\}$.
- (f) Cho biết tác động của $X \otimes I$ và $I \otimes X$ trên các vector cơ sở Ω .
- (g) Cho biết tác động của $\text{CNOT} = \text{CNOT}_{10}$ và CNOT_{01} trên các vector cơ sở Ω .
- (h) Cho biết tác động của SWAP trên các vector cơ sở Ω .

4.23 Cổng Mølmer-Sørensen (Mølmer-Sørensen gate, MS) là cổng 2 qubit được xác định bởi

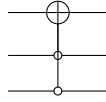
$$\begin{aligned} \text{MS}|00\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + i|11\rangle), & \text{MS}|01\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - i|10\rangle), \\ \text{MS}|10\rangle &= \frac{1}{\sqrt{2}}(|10\rangle - i|01\rangle), & \text{MS}|11\rangle &= \frac{1}{\sqrt{2}}(|11\rangle + i|00\rangle). \end{aligned}$$

- (a) Tìm ma trận biểu diễn MS.
- (b) Cho biết tác động của MS lên $|\psi\rangle = \sum_{k=0}^3 \alpha_k |k\rangle$.
- (c) Chứng minh $\text{MS}^8 = I$.

4.24 Chứng minh cổng Toffoli có thể được “thi công” bằng cổng CNOT và các cổng 1 qubit thông dụng như sau



4.25 Cổng Toffoli (CCNOT) thực hiện thao tác “lật qubit mục tiêu khi cả 2 qubit điều khiển là 1”. **Cổng phản Toffoli** (anti-Toffoli gate), ngược lại, thực hiện thao tác “lật qubit mục tiêu khi cả 2 qubit điều khiển là 0”. Cổng phản Toffoli được vẽ bằng kí hiệu chấm rỗng trên các qubit điều khiển như sau



- Cho biết biến đổi của mạch trên các vector của cơ sở tính toán $|abc\rangle$, $a, b, c \in \{0, 1\}$.
- Từ Câu (a) nhận diện phiên bản mạch cổ điển tương ứng.
- Xác định ma trận biểu diễn mạch.
- Xác định biến đổi của mạch lên trạng thái $|\psi\rangle = \sum_{k=0}^7 \alpha_k |k\rangle \in \mathbb{C}^8$.

4.26 Xét trạng thái 3 qubit

$$|\text{GHZ}\rangle = \frac{1}{\sqrt{2}}|000\rangle + \frac{1}{\sqrt{2}}|111\rangle.$$

- Chứng minh $|\text{GHZ}\rangle$ là trạng thái vướng.
- Khảo sát phép đo riêng qubit 0, qubit 1, qubit 2 và nhận xét.
- Thiết kế mạch 3 qubit để tạo trạng thái $|\text{GHZ}\rangle$.

4.27 Làm lại Bài tập 4.26 cho trạng thái 3 qubit

$$|\text{W}\rangle = \frac{1}{\sqrt{3}}|001\rangle + \frac{1}{\sqrt{3}}|010\rangle + \frac{1}{\sqrt{3}}|100\rangle.$$

4.28 Tương tự Ví dụ 4.4.1 và 4.4.2, thiết kế mạch lượng tử mô phỏng mạch cộng đầy đủ FA ở Ví dụ 2.2.2.

4.29 Cho phép toán logic $f : \mathbb{B}^3 \rightarrow \mathbb{B}^3$ biến 3 bit đầu vào ABC thành 3 bit đầu ra DEF được xác định bởi bảng chân trị sau

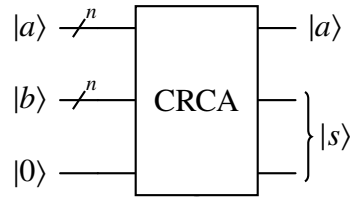
A	B	C	D	E	F
0	0	0	1	0	0
0	0	1	1	0	1
0	1	0	1	1	0
0	1	1	1	1	1
1	0	0	0	1	0
1	0	1	0	0	1
1	1	0	0	1	1
1	1	1	0	0	0

- (a) Cho thấy f khả nghịch.
- (b) Thiết kế mạch logic (và đơn giản mạch) cho f .
- (c) Chuyển mạch ở Câu (b) thành mạch lượng tử với chỉ 3 qubit vào (chứa A, B, C) và 3 qubit ra (chứa D, E, F), có thể dùng thêm các qubit phụ trợ nhưng các qubit này phải được xóa về $|0\rangle$.
- (d) Cho biết kết quả chạy mạch ở Câu (c) với trạng thái đầu vào 3 qubit là

$$|\psi\rangle = \frac{1}{\sqrt{3}}(|000\rangle + |100\rangle + |111\rangle).$$

4.30 Chuyển mạch cộng RCA ở Ví dụ 2.2.3 thành mạch lượng tử.

4.31 Để thiết kế mạch cộng 2 số nhị phân n bit $a, b \in \mathbb{B}^n$ được tổng $s = a + b$ là số nhị phân $n + 1$ bit ta có thể dùng mạch lượng tử $n + (n + 1) = 2n + 1$ qubit có sơ đồ như sau



Mạch có hai thanh ghi, thanh ghi trên gồm n qubit chứa đầu vào $|a\rangle$ và đầu ra cũng là $|a\rangle$, thanh ghi dưới gồm $n + 1$ qubit, chứa đầu vào $|0\rangle |b\rangle$ và đầu ra là $|s\rangle$. Lưu ý là phép toán này khả nghịch do với mỗi đầu ra s, a ta xác định được duy nhất đầu vào $0, b, a$ với $b = s - a$ nên có thể hiện thực thành mạch lượng tử.

- (a) Thiết kế mạch QRCA (Quantum RCA) có “giao tiếp” như trên (có thể dùng thêm các qubit phụ trợ nếu cần).
- (b) So sánh số lượng cổng lượng tử cơ bản của mạch này với mạch ở Bài tập 4.30.

4.32 Cho 2 số nhị phân $a, b \in \mathbb{B}^n$, tìm quan hệ giữa $\overline{a + b}$ và $a - b$. Từ đó thiết kế mạch lượng tử thực hiện phép trừ 2 số nhị phân n bit.

4.33 Chứng minh Mệnh đề 4.5.1.

4.34 Thiết kế mạch để giúp sao chép các trạng thái lượng tử

- (a) $|+\rangle$. (c) $|ab\rangle, a, b \in \{0, 1\}$. (e) $|\text{GHZ}\rangle$.
 (b) $|\Phi^+\rangle$. (d) $|ab\rangle, a, b \in \{+, -\}$. (f) $|W\rangle$.

4.35 Thiết kế mạch n qubit để cài đặt R_u , phép toán phản xạ quanh trục $|u\rangle$, với

- (a) $|u\rangle = |1^n\rangle$. (b) $|u\rangle = |0^n\rangle$. (c) $|u\rangle = |+\rangle^{\otimes n}$. (d) $|u\rangle = |-\rangle^{\otimes n}$.

4.36 Chứng minh, với cơ sở trực chuẩn bất kì $\{|u_0\rangle, |u_1\rangle\}$ của \mathbb{C}^2 , ta có

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}} |b_0\rangle |b_0\rangle + \frac{1}{\sqrt{2}} |b_1\rangle |b_1\rangle.$$

Chương 5

Các thuật toán lượng tử cơ bản

Chương này trình bày các thuật toán và giao thức lượng tử cơ bản. Mặc dù đơn giản nhưng các thuật toán và giao thức này cũng cho thấy các ưu thế tính toán so với các thuật toán cổ điển tương ứng mà ta thường gọi là **ưu thế lượng tử** (quantum advantage, quantum supremacy).

5.1 Mã siêu đặc và dịch chuyển lượng tử

Phần này cho thấy vướng lượng tử là một nguồn tài nguyên mà khi được vận dụng khéo léo sẽ mang lại nhiều lợi thế như trong các ứng dụng liên quan đến truyền tin là mã siêu đặc và dịch chuyển lượng tử. Với mã siêu đặc, ta có thể dùng 1 qubit để truyền thông tin thay cho 2 bit cổ điển. Ngược lại, với dịch chuyển lượng tử, ta dùng 2 bit cổ điển để “truyền” trạng thái lượng tử. Cả 2 giao thức này đều cần các qubit ở trạng thái vướng.

5.1.1 Nhắc lại vướng lượng tử

Ta đã biết, 2 qubit (A, B) có trạng thái $|\psi\rangle \in \mathbb{C}^4$ vướng nếu $|\psi\rangle$ không tách được, tức là không có $|\psi_A\rangle, |\psi_B\rangle \in \mathbb{C}^2$ sao cho

$$|\psi\rangle = |\psi_A\rangle \otimes |\psi_B\rangle = |\psi_A\rangle |\psi_B\rangle.$$

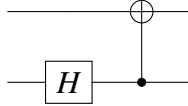
Với hệ 2 qubit, các trạng thái của cơ sở Bell $\{|\Phi^+\rangle, |\Psi^+\rangle, |\Phi^-\rangle, |\Psi^-\rangle\}$ là các trạng thái vướng điển hình.

Ví dụ 4.2.2 cho thấy ta có thể biến đổi $|\Phi^+\rangle$ thành $|\Psi^+\rangle$ bằng cách tác động X lên qubit B như sau

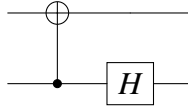
$$|\Phi^+\rangle \left\{ \begin{array}{c} \text{---} \boxed{X} \text{---} \\ \text{---} \end{array} \right\} |\Psi^+\rangle$$

Tương tự, Bài tập 4.15 cho thấy cách biến đổi $|\Phi^+\rangle$ thành $|\Phi^-\rangle, |\Psi^-\rangle$.

Ví dụ 4.3.1 và Bài tập 4.17 cho thấy mạch sau



chuyển cơ sở tính toán thành cơ sở Bell, tức là biến $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ lần lượt thành $|\Phi^+\rangle, |\Psi^+\rangle, |\Phi^-\rangle, |\Psi^-\rangle$. Ngược lại biến đổi ngược sau



chuyển cơ sở Bell thành cơ sở tính toán. Nhớ rằng, nghịch đảo của H và CNOT là chính chúng.

5.1.2 Mã siêu đặc

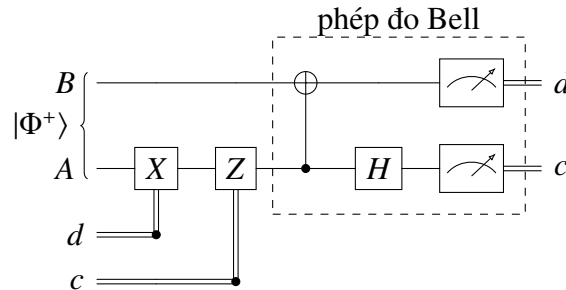
Giả sử Alice muốn gửi thông tin có nội dung là 1 lựa chọn trong 4 lựa chọn nào đó cho Bob.¹ Đánh số 4 lựa chọn là 0, 1, 2, 3 và bằng cách dùng số nhị phân, Alice cần dùng 2 bit cổ điển để mô tả lựa chọn của mình cho Bob (00 là 0, 01 là 1, 10 là 2, 11 là 3).

Rất thú vị, nếu Alice và Bob chia sẻ cặp qubit (A, B) ở trạng thái vướng $|\Phi^+\rangle$ thì Alice chỉ cần gửi cho Bob 1 qubit (thay vì 2 bit cổ điển) để truyền thông tin trên. Cụ thể, gọi cd là 2 bit biểu diễn lựa chọn của Alice, sau đây là các bước

1. Alice và Bob chia sẻ cặp qubit (A, B) ở trạng thái $|\Phi^+\rangle$, Alice giữ qubit A còn Bob giữ qubit B .
2. Nếu $d = 1$, Alice tác động cổng X lên qubit A (nếu $d = 0$ thì không).
3. Nếu $c = 1$, Alice tác động cổng Z lên qubit A (nếu $c = 0$ thì không).
4. Alice gửi qubit A cho Bob.
5. Bob thực hiện phép đo cặp qubit (A, B) theo cơ sở Bell sẽ được kết quả đo là 2 bit cd mà Alice muốn gửi.

¹Alice và Bob là các nhân vật tượng trưng hay được dùng trong các giao thức truyền tin.

Giao thức (các bước) trên có thể được mô tả bằng mạch sau



Ý tưởng của giao thức trên khá đơn giản

- Alice mã hóa lựa chọn của mình bằng các trạng thái Bell: $|\Phi^+\rangle$ là 00, $|\Psi^+\rangle$ là 01, $|\Phi^-\rangle$ là 10, $|\Psi^-\rangle$ là 11.
- Quan trọng là từ cặp qubit (A, B) ban đầu ở trạng thái $|\Phi^+\rangle$, Alice có thể biến đổi sang các trạng thái Bell khác bằng cách chỉ tác động lên qubit A (Alice giữ A, Bob giữ B). Rất may, điều này có thể làm được bằng cách tác động X và/hoặc Z lên chỉ qubit A như đã thấy.
- Sau khi nhận qubit A từ Alice, Bob giữ đủ cặp qubit (A, B) nên có thể thực hiện phép đo trong cơ sở Bell, còn gọi là **phép đo Bell** (Bell measurement), để nhận lại đúng chuỗi 2 bit mà Alice muốn gửi. Phép đo Bell có thể được thực hiện bằng mạch chuyển cơ sở Bell sang cơ sở tính toán và đo (theo cơ sở tính toán).

Giao thức trên thường được gọi là **mã đậm đặc** (superdense coding). Nhận xét

- Ta vẫn cần 2 qubit, hơn nữa, rất quan trọng, 2 qubit phải ở trạng thái vướng (chẳng hạn $|\Phi^+\rangle$).²
- Bằng cách dùng nhiều cặp qubit ở trạng thái vướng, mã đậm đặc có thể được mở rộng để truyền thông tin phức tạp hơn, trong đó thay vì gửi $2n$ bit cổ điển thì chỉ cần gửi n qubit.
- Ta bỏ qua các chi tiết như cách chuẩn bị cặp qubit vướng, cách giữ cho các qubit vẫn ở trạng thái chuẩn bị, cách truyền/gửi qubit hiệu quả, ... Các vấn đề này rất quan trọng trong thực tế và đòi hỏi các chi tiết vật lý, kỹ thuật, công nghệ là những thứ mà tài liệu này “không đụng đến”.

²các cặp qubit vướng, như vậy, được xem là tài nguyên (resource) cho tính toán lượng tử.

5.1.3 Dịch chuyển lượng tử

Trong phần trước, Alice gửi một qubit để truyền thông điệp 2 bit cho Bob. Phần này, Alice gửi 2 bit để truyền trạng thái của một qubit. Cụ thể, Alice muốn gửi cho Bob trạng thái $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ của một qubit Q mà không biết α, β . Cũng như phần trước, Alice và Bob được dùng một tài nguyên là cặp qubit ở trạng thái vướng $|\Phi^+\rangle$.

Cách hiển nhiên nhất là Alice gửi trực tiếp Q cho Bob một cách vật lý.³ Tuy nhiên, ở đây ta giả sử giải pháp này không tiến hành được. Lưu ý, không giống thông tin cổ điển có thể dễ dàng được sao chép và truyền/nhận, Alice không thể sao chép trạng thái $|\psi\rangle$ và gửi đi vì định lý không nhân bản cấm việc này. Hơn nữa, Alice cũng không thể chỉ dùng cách truyền tin cổ điển để chuyển $|\psi\rangle$ cho Bob vì nếu một bên thứ 3 như Charlie⁴ nhận cùng thông tin như Bob thì Charlie cũng tạo được trạng thái $|\psi\rangle$, vi phạm định lý không nhân bản. Mấu chốt là Alice và Bob chia sẻ cặp qubit ở trạng thái vướng (mà Charlie không được tham gia).

Trước hết, Alice và Bob chia sẻ cặp qubit (A, B) ở trạng thái $|\Phi^+\rangle$, Alice giữ qubit A còn Bob giữ qubit B . Alice cũng giữ qubit Q có trạng thái $|\psi\rangle$. Sau đây là các bước cần tiến hành

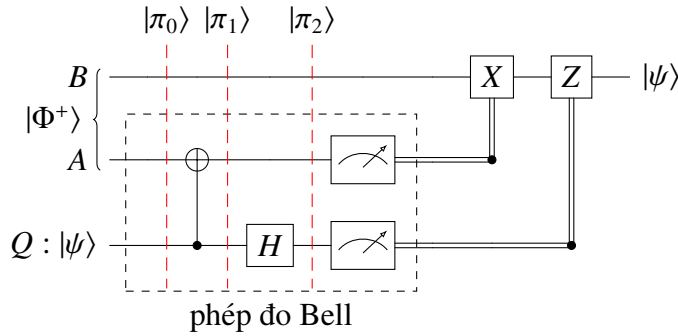
1. Alice tác động CNOT trên cặp qubit (Q, A) rồi tác động H lên Q .
2. Alice đo cặp qubit (Q, A) (theo cơ sở tính toán) được kết quả là 2 bit qa .
3. Alice gửi 2 bit qa cho Bob.
4. Bob nhận qa và tác động lên qubit B mình giữ như sau
 - Nếu $a = 1$, Bob tác động cổng X lên qubit B (nếu $b = 0$ thì không).
 - Nếu $q = 1$, Bob tác động cổng Z lên qubit B (nếu $q = 0$ thì không).

Kết quả, qubit B sẽ có trạng thái là $|\psi\rangle$ là trạng thái mà Alice muốn gửi.

Giao thức trên có thể được mô tả bằng mạch sau

³tương tự như gửi hàng qua bưu điện.

⁴sau Alice và Bob thì các nhân vật khác như Charlie, Eve cũng xuất hiện khi cần trong các giao thức.



Tính đúng đắn của giao thức trên được kiểm chứng qua từng bước như sau. Trước hết, giả sử trạng thái của Q là $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, ta có trạng thái bắt đầu của hệ 3 qubit (Q, A, B) là

$$\begin{aligned} |\pi_0\rangle &= |\psi\rangle|\Phi^+\rangle = (\alpha|0\rangle + \beta|1\rangle)|\Phi^+\rangle = \alpha|0\rangle|\Phi^+\rangle + \beta|1\rangle|\Phi^+\rangle \\ &= \alpha|0\rangle\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) + \beta|1\rangle\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ &= \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle). \end{aligned}$$

Sau khi Alice tác động CNOT lên (Q, A), trạng thái của hệ là

$$|\pi_1\rangle = \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|110\rangle + \beta|101\rangle).$$

Lưu ý, trong $|\pi_0\rangle$, Q phân tách với (A, B) nhưng trong $|\pi_1\rangle$, cả 3 qubit (Q, A, B) vướng nhau. Sau khi Alice tác động Hadamard lên Q , trạng thái của hệ là

$$\begin{aligned} |\pi_2\rangle &= \frac{1}{\sqrt{2}}(\alpha|+00\rangle + \alpha|+11\rangle + \beta|-10\rangle + \beta|-01\rangle) \\ &= \frac{1}{2}(\alpha|000\rangle + \alpha|100\rangle + \alpha|011\rangle + \alpha|111\rangle + \beta|010\rangle - \beta|110\rangle + \beta|001\rangle - \beta|101\rangle) \\ &= \frac{1}{2}|00\rangle(\alpha|0\rangle + \beta|1\rangle) + \frac{1}{2}|01\rangle(\beta|0\rangle + \alpha|1\rangle) \\ &\quad + \frac{1}{2}|10\rangle(\alpha|0\rangle - \beta|1\rangle) + \frac{1}{2}|11\rangle(-\beta|0\rangle + \alpha|1\rangle). \end{aligned}$$

Bây giờ, Alice đo (Q, A) sẽ được 1 trong 4 trường hợp (với xác suất đều là 25%) và theo đó qubit B sẽ sụp thành trạng thái tương ứng

- 00: B thành $\alpha|0\rangle + \beta|1\rangle$,
- 01: B thành $\beta|0\rangle + \alpha|1\rangle$,
- 10: B thành $\alpha|0\rangle - \beta|1\rangle$,

- 11: B thành $-\beta|0\rangle + \alpha|1\rangle$.

Sau đó, Alice báo kết quả cho Bob (2 bit qa) mà theo đó Bob sẽ “hiệu chỉnh” trạng thái của B để được $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Chẳng hạn, 00 thì không cần làm gì, còn 11 thì Bob tác động X rồi Z lên B

$$-\beta|0\rangle + \alpha|1\rangle \xrightarrow{X} \alpha|0\rangle - \beta|1\rangle \xrightarrow{Z} \alpha|0\rangle + \beta|1\rangle.$$

Toàn bộ qui trình trên có thể được hiểu đơn giản hơn từ nhận xét (Bài tập 5.4)

$$|\pi_0\rangle = |\psi\rangle|\Phi^+\rangle = \frac{1}{2}(|\Phi^+\rangle|\psi\rangle + |\Phi^-\rangle(X|\psi\rangle) + |\Psi^+\rangle(Z|\psi\rangle) + |\Psi^-\rangle(XZ|\psi\rangle)).$$

Như vậy, bằng cách đo 2 qubit dưới của $|\pi_0\rangle$ theo cơ sở Bell và tác động các cổng X, Z phù hợp (nhớ rằng X, Z có nghịch đảo là chính nó) lên qubit trên cùng ta có trạng thái của qubit trên cùng là $|\psi\rangle$ như mong đợi.

Giao thức trên thường được gọi là **dịch chuyển lượng tử** (quantum teleportation). Nhận xét

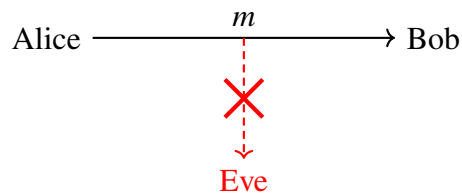
- Qubit vật lý Q không được dịch chuyển, chỉ có trạng thái của qubit là $|\psi\rangle$ được “dịch chuyển” vào qubit vật lý B .⁵
- Trạng thái ban đầu của qubit Q bị hủy (sau khi Alice đo), do đó không vi phạm định lý không nhân bản.
- Sau khi kết thúc giao thức thì Alice và Bob không còn chia sẻ cặp qubit vướng, nói cách khác, tài nguyên này “đã được dùng hết”.
- Quá trình dịch chuyển cần 2 bit kết quả đo Alice gửi cho Bob nên quá trình này không nhanh hơn tốc độ truyền bit cổ điển.
- Cũng như giao thức trước, ta bỏ qua các chi tiết thực tế. Chẳng hạn, Alice và Bob gặp nhau và chuẩn bị trước cặp qubit vướng, sau này, khi “đụng việc”, Alice mới dịch chuyển trạng thái của một qubit mình cần đến Bob mà không cần gặp Bob cũng không thể gửi một cách vật lý qubit đó đến Bob (mặc dù Alice có thể gửi các bit cổ điển đến Bob).

⁵có thể nói rằng qubit logic được chuyển, tương tự như khi chuyển tiền, tiền vật lý (tiền mặt) không thực sự được chuyển mà giá trị của tiền (tiền logic) được chuyển.

5.2 Trao đổi khóa lượng tử

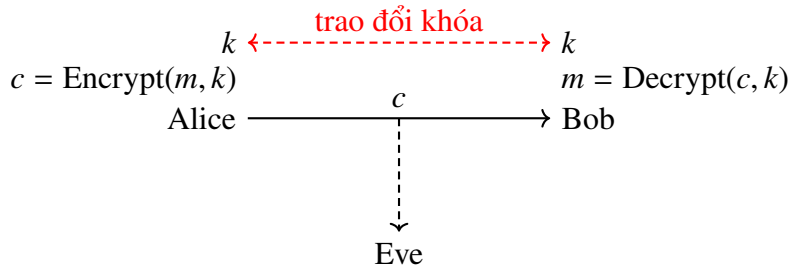
5.2.1 Mã hoá và trao đổi khóa

Alice muốn gửi tin nhắn riêng tư m cho Bob mà không muốn Eve (hay bất kỳ ai khác) biết m . Việc này không khó nếu Alice có thể gặp riêng Bob. Ngược lại, nếu Alice dùng các kênh truyền tin công cộng (như mạng Internet) thì Eve có thể “nhìn lén” m như Hình 5.1. Làm sao để “che giấu” m trong trường hợp này?



Hình 5.1: Eve có thể nhìn lén tin nhắn Alice gửi cho Bob.

Nếu Alice và Bob đồng thuận trước một **khóa bí mật** (secret key) chung là k thì Alice có thể **mã hóa** (encrypt) bản rõ (plaintext) m thành bản mã (ciphertext) $c = \text{Encrypt}(m, k)$, sau đó gửi c cho Bob. Alice khi nhận c có thể dùng khóa k để **giải mã** (decrypt) bản mã c thành tin nhắn ban đầu $m = \text{Decrypt}(c, k)$ như lược đồ ở Hình 5.2.⁶



Hình 5.2: Lược đồ mã hóa Alice dùng để gửi tin nhắn cho Bob.

Điều làm cho Eve không thể biết bản rõ m , dù có thể nhìn lén bản mã c , là không biết khóa k . Do đó khóa k cần phải được giữ bí mật nên hệ thống mã này thường được gọi là mã hóa khóa bí mật (secret-key encryption).⁷ Hơn nữa, việc đồng thuận

⁶mặc dù chung thuật ngữ tiếng Việt là mã hóa/giải mã nhưng encrypt/decrypt có nghĩa hơi khác so với encode/decode.

⁷cách mã hóa này còn được gọi là mã hóa đối xứng (symmetric encryption) vì dùng cùng một khóa để mã hóa và giải mã.

một (hay nhiều) khoá bí mật chung, còn gọi là **trao đổi khóa** (key distribution), là điều tối quan trọng mà Alice và Bob cần phải làm được.

Ví dụ 5.2.1. (Hệ mã OTP) Giả sử Alice muốn gửi tin nhắn “Hi” cho Bob, được viết dưới dạng chuỗi nhị phân 2 byte bằng mã ASCII là

$$m = 01001000 \ 01101001.$$

Alice và Bob đồng thuận khoá bí mật chung là chuỗi nhị phân cùng chiều dài gồm các bit được chọn ngẫu nhiên, chẳng hạn

$$k = 01101011 \ 10011011.$$

Alice và Bob có thể dùng **hệ mã One-Time Pad** (OTP) theo các bước sau

1. Alice mã hoá m thành bản mã

$$c = \text{Encrypt}(m, k) = m \oplus k = 00100011 \ 11110010.$$

2. Alice gửi bản mã c cho Bob.
3. Bob giải mã c để thu được tin nhắn

$$m = \text{Decrypt}(c, k) = c \oplus k = 01001000 \ 01101001$$

là chuỗi “Hi” ban đầu.

Lưu ý, để an toàn, khóa k chỉ được phép dùng một lần vì nếu Alice dùng cùng khóa k để mã hóa bản rõ m_1, m_2 được các bản mã $c_1 = m_1 \oplus k, c_2 = m_2 \oplus k$ thì Eve sau khi nhìn lên c_1, c_2 có thể tính

$$k' = c_1 \oplus c_2 = (m_1 \oplus k) \oplus (m_2 \oplus k) = (m_1 \oplus m_2) \oplus (k \oplus k) = m_1 \oplus m_2.$$

Nếu m_1 được tiết lộ (chẳng hạn, tin nhắn này không còn quan trọng sau một thời gian) thì Eve có thể dùng k' để biết m_2 vì

$$m_1 \oplus k' = m_1 \oplus m_1 \oplus m_2 = m_2.$$

Hệ mã OTP có tính bảo mật rất cao, tuy nhiên, không được dùng nhiều vì đòi hỏi khóa bí mật có chiều dài như tin nhắn. Điều này làm cho việc trao đổi khóa rất khó khăn. Các hệ mã hóa khóa bí mật khác như AES (Advanced Encryption Standard) chỉ yêu cầu khóa có kích thước ngắn (128, 192 hay 256 bit). \square

5.2.2 Hệ mã hoá khoá công khai

Một cách để Alice và Bob trao đổi khoá bí mật là sử dụng **hệ mã hoá khoá công khai** (public key cryptosystem), trong đó, khoá được chia làm 2 phần: một phần công khai dùng để mã hóa (gọi là khoá công khai - public key), và phần còn lại giữ riêng dùng để giải mã (được gọi là khoá riêng - private key).⁸ Cụ thể, Alice muốn gửi tin nhắn m cho Bob, thì tiến hành theo các bước

1. Sinh khoá (Key Generation): Bob tạo khoá công khai và khoá bí mật (của Bob), sau đó, Bob công bố khoá công khai của mình cho mọi người và giữ kín khoá bí mật của mình.
2. Mã hóa (Encryption): mọi người có thể dùng khoá công khai của Bob để mã hoá tin nhắn m thành bản mã c để gửi cho Bob.
3. Giải mã (Decryption): khi Bob nhận bản mã c , Bob dùng khoá bí mật của mình để giải mã thành tin nhắn m ban đầu.

Trong thực tế, vì các hệ mã hóa khóa bí mật có thể được tiến hành nhanh với chi phí rẻ hơn nhiều so với các hệ mã hóa khóa công khai nên mã hoá khóa công khai thường được dùng để trao đổi khóa bí mật chung. Sau đó, mã hoá khóa bí mật được dùng để mã hoá và giải mã tin nhắn.

Ví dụ 5.2.2. (Hệ mã RSA *) Hệ mã RSA⁹ gồm 3 bước sau

1. Sinh khóa: Bob thực hiện các bước sau
 - (a) Chọn 2 số nguyên tố lớn p, q và tính $n = pq$. Chiều dài tính theo bit của n được gọi là kích thước khóa (key length).
 - (b) Tính $\phi(n) = (p - 1)(q - 1)$.
 - (c) Chọn số nguyên e ngẫu nhiên thỏa $1 < e < \phi(n)$ và e nguyên tố cùng nhau với $\phi(n)$ (tức là $\gcd(e, \phi(n)) = 1$).
 - (d) Tính d thỏa $ed = 1 \bmod \phi(n)$, ta còn kí hiệu $d = e^{-1} \bmod \phi(n)$.
 - (e) Công khai khóa $pk = (n, e)$ và giữ bí mật khóa $sk = d$.
2. Mã hóa: Alice sử dụng khóa công khai $pk = (n, e)$ của Bob để mã hoá tin nhắn m thành bản mã

$$c = m^e \bmod n.$$

⁸cách mã hóa này còn được gọi là mã hóa bất đối xứng (asymmetric encryption) vì dùng các khóa khác nhau để mã hóa và giải mã.

⁹được đặt tên theo các tác giả đề xuất hệ là Rivest, Shamir, và Adleman.

3. Giải mã: Bob sử dụng khoá bí mật $sk = d$ để giải mã bản mã c thành tin nhắn ban đầu

$$m = c^d \bmod n.$$

Ví dụ, Bob sinh khoá bằng việc chọn và tính những thông số sau đây

1. Chọn $p = 17$, $q = 41$ và tính $n = pq = 17 \cdot 41 = 697$.
2. Tính $\phi(n) = (p - 1)(q - 1) = 16 \cdot 40 = 640$.
3. Chọn $e = 3$ nên $d = e^{-1} = 3^{-1} \pmod{640} = 427 \pmod{640}$.

Alice muốn gửi tin nhắn $m = 104$ cho Bob thì sử dụng khoá công khai gồm $e = 3$ và $n = 697$ để tính bản mã

$$c = m^e = 104^3 \pmod{697} = 603 \pmod{697}.$$

Bob sử dụng $d = 427$ để giải mã về tin nhắn

$$m = c^d = 603^{427} \pmod{697} = 104 \pmod{697}.$$

Tính đúng đắn của hệ mã RSA dựa trên định lý Euler (Bài tập 5.9). Tính bảo mật phụ thuộc vào “bài toán khó” là phân tích thừa số nguyên tố (phân tích $n = pq$) vì biết $\phi(n) = (p - 1)(q - 1)$ có thể tính được khoá bí mật $d = e^{-1} \bmod \phi(n)$. Tuy nhiên, với sự xuất hiện của máy tính lượng tử thì hệ mã RSA không còn an toàn do bài toán phân tích thừa số nguyên tố có thể được “giải nhanh” bằng thuật toán Shor (được tìm hiểu trong Phần 6.3). Hiện tại, kích thước khoá của RSA được NIST đưa ra để đảm bảo an toàn cho đến năm 2030 là 2048 bit. \square

5.2.3 Giao thức BB84

Năm 1984, Charles Bennett và Gilles Brassard đề xuất giao thức đầu tiên dùng tính toán lượng tử để trao đổi khóa, gọi là giao thức BB84. Tính an toàn của giao thức dựa trên định lý không nhân bản (Phần 4.5). Mấu chốt, Alice và Bob có thể phát hiện được việc Eve nhìn lén. Các bước hoạt động của giao thức này như sau

1. Alice gửi một dãy qubit ngẫu nhiên cho Bob.
 - (a) Alice chọn ngẫu nhiên 1 dãy bit.
 - (b) Với mỗi bit, Alice chọn ngẫu nhiên một trong hai cơ sở

$$X = \{|+\rangle, |-\rangle\}, \quad Z = \{|0\rangle, |1\rangle\}$$

và tạo qubit với trạng thái tương ứng theo qui tắc như bảng sau
(nếu bit là 0 và cơ sở là Z thì tạo qubit có trạng thái là $|0\rangle$, ...)

	0	1
Z	$ 0\rangle$	$ 1\rangle$
X	$ +\rangle$	$ -\rangle$

- (c) Alice tạo dãy qubit theo qui tắc trên cho từng bit và gửi dãy qubit cho Bob.
2. Bob tính dãy bit qua phép đo trên dãy qubit nhận được từ Alice.
- (a) Với mỗi qubit, Bob chọn ngẫu nhiên cơ sở X hoặc Z.
- (b) Bob thực hiện phép đo trên từng qubit theo cơ sở đã chọn.
- (c) Bob chuyển từ qubit về bit theo cách tương tự với cách Alice chuyển từ bit sang qubit.

$$\begin{cases} \{|0\rangle, |+\rangle\} \rightarrow 0 \\ \{|1\rangle, |-\rangle\} \rightarrow 1 \end{cases}$$

3. Alice và Bob đồng thuận dãy bit đại diện cho khoá bí mật.
- (a) Alice và Bob cùng công khai dãy cơ sở.
- (b) Từ đó, Alice và Bob chọn các bit mà tại đó cơ sở là giống nhau.
- (c) Ghép các bit đó tạo thành khoá bí mật chung cho Alice và Bob.

Ví dụ 5.2.3. Xét một trường hợp cụ thể của BB84 như trong Bảng 5.1. Alice chọn ngẫu nhiên dãy bit (dòng 1) và cơ sở (dòng 2) rồi gửi dãy qubit với các trạng thái tương ứng (dòng 3) cho Bob.

Dãy bit Alice chọn	1	0	0	0	1	0	1	1	1
Dãy cơ sở của Alice	X	Z	X	Z	Z	Z	X	Z	Z
Dãy qubit Alice gửi	$ -\rangle$	$ 0\rangle$	$ +\rangle$	$ 0\rangle$	$ 1\rangle$	$ 0\rangle$	$ -\rangle$	$ 1\rangle$	$ 1\rangle$
Dãy cơ sở của Bob	X	Z	<u>Z</u>	Z	<u>X</u>	Z	X	Z	<u>X</u>
Kết quả đo của Bob	$ -\rangle$	$ 0\rangle$	<u>$1\rangle$</u>	$ 0\rangle$	<u>$+\rangle$</u>	$ 0\rangle$	$ -\rangle$	$ 1\rangle$	<u>$-\rangle$</u>
Dãy bit của Bob	1	0	<u>1</u>	0	<u>0</u>	0	1	1	<u>1</u>
Khóa bí mật	1	0		0		0	1	1	

Bảng 5.1: Một ví dụ minh họa giao thức BB84.

Bob chọn ngẫu nhiên dãy cơ sở (dòng 4) thực hiện phép đo để được kết quả (dòng 5) và chuyển thành dãy bit (dòng 6). Lưu ý, nếu cơ sở mà Bob chọn không giống với Alice (được gạch dưới ở dòng 4) thì bit của Bob có thể không giống bit của

Alice. Chẳng hạn, nếu Alice chọn cơ sở Z và gửi $|1\rangle$ (ứng với bit 1) còn Bob chọn cơ sở X để đo thì sẽ được $|+\rangle$ hoặc $|-\rangle$ và do đó chuyển thành bit 0 hoặc 1.

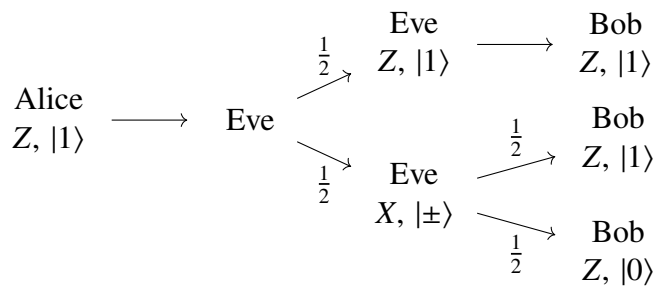
Sau cùng, Alice và Bob công khai dãy cơ sở và giữ lại các bit nơi cơ sở giống nhau làm khóa bí mật (dòng 7). Chẳng hạn, trong ví dụ này là dãy bit 100011. \square

Các tham số được công khai khi thực hiện giao thức BB84 gồm dãy qubit Alice gửi cho Bob và các dãy cơ sở mà Alice và Bob sử dụng. Eve có thể tấn công bằng cách lưu dãy qubit Alice gửi để khi Alice và Bob công khai dãy cơ sở thì Eve thực hiện các bước tương tự như Bob, từ đó tính được khóa bí mật chung của Alice và Bob. Tuy nhiên, việc lưu dãy qubit là không thể do vi phạm định lý không nhân bản.

Do vậy, Eve chỉ có thể “đo lén” các qubit Alice gửi rồi chuyển tiếp các qubit sau khi đo cho Bob. Eve tốt nhất nên chọn một trong hai cơ sở là X hoặc Z để đo với hi vọng dãy cơ sở mà mình chọn giống với Alice hoặc Bob càng nhiều càng tốt (lưu ý, lúc này Eve chưa biết các cơ sở mà Alice hay Bob dùng). Làm sao đảm bảo giao thức BB84 là an toàn khi Eve đo lén như vậy?

Rất may, do các tính chất của tính toán lượng tử, Alice và Bob có thể kiểm tra được Eve có đo lén khi đang thực hiện giao thức hay không bằng cách: sau khi tính khóa bí mật chung ở bước cuối, Alice và Bob chấp nhận công khai một vài bit của khóa để kiểm tra sự can thiệp của Eve. Ta sẽ thấy, nếu Eve đo lén thì xác suất bị phát hiện là rất cao. Cuối cùng, Alice và Bob dùng phần chưa được công khai của khóa để làm khóa bí mật (nếu không phát hiện Eve can thiệp).

Nếu công khai 1 bit chung thì xác suất Eve bị phát hiện là bao nhiêu? Giả sử bit của Alice là 1 và cơ sở của Alice (và Bob) chọn là Z (các trường hợp khác tương tự).



Nếu Eve đo lén theo cơ sở X (xác suất $\frac{1}{2}$) thì Bob sẽ nhận được $|+\rangle$ hoặc $|-\rangle$, khi đó, đo theo Z Bob sẽ được $|0\rangle$ hoặc $|1\rangle$ (xác suất đều là $\frac{1}{2}$) nên trường hợp bit của Bob là 0 khác với bit 1 của Alice xảy ra với xác suất

$$\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$$

Vì nếu dùng cùng cơ sở thì bit của Alice và Bob chắc chắn giống nhau (nếu Eve không can thiệp) nên xác suất Eve bị phát hiện là $\frac{1}{4}$ và xác suất không bị phát hiện là $\frac{3}{4}$.

Nếu Alice và Bob công khai n bit của khoá bí mật chung để kiểm tra sự can thiệp của Eve thì do các bit (qubit) được chọn là độc lập với nhau nên xác suất Eve bị phát hiện là (chỉ cần 1 bit không giống nhau thì Eve bị phát hiện)

$$P(\text{"Eve bị phát hiện"}) = 1 - \left(\frac{3}{4}\right)^n.$$

Xác suất này rất lớn khi n vừa phải, chẳng hạn $n = 50$ thì xác suất này khoảng 0.9999994.

Để sử dụng giao thức BB84, ta cần xây dựng kênh trao đổi thông tin lượng tử (quantum network) để có thể gửi và nhận (trạng thái) các qubit.

5.2.4 Giao thức B92

Năm 1992, Charles Bennett đề xuất một giao thức trao đổi khóa lượng tử đơn giản hơn BB84, gọi là giao thức B92. Các bước hoạt động của giao thức này như sau

1. Alice chọn ngẫu nhiên một dãy bit và với mỗi bit, Alice tạo qubit có trạng thái tương ứng như sau

$$\begin{cases} 0 \rightarrow |0\rangle \\ 1 \rightarrow |+\rangle \end{cases},$$

rồi gửi dãy qubit cho Bob.

2. Với mỗi qubit $|a\rangle$, Bob chọn ngẫu nhiên một trong hai cơ sở X hoặc Z , thực hiện phép đo và chuyển qubit về bit b tùy theo cơ sở được chọn như sau

- Cơ sở X :

- nếu $X|a\rangle = |-\rangle$ thì $b = 0$ (do $|a\rangle$ phải là $|0\rangle$),
- nếu $X|a\rangle = |+\rangle$ thì b không biết (do $|a\rangle$ có thể là $|0\rangle$ hoặc $|+\rangle$).

- Cơ sở Z :

- nếu $Z|a\rangle = |1\rangle$ thì $b = 1$ (do $|a\rangle$ phải là $|+\rangle$),
- nếu $Z|a\rangle = |0\rangle$ thì b không biết (do $|a\rangle$ có thể là $|0\rangle$ hoặc $|+\rangle$).

3. Bob công khai những vị trí mà Bob không biết (những vị trí Bob đo ra $|0\rangle$ hoặc $|+\rangle$ như trên). Alice và Bob đồng thuận dãy bit trên các vị trí Bob biết chắc.

Tương tự như giao thức BB84, Alice và Bob công khai một vài bit của dãy bit đồng thuận để kiểm tra Eve có đo lén hay không.

Ví dụ 5.2.4. Xét một trường hợp cụ thể của BB84 như trong Bảng 5.2. Alice chọn ngẫu nhiên dãy bit (dòng 1) rồi gửi dãy qubit với các trạng thái tương ứng (dòng 2) cho Bob.

Dãy bit Alice chọn	1	0	0	0	1	0	1	1	1
Dãy qubit Alice gửi	$ +\rangle$	$ 0\rangle$	$ 0\rangle$	$ 0\rangle$	$ +\rangle$	$ 0\rangle$	$ +\rangle$	$ +\rangle$	$ +\rangle$
Dãy cơ sở của Bob	Z	Z	X	X	X	Z	X	Z	Z
Kết quả đo của Bob	$ 1\rangle$	$ 0\rangle$	$ +\rangle$	$ -\rangle$	$ +\rangle$	$ 0\rangle$	$ +\rangle$	$ 1\rangle$	$ 0\rangle$
Dãy bit của Bob	1	?	?	0	?	?	?	1	?
Khóa bí mật	1			0				1	

Bảng 5.2: Một ví dụ minh họa giao thức B92.

Bob chọn ngẫu nhiên dãy cơ sở (dòng 3) thực hiện phép đo để được kết quả (dòng 4) và chuyển thành dãy bit (dòng 5). Bob công khai các vị trí không biết (?) và đồng thuận các bit còn lại với Alice (dòng 6). \square

5.2.5 Giao thức E91

Năm 1991, Artur Ekert đề xuất một giao thức trao đổi khóa lượng tử hoàn toàn khác BB84, gọi là giao thức E91. Giao thức này hoạt động dựa trên vướng lượng tử. Nhớ lại, nếu cặp qubit ở trạng thái vướng

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

thì khi đo qubit nào đi nữa thì qubit còn lại cũng sụp đổ về trạng thái giống qubit kia (cùng là $|0\rangle$ hay $|1\rangle$ nếu đo theo Z). Như vậy, bằng cách chia sẻ n cặp qubit ở trạng thái vướng $|\Phi^+\rangle$, khi cần khóa bí mật, Alice và Bob chỉ cần thực hiện phép đo (không quan trọng ai đo trước đo sau) sẽ được cùng dãy bit “thật sự” ngẫu nhiên.

Tuy nhiên, nếu các cặp qubit không còn ở trạng thái vướng (chẳng hạn do tác động môi trường hay do Eve đo lén các qubit) thì dãy bit của Alice và Bob có thể không giống nhau. Để kiểm tra việc này, giao thức có thể hoạt động như sau

1. Alice và Bob mỗi người giữ mỗi qubit của dãy cặp qubit ở trạng thái vướng $|\Phi^+\rangle$.

2. Alice và Bob mỗi người chọn ngẫu nhiên cơ sở X hoặc Z cho từng qubit để thực hiện phép đo trên dây qubit. Lưu ý, $|\Phi^+\rangle$ cũng có thể được viết lại là (\clubsuit)

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(|++\rangle + |--\rangle)$$

nên đo theo Z hay X thì Alice và Bob đều ra cùng kết quả.

3. Alice và Bob cùng công khai dãy cơ sở, từ đó, đồng thuận các bit nơi cơ sở giống nhau.

Bằng cách công khai một vài bit đồng thuận, Alice và Bob có thể phát hiện được các cặp qubit ban đầu có còn ở trạng thái vướng hay không, tương tự như giao thức BB84.

Ví dụ 5.2.5. Xét một trường hợp cụ thể của E91 như trong Bảng 5.3. Alice và Bob chọn ngẫu nhiên dãy cơ sở (dòng 1 và dòng 3) rồi thực hiện phép đo. Tại nơi cơ sở giống nhau (dòng 5) thì kết quả đo của Alice và Bob phải giống nhau (nếu cặp qubit tương ứng còn ở trạng thái vướng $|\Phi^+\rangle$). Cuối cùng, ở các nơi đồng thuận, Alice và Bob lấy các bit (dòng 6) từ qubit theo qui tắc

$$\begin{cases} \{|0\rangle, |+\rangle\} \rightarrow 0 \\ \{|1\rangle, |-\rangle\} \rightarrow 1 \end{cases}$$

□

Dãy cơ sở của Alice	X	Z	X	Z	X	Z	Z
Kết quả đo của Alice	$ +\rangle$	$ 0\rangle$	$ -\rangle$	$ 0\rangle$	$ -\rangle$	$ 1\rangle$	$ 1\rangle$
Dãy cơ sở của Bob	X	X	Z	Z	X	X	Z
Kết quả đo của Bob	$ +\rangle$	$ +\rangle$	$ 0\rangle$	$ 0\rangle$	$ -\rangle$	$ -\rangle$	$ 1\rangle$
Đồng thuận	✓			✓	✓		✓
Khóa bí mật	0			0	1		1

Bảng 5.3: Một ví dụ minh họa giao thức E91.

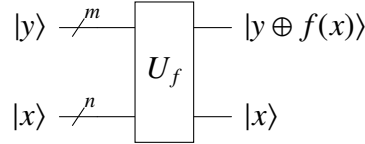
5.3 Các thuật toán truy vấn lượng tử

Phần này trình bày các thuật toán lượng tử được phát minh sớm nhất. Mặc dù đơn giản và không có ứng dụng thực tế rõ ràng nhưng các thuật toán này cung cấp các ý tưởng cho các thuật toán phức tạp và có ứng dụng thực tế trong các phần sau.

5.3.1 Oracle lượng tử và độ phức tạp truy vấn

Trong nhiều bài toán, ta cần làm việc với một ánh xạ $f : \mathbb{B}^n \rightarrow \mathbb{B}^m$ nhưng không biết gì về f ngoài việc có thể biết được giá trị $f(x) \in \mathbb{B}^m$ tại $x \in \mathbb{B}^n$ nào đó do ta chọn. Một cách hình tượng, **hộp đen** (black box) f được quản lý bởi một nhà tiên tri (**oracle**) mà khi có nghi vấn x ta có thể **truy vấn** (query) để được câu trả lời $f(x)$. Việc truy vấn thường rất tốn kém nên số lần truy vấn thường được dùng để đánh giá tính hiệu quả của thuật toán và thường được gọi là **độ phức tạp truy vấn** (query complexity).¹⁰

Với $f : \mathbb{B}^n \rightarrow \mathbb{B}^m$, trong tính toán lượng tử, oracle cho f thường được cho bằng cổng lượng tử U_f có đầu vào - đầu ra như sau



Ta không biết về cấu trúc mạch của U_f mà chỉ biết U_f là một phép toán lượng tử hợp lệ

$$|x\rangle|y\rangle \xrightarrow{U_f} |x\rangle|y \oplus f(x)\rangle, \quad \forall x \in \mathbb{B}^n, y \in \mathbb{B}^m.$$

m qubit ở trên được gọi là **thanh ghi trả lời** (answer register) và n qubit ở dưới được gọi là **thanh ghi đầu vào** (input register) vì để thực hiện truy vấn $x \in \mathbb{B}^n$ ta đặt $|x\rangle$ vào thanh ghi đầu vào và $|0^m\rangle$ vào thanh ghi trả lời, khi đó U_f cho kết quả $|0^m \oplus f(x)\rangle = |f(x)\rangle$ ở thanh ghi trả lời (thanh ghi đầu vào giữ nguyên trạng thái $|x\rangle$). U_f được gọi là **oracle lượng tử** (quantum oracle) cho f . Lưu ý, với mọi $f : \mathbb{B}^n \rightarrow \mathbb{B}^m$, ta đều có thể xây dựng mạch logic cho f và dùng kĩ thuật ở Phần 4.4 ta có thể xây dựng được U_f nên luôn có oracle lượng tử cho f .

Trường hợp hay gặp $m = 1$, thanh ghi trả lời chỉ gồm 1 qubit, được gọi là **qubit trả lời** (answer qubit). Với cách truy vấn thông thường, ta đặt $|0\rangle$ vào qubit trả lời, khi đó

$$U_f(|x\rangle|0\rangle) = |x\rangle|f(x)\rangle$$

nên có thể nói, kết quả $f(x)$ được “mã ở dạng bit” trong qubit trả lời. Rất thú vị,

¹⁰Hình tượng hơn nữa, ta cần đến một đền thờ (oracle) để xin lời dạy từ Đấng Tối Cao với lễ vật thịnh soạn.

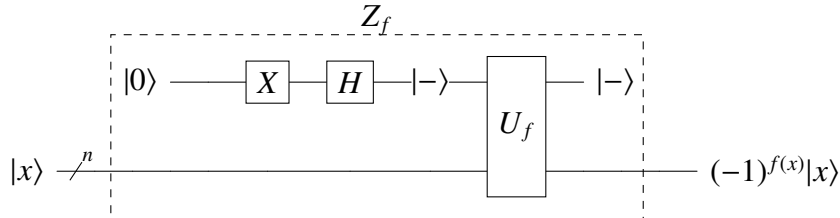
nếu đặt $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ vào qubit trả lời, ta có

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

Ta xem như qubit trả lời vẫn giữ trạng thái $|-\rangle$ nhưng thanh ghi đầu vào thay đổi trạng thái $|x\rangle$ thành $(-1)^{f(x)}|x\rangle$ nên có thể nói, kết quả $f(x)$ được “mã ở dạng pha” trong thanh ghi đầu vào. Hiện tượng này được gọi là “**đá pha**” (phase kickback) mà cơ bản thì $|x\rangle|-\rangle$ là vector riêng của U_f với trị riêng là $(-1)^{f(x)}$. Ta sẽ hiểu rõ hơn hiện tượng này ở Phần 6.2.2.

Dạng hoạt động này của U_f được gọi là **oracle pha** (phase oracle) mà đôi khi được kí hiệu là Z_f . Thông thường, ta bỏ qua qubit trả lời do nó không thay đổi (giữ trạng thái $|-\rangle$), khi đó ta xem hoạt động của Z_f như sau

$$|x\rangle \xrightarrow{Z_f} (-1)^{f(x)}|x\rangle, \quad x \in \mathbb{B}^n.$$



5.3.2 Thuật toán Deutsch

Thuật toán Deutsch (và mở rộng Deutsch-Jozsa) là một trong các thuật toán đầu tiên cho thấy ưu thế của tính toán lượng tử so với tính toán cổ điển.

Xét các phép toán trên 1 bit $f : \mathbb{B} \rightarrow \mathbb{B}$, từ Bảng 2.5 ta biết có 4 phép toán như vậy

Đặt

$$b_0 = f(0), \quad b_1 = f(1)$$

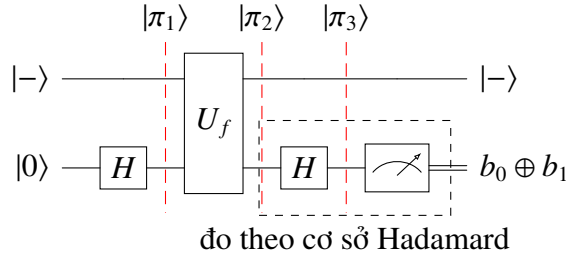
ta có thể xem rằng có chuỗi 2 bit b_1b_0 bí mật mà ta có thể truy vấn $f(x) = b_x$ với chỉ số x ($x = 0, 1$) để biết bit tương ứng có giá trị gì (là 0 hay 1).

x	f_1	f_2	f_3	f_4	
0	0	1	0	1	$\leftarrow b_0$
1	1	0	0	1	$\leftarrow b_1$
$b_0 \oplus b_1$	1	1	0	0	
	cân bằng		hằng		

Bảng 5.4: Hàm cân bằng và hàm hằng trên 1 bit.

Giả sử ta muốn biết $b_0 \oplus b_1$, từ Bảng 5.4, ta thấy rằng việc này cũng có nghĩa là ta muốn biết f là cân bằng hay không. Cân bằng nghĩa là có số lượng bit 0 và bit 1 như nhau. Ngược lại, không cân bằng, trong trường hợp này sẽ là hàm hằng, luôn là 0 hay 1. f_1, f_2 thì cân bằng trong khi f_3, f_4 thì không.

Rõ ràng, với tính toán cổ điển, ta cần biết chính xác 2 giá trị b_0 và b_1 để tính $b_0 \oplus b_1$, nghĩa là cần 2 lần truy vấn oracle của f . Rất thú vị, với tính toán lượng tử, ta chỉ cần 1 lần truy vấn để tính $b_0 \oplus b_1$ bằng **thuật toán Deutsch** như sau



Giả sử hàm f được cho bởi oracle lượng tử U_f dạng pha như trong phần trên, ý tưởng thuật toán Deutsch khá đơn giản. Sau đây là các bước biến đổi trạng thái của qubit 1 (vì qubit 0 có trạng thái $|-\rangle$ được giữ cố định nên ta không đề cập đến cho đỡ rối)

- Áp dụng cổng H vào $|0\rangle$ để tạo “tổ hợp đều”

$$|\pi_1\rangle = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle).$$

- Truy vấn oracle pha U_f với đầu vào là $|+\rangle$ để mã $f(x)$ vào các pha

$$\begin{aligned} |\pi_2\rangle &= \frac{1}{\sqrt{2}}[(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle] = \frac{1}{\sqrt{2}}[(-1)^{b_0}|0\rangle + (-1)^{b_1}|1\rangle] \\ &= (-1)^{b_0} \frac{1}{\sqrt{2}}[|0\rangle + (-1)^{b_1-b_0}|1\rangle]. \end{aligned}$$

Trông phức tạp nhưng nếu chia theo 2 trường hợp $b_0 = b_1$ và $b_0 \neq b_1$, ta có

$$|\pi_2\rangle = \begin{cases} (-1)^{b_0} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = (-1)^{b_0}|+\rangle \equiv |+\rangle & \text{nếu } b_0 = b_1 \\ (-1)^{b_0} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = (-1)^{b_0}|-\rangle \equiv |-\rangle & \text{nếu } b_0 \neq b_1 \end{cases}.$$

Lưu ý, pha chung không có ý nghĩa vật lý nên $-|+\rangle$ cũng là $|+\rangle$.

- Bấy chừ, ta chỉ cần đo qubit 1 theo cơ sở Hadamard là có thể phân biệt $|+\rangle$ hay $|-\rangle$ cũng là $b_0 = b_1$ hay không. Ta đã biết trong Phần ?, phép đo theo cơ sở Hadamard có thể được thực hiện bằng cách dùng cổng H để biến đổi cơ sở Hadamard về cơ sở tính toán và đo theo cơ sở tính toán,

$$|\pi_3\rangle = \begin{cases} |0\rangle & \text{nếu } b_0 = b_1 \\ |1\rangle & \text{nếu } b_0 \neq b_1 \end{cases}.$$

Cũng lưu ý, nếu xem 1 là đúng, 0 là sai thì $b_0 \oplus b_1$ chính là kết quả luận lý của khẳng định $b_0 \neq b_1$ (cũng là khẳng định f cân bằng). Tóm lại, kết quả khi thực hiện phép đo qubit 1 chính là $b_0 \oplus b_1$.

Có thể nói rằng, bằng cách dùng “tổ hợp đều” ($|+\rangle$) và oracle pha để mã đồng thời $f(x)$ vào các pha mà ta đã biến câu hỏi khó $b_0 \neq b_1$ trong tính toán cổ điển thành câu hỏi dễ trong cơ sở Hadamard của tính toán cổ điển. Và, kết quả, ta chỉ cần 1 lần truy vấn oracle thay vì 2! Cũng lưu ý, để trả lời câu hỏi này, thuật toán Deutsch đã “né” câu hỏi về giá trị của b_0, b_1 .

5.3.3 Thuật toán Deutsch-Jozsa

Thuật toán Deutsch-Jozsa mở rộng thuật toán Deutsch cho hàm n bit đầu vào (thay vì 1) $f : \mathbb{B}^n \rightarrow \mathbb{B}$. Cũng vậy, ta nói f là hàm hằng (constant) nếu f luôn cho đầu ra là 0 hoặc 1 với mọi đầu vào; f là hàm cân bằng (balanced) nếu số trường hợp đầu ra 0 và 1 là như nhau. Bảng 5.5 minh họa một vài hàm có $n = 2$ bit đầu vào: f_1, f_2 là 2 hàm hằng, f_3, f_4 là các hàm cân bằng, f_5 không là hàm hằng, cũng không là hàm cân bằng.

x_1	x_0	f_1	f_2	f_3	f_4	f_5
0	0	0	1	0	1	0
0	1	0	1	0	0	0
1	0	0	1	1	0	0
1	1	0	1	1	1	1

Bảng 5.5: Vài hàm có 2 bit đầu vào.

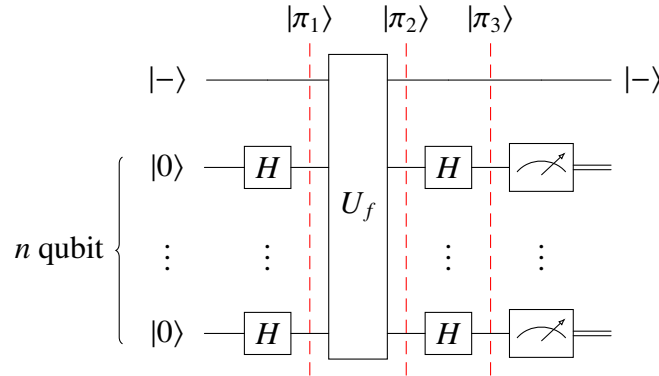
Bài toán đặt ra là, giả sử ta có $f : \mathbb{B}^n \rightarrow \mathbb{B}$ thuộc một trong hai loại: hàm cân bằng hoặc hàm hằng, xác định f thuộc thể loại nào.

Trong tính toán cổ điển, ta có thể thực hiện $2^{n-1} + 1$ lần truy vấn f (trong trường hợp tệ nhất) để xác định f có cân bằng hay không, vì, xui nhất khi ta truy vấn 2^{n-1} lần (phân nửa trường hợp) mà đều được cùng kết quả (nếu có 2 trường hợp ra kết quả khác nhau thì f phải là hàm cân bằng), thực hiện thêm một truy vấn thì nếu khác các lần trước thì là hàm cân bằng, ngược lại là hàm hằng.

Lời giải trên có độ phức tạp $O(2^n)$, hàm mũ theo số bit n . Sử dụng thuật toán ngẫu nhiên sau đây ta có độ phức tạp $O(1)$ với xác suất sai rất nhỏ. (Bài tập 5.20 kiểm tra tính đúng đắn của thuật toán này).

1. Chọn ngẫu nhiên đầu vào là một chuỗi n bit b .
2. Nếu $f(b)$ khác với các giá trị đầu ra trước đó thì dừng, báo f là hàm cân bằng.
3. Lặp lại Bước 1 t lần.
4. Dừng, báo f là hàm hằng.

Mở rộng thuật toán Deutsch, thuật toán Deutsch-Jozsa cho phép kiểm tra hàm f là có là hàm cân bằng (chắc chắn đúng) với chỉ 1 lần truy vấn oracle.



Tương tự thuật toán Deutsch, giả sử hàm f được cho bởi oracle lượng tử U_f dạng pha, các bước biến đổi trạng thái của n qubit đầu vào (ta cũng bỏ qua qubit 0 có trạng thái $|- \rangle$ không đổi)

- Áp dụng các cổng H vào các bit đầu vào $|0\rangle$ để tạo “tổ hợp đều”

$$\pi_1 = H^{\otimes n} |0\rangle^{\otimes n} = |+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$$

Bài tập 5.21 yêu cầu chứng minh đẳng thức cuối.

- Truy vấn oracle pha U_f để mã $f(x)$ vào các pha

$$|\pi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} |x\rangle$$

- Áp dụng các cổng H lần nữa được

$$|\pi_3\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} H^{\otimes n} |x\rangle.$$

Cho $x \in \{0,1\}^n$, ta có (Bài tập 5.22)

$$H^{\otimes n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} (-1)^{x \cdot z} |z\rangle, \quad (5.1)$$

trong đó, nếu $x = x_{n-1} \dots x_1 x_0$, $z = z_{n-1} \dots z_1 z_0$ thì

$$x \cdot z = x_{n-1} z_{n-1} + \dots + x_1 z_1 + x_0 z_0.$$

Do đó

$$\begin{aligned} |\pi_3\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} \left[(-1)^{f(x)} \frac{1}{\sqrt{2^n}} \left(\sum_{z \in \{0,1\}^n} (-1)^{x \cdot z} |z\rangle \right) \right] \\ &= \sum_{z \in \{0,1\}^n} \left(\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{f(x) + x \cdot z} \right) |z\rangle. \end{aligned} \quad (5.2)$$

- Ket cơ sở $|0\rangle^{\oplus n} = |0^n\rangle$ có amplitude trong $|\pi_3\rangle$ là

$$\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{f(x)}$$

do $z = 0^n$ thì $x \cdot z = 0$ (với mọi x).

Bấy chừ, nếu f là

- hàm hằng thì $f(x) = c$ với mọi x nên

$$\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} = \frac{1}{2^n} \cdot 2^n \cdot (-1)^c = (-1)^c.$$

Do đó, xác suất để đo được 0^n là $\left((-1)^c\right)^2 = 1$.

- hàm cân bằng thì có 2^{n-1} đầu vào thoả $f(x) = 0$ và 2^{n-1} đầu vào còn lại thoả $f(x) = 1$ nên

$$\sum_{x \in \{0,1\}^n} (-1)^{f(x)} = 0.$$

Do đó, xác suất để đo được 0^n là 0.

Như vậy, nếu kết quả đo là 0^n thì ta khẳng định f là hàm hằng, ngược lại, kết quả đo không là chuỗi 0^n thì f là hàm cân bằng.

5.3.4 Thuật toán Bernstein-Vazirani

Trước hết, cho $x = x_{n-1} \dots x_1 x_0, y = y_{n-1} \dots y_1 y_0 \in \mathbb{B}^n$ là 2 chuỗi nhị phân n bit, ta định nghĩa

$$x \cdot y = x_{n-1}y_{n-1} \oplus \dots \oplus x_1y_1 \oplus x_0y_0$$

là **tích vô hướng nhị phân** (binary dot product) của x, y . Ta thấy

$$x \cdot y = \begin{cases} 1 & \text{nếu } x_{n-1}y_{n-1} + \dots + x_1y_1 + x_0y_0 \text{ lẻ,} \\ 0 & \text{nếu } x_{n-1}y_{n-1} + \dots + x_1y_1 + x_0y_0 \text{ chẵn,} \end{cases}$$

nên có thể hiểu $x \cdot y$ là tính lẻ của các bit của x ở các chỉ số mà các bit của y là 1 (cũng là tính lẻ của các bit của y ở các chỉ số mà các bit của x là 1).

Bài toán đặt ra là: giả sử có hàm $f : \mathbb{B}^n \rightarrow \mathbb{B}$ với đầu ra

$$f(x) = s \cdot x$$

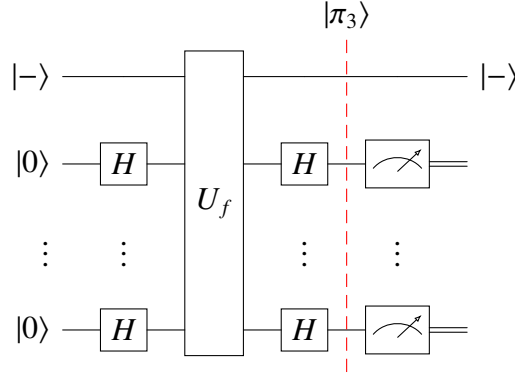
được tính theo tham số $s \in \mathbb{B}^n$ không biết trước. Tìm s .

Trong tính toán cổ điển, gọi $e^{(i)}$ là chuỗi n bit với bit thứ i ($i = 0, \dots, n-1$) là 1 và các bit khác là 0 thì từ nhận xét

$$f(e^{(i)}) = s_{n-1}e_{n-1}^{(i)} \oplus \dots \oplus s_1e_1^{(i)} \oplus s_0e_0^{(i)} = s_i$$

nên ta có thể truy vấn f với các chuỗi $e^{(i)}$ để có được các bit của s . Cách này cần n lần truy vấn.

Rất thú vị, với tính toán lượng tử ta chỉ cần 1 lần truy vấn. Thật vậy, giả sử f được cho bằng oracle U_f , bằng cách dùng mạch của thuật toán Deutsch-Josza



từ (5.2) ta có

$$\begin{aligned} |\pi_3\rangle &= \sum_{z \in \{0,1\}^n} \left(\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{f(x)+x \cdot z} \right) |z\rangle \\ &= \sum_{z \in \{0,1\}^n} \left(\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{s \cdot x + x \cdot z} \right) |z\rangle = \sum_{z \in \{0,1\}^n} \left(\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{(s \oplus z) \cdot x} \right) |z\rangle. \end{aligned}$$

Nhận xét, nếu $z = s$ thì $s \oplus z$ có bit thứ i là $s_i \oplus z_i = 0$ nên $s \oplus z = 0^n$. Do đó, amplitude của $|s\rangle$ trong $|\pi_3\rangle$ là

$$\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^0 = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} 1 = \frac{2^n}{2^n} = 1.$$

Như vậy, xác suất đo ra chuỗi s là 100%.

Ta thấy thuật toán cổ điển cần n lần truy vấn. Người ta cũng chứng minh được các thuật toán ngẫu nhiên cũng cần n lần truy vấn. Như vậy, độ phức tạp truy vấn của tính toán lượng tử là $O(1)$ còn tính toán cổ điển là $O(n)$. Thuật toán này thường được gọi là thuật toán Bernstein-Vaziran và là ví dụ cho thấy ưu thế của tính toán lượng tử.¹¹ Tuy nhiên, cả 2 đều thuộc lớp đa thức, thuật toán trong phần sau sẽ cho thấy ưu thế lớn hơn nữa.

5.3.5 Thuật toán Simon

Bài toán “Secret XOR Mask”: giả sử có hàm $f : \mathbb{B}^n \rightarrow \mathbb{B}^n$ thỏa

$$f(x) = f(y) \Leftrightarrow x = y \text{ hoặc } x \oplus s = y, \forall x, y \in \mathbb{B}^n$$

¹¹như mạch lượng tử cho thấy, đây thật ra là thuật toán Deutsch-Jozsa cho bài toán “tìm chuỗi s bí mật trong tích vô hướng nhị phân”. Bernstein-Vaziran là tên của thuật toán cho bài toán phức tạp hơn là “recursive Fourier sampling”.

với tham số $s \in \mathbb{B}^n$ không biết trước. Tìm s .

Nhận xét

- Nếu $s = 0^n$ thì f là hàm một-một (one-to-one), tức đơn ánh.
- Nếu $s \neq 0^n$ thì f là hàm hai-một (two-to-one), tức là có 2 đầu vào khác nhau cho cùng một đầu ra.

Bảng 5.6 nêu vài ví dụ hàm $\mathbb{B}^2 \rightarrow \mathbb{B}^2$ thỏa yêu cầu trên. f_1, f_2 có chuỗi bí mật $s = 00$ nên là các hàm đơn ánh. f_3, f_4 có chuỗi bí mật $s = 10$ nên có đầu ra giống nhau cho các cặp đầu vào 00, 10 và 01, 11. f_5 có chuỗi bí mật $s = 11$ nên có đầu ra giống nhau cho các cặp đầu vào 00, 11 và 01, 10.

x_1	x_0	f_1	f_2	f_3	f_4	f_5
0	0	00	01	00	11	01
0	1	01	10	01	10	10
1	0	10	11	00	11	10
1	1	11	00	01	10	01

Bảng 5.6: Vài hàm 2 bit thỏa bài toán Secret XOR Mask.

Cách tiếp cận cổ điển đối với bài toán Secret XOR Mask là tìm kiếm một đụng độ (collision), tức là tìm cặp x, y thỏa $x \neq y$ và $f(x) = f(y)$, khi đó

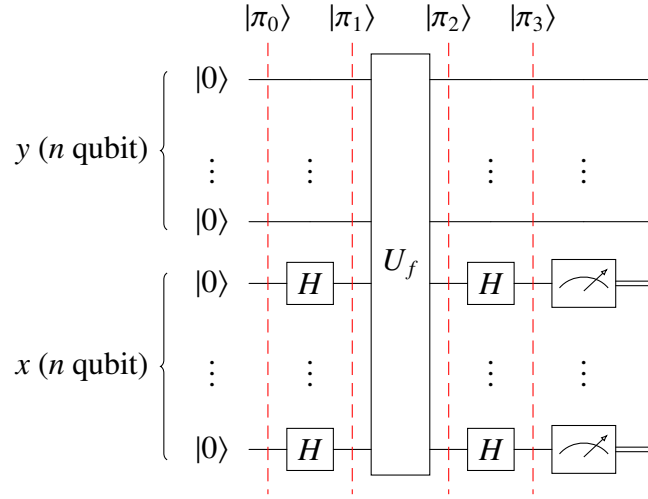
$$x \oplus y = x \oplus (x \oplus s) = (x \oplus x) \oplus s = 0^n \oplus s = s.$$

Trường hợp vét cạn, độ phức tạp để tìm được s là $O(2^{n-1} + 1)$ truy vấn. Người ta chứng minh được, dùng thuật toán ngẫu nhiên có thể tìm được s (với xác suất lớn) với $O\left(2^{\frac{n}{2}}\right)$ truy vấn. Như vậy, việc giải bài toán Secret XOR Mask với các thuật toán cổ điển yêu cầu độ phức tạp mũ theo n .

Rất thú vị, với tính toán lượng tử, ta có thể giải bài toán Secret XOR Mask với độ phức tạp đa thức theo n . Trước hết, hàm f được cho bằng oracle U_f trên hệ $2n$ qubit sao cho với mọi cặp $|x\rangle, |y\rangle$ với $x, y \in \mathbb{B}^n$, ta có

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

Sau đây là sơ đồ mạch, gọi là thuật toán Simon



Ban đầu cả 2 thanh ghi đều khởi động với các ket $|0\rangle^{\otimes n}$

$$|\pi_0\rangle = |0\rangle^{\otimes n} |0\rangle^{\otimes n}$$

Sau khi áp dụng các cổng Hadamard cho từng qubit ở thanh ghi x ta có

$$|\pi_1\rangle = H^{\otimes n}(|0\rangle^{\otimes n}) |0\rangle^{\otimes n} = |+\rangle^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle |0\rangle^{\otimes n}.$$

Sau khi áp dụng cổng U_f lên cặp thanh ghi x, y ta có

$$|\pi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} U_f(|x\rangle |0\rangle^{\otimes n}) = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle.$$

Sau khi áp dụng cổng Hadamard cho từng qubit ở thanh ghi x , từ (5.1) ta có

$$\begin{aligned} |\pi_3\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} H^{\otimes n}(|x\rangle) |f(x)\rangle = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} \left(\sum_{z \in \{0,1\}^n} (-1)^{x \cdot z} |z\rangle \right) |f(x)\rangle \\ &= \frac{1}{2^n} \sum_{z \in \{0,1\}^n} |z\rangle \left(\sum_{x \in \{0,1\}^n} (-1)^{z \cdot x} |f(x)\rangle \right) \end{aligned}$$

Khi đo thanh ghi x xác suất được chuỗi $z \in \{0,1\}^n$ là

$$P(z) = \left\| \frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{z \cdot x} |f(x)\rangle \right\|^2$$

Nhận xét

- Trường hợp f là hàm một-một ($s = 0^n$), ta có (Bài tập 5.25)

$$P(z) = \frac{1}{2^n}. \quad (5.3)$$

- Trường hợp f là hàm hai-một ($s \neq 0^n$), ta có (Bài tập 5.25)

$$\begin{aligned} P(z) &= \left\| \frac{1}{2^n} \sum_{\substack{y=f(a)=f(b) \\ a \oplus b = s}} \left((-1)^{z \cdot a} + (-1)^{z \cdot b} \right) |y\rangle \right\|^2 \\ &= \left\| \frac{1}{2^n} \sum_{y=f(a)} (-1)^{z \cdot a} (1 + (-1)^{z \cdot s}) |y\rangle \right\|^2 \\ &= \begin{cases} \frac{1}{2^{n-1}} & \text{nếu } z \cdot s = 0 \\ 0 & \text{nếu } z \cdot s = 1. \end{cases} \end{aligned} \quad (5.4)$$

Như vậy, mỗi lần “chạy” thuật toán Simon ta thu được chuỗi $z \in \mathbb{B}^n$ thỏa

$$z \cdot s = 0$$

(lưu ý, khi $s = 0^n$ thì $z \cdot s = 0$ với mọi z .)

Bằng cách chạy nhiều lần (mỗi lần chạy là một lần truy vấn f) ta thu đủ số lượng chuỗi z để có thể tìm được s . Bước này được thực hiện bằng các thuật toán cổ điển với $O(n)$ lần truy vấn f (Bài tập 5.26). Như vậy, đối với bài toán Secret XOR Mask, ta cần độ phức tạp mũ với tính toán cổ điển, nhưng chỉ cần độ phức tạp tuyến tính với tính toán lượng tử.

Tất cả các thuật toán trong phần này có chung một khuôn mẫu là: (1) áp dụng các cổng Hadamard để có trạng thái tổ hợp đều, (2) truy vấn oracle lượng tử để mã thông tin của f vào tất cả các đầu vào có thể (“song song lượng tử”), (3) áp dụng các cổng Hadamard lần nữa để “làm nổi” câu trả lời (giao thoa) và đo. Bảng 5.7 tóm tắt các thuật toán lượng tử trong phần này và ưu thế tương ứng của chúng về độ phức tạp truy vấn.

5.4 Thuật toán Grover

Phần này tiếp tục cho thấy ưu thế về độ phức tạp truy vấn của tính toán lượng tử trong bài toán tìm kiếm vét cạn. Thay vì cần $O(n)$ truy vấn với tính toán cổ điển, thuật toán Grover giúp giảm còn $O(\sqrt{n})$ với tính toán lượng tử, một ưu thế bậc hai (quadratic advantage). Hơn nữa, thuật toán Grover có thể được áp dụng trong rất nhiều bài toán khác nhau, mang lại một ưu thế phổ quát của tính toán lượng tử.

Bài toán	Số truy vấn cổ điển	Thuật toán lượng tử	Số truy vấn lượng tử
n bit Parity	n	Deutsch	$n/2$
Constant vs Balanced	$2^{n-1} + 1$ hoặc $O(1)$	Deutsch-Jozsa	1
Dot Product String	n	Bernstein-Vazirani	1
Secret XOR Mask	$O(2^{n/2})$	Simon	$O(n)$

Bảng 5.7: Các thuật toán truy vấn lượng tử và ưu thế về độ phức tạp truy vấn.

5.4.1 Tìm kiếm vét cạn

Tìm kiếm (searching) là việc, từ một tập các đối tượng, chỉ ra các đối tượng thỏa tiêu chí nào đó. Để tính toán, các đối tượng được mã thành các chuỗi nhị phân độ dài n và tiêu chí tìm kiếm được mô hình thành oracle $f : \{0, 1\}^n \rightarrow \{0, 1\}$ thỏa yêu cầu

$$f(x) = \begin{cases} 1, & x = w, \\ 0, & x \neq w. \end{cases}$$

Ta xem f như là hàm “nhận diện” chuỗi cần tìm: $f(x)$ là 1 báo rằng x là chuỗi cần tìm, ngược lại, là 0 báo rằng không phải. Để đơn giản, ta giả sử chỉ có đúng một chuỗi cần tìm là $w \in \{0, 1\}^n$, còn được gọi là **lời giải** (solution). Bài toán đặt ra là tìm w .¹²

Nếu ta không có bất kỳ thông tin nào khác về bài toán (như các cấu trúc đặc biệt trong cách mã đối tượng thành chuỗi nhị phân hay cấu trúc của hàm f), ta chỉ có thể truy vấn f để tìm w thì bài toán được gọi là **tìm kiếm vét cạn** (brute-force search) hay **tìm kiếm không cấu trúc** (unstructured search). Khi đó, rõ ràng, trong trường hợp xấu nhất ta phải kiểm tra tất cả các chuỗi có thể nên cần phải truy vấn $N = 2^n$ lần. Việc chọn ngẫu nhiên chuỗi để kiểm tra cũng cần trung bình $O(N)$ lần truy vấn để tìm thấy (Bài tập 5.27).

Ví dụ 5.4.1. Cho số thực k và a là danh sách gồm N số thực a_0, a_1, \dots, a_{N-1} , cần tìm vị trí i trong a ($i \in \{0, 1, \dots, N-1\}$) sao cho $a_i = k$. Để đơn giản, ta giả sử $N = 2^n$ và a chỉ có đúng một vị trí chứa k . Dùng chuỗi số nhị phân để mã cho vị trí, ta mô hình bài toán thành oracle $f : \{0, 1\}^n \rightarrow \{0, 1\}$ như sau

$$f(x) = \begin{cases} 1, & a_{[x]} = k, \\ 0, & a_{[x]} \neq k. \end{cases}$$

¹²các đối tượng cần tìm rất đa dạng trong thực tế, tuy nhiên, bằng cách mã hóa các đối tượng (và hàm f) dùng chuỗi nhị phân, ta luôn có thể đưa bài toán về dạng trên.

Chuỗi nhị phân mô tả vị trí chứa k của a chính là lời giải. Bài toán trên có thể được giải bằng thuật toán tìm kiếm vét cạn (Thuật toán 10). Rõ ràng, trong trường hợp xấu nhất (k nằm ở vị trí cuối cùng), ta cần truy vấn (kiểm tra $a_i = k$?) N lần.

Thuật toán 10 Thuật toán tìm kiếm vét cạn.

Input: a, k

Output: vị trí i sao cho $a_i = k$

```

1: for  $i = 0$  to  $N - 1$  do
2:   if  $a_i = k$  then return  $i$ 
3:   end if
4: end for

```

Lưu ý, trong vài trường hợp, ta có thể tìm kiếm nhanh hơn bằng cách khai thác cấu trúc đặc biệt của bài toán (khi đó bài toán không được gọi là tìm kiếm vét cạn). Chẳng hạn, ở trên, nếu các phần tử trong danh sách a được sắp xếp tăng dần, tức là $a_i \leq a_j$ nếu $i \leq j$ thì ta có thể dùng thuật toán **tìm kiếm nhị phân** (binary search) (Thuật toán 11). Với thuật toán này, trường hợp xấu nhất, ta chỉ cần truy vấn $n = \log N$ lần (Bài tập 5.28). Đây là mức độ giảm cực kỳ lớn, chẳng hạn với $N = 2^{64}$, thuật toán tìm kiếm vét cạn có thể tốn $N = 18446744073709551616$ lần, trong khi thuật toán tìm kiếm nhị phân chỉ tốn $n = \log N = 64$ lần. \square

Thuật toán 11 Thuật toán tìm kiếm nhị phân.

Input: a, k với a được sắp tăng dần

Output: vị trí i sao cho $a_i = k$

```

1:  $l = 1; r = N - 1$ 
2: while  $l \leq r$  do
3:    $i = \lfloor (l + r) / 2 \rfloor$ 
4:   if  $a_i = k$  then return  $i$ 
5:   else if  $a_i < k$  then
6:      $l = i + 1$ 
7:   else
8:      $r = i - 1$ 
9:   end if
10: end while

```

5.4.2 Thuật toán Grover

Thuật toán Grover được đề xuất bởi Lov Grover vào năm 1996 có khả năng tìm kiếm vét cạn với $O(\sqrt{N})$ lần truy vấn. Về cơ bản, thuật toán Grover thực hiện truy

vẫn trên trạng thái chồng chất tất cả các chuỗi để khai thác khả năng “song song lượng tử” (tương tự các thuật toán truy vấn oracle khác trong Phần 5.3). Quan trọng hơn, thuật toán Grover lặp lại việc này nhiều lần xen kẽ với các thao tác khác để tăng độ lớn amplitude của lời giải (hiện tượng giao thoa). Sau $O(\sqrt{N})$ lần lặp thì xác suất đo ra lời giải là rất cao.

Giả sử f được cho ở dạng oracle pha (Phần 5.3.1) U_f ,

$$U_f |x\rangle = (-1)^{f(x)} |x\rangle, \quad \forall x \in \{0, 1\}^n.$$

Thuật toán Grover bắt đầu với trạng thái “tổ hợp đều”

$$|s\rangle = |+\rangle^{\otimes n} = \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle$$

với $N = 2^n$. Nhớ rằng, $|s\rangle$ có thể được tạo bằng cách áp dụng các cổng Hadamard lên các qubit $|0\rangle$,

$$H^{\otimes n} |0\rangle^{\otimes n} = |+\rangle^{\otimes n}.$$

Với $w \in \{0, 1\}^n$ là lời giải duy nhất, ta có

$$\begin{aligned} |s\rangle &= \frac{1}{\sqrt{N}}(|w\rangle + \sum_{x \neq w} |x\rangle) = \frac{1}{\sqrt{N}}|w\rangle + \frac{\sqrt{N-1}}{\sqrt{N}} \underbrace{\frac{1}{\sqrt{N-1}} \sum_{x \neq w} |x\rangle}_{|r\rangle} \\ &= \frac{1}{\sqrt{N}}|w\rangle + \sqrt{\frac{N-1}{N}}|r\rangle. \end{aligned}$$

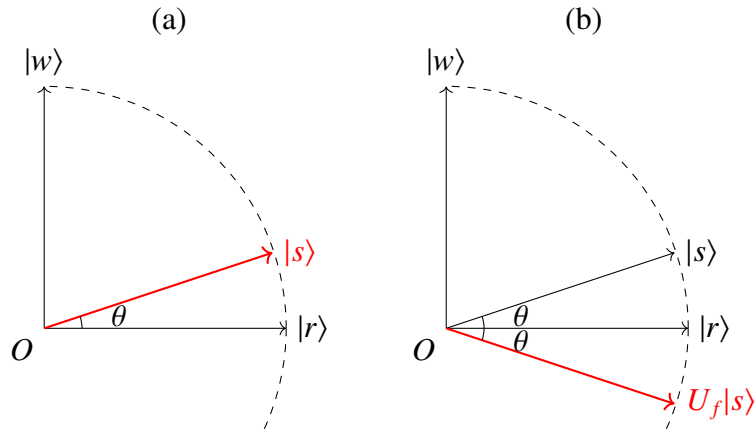
Đặt $\theta = \arcsin \frac{1}{\sqrt{N}}$ ta có

$$\sin \theta = \frac{1}{\sqrt{N}}, \quad \cos \theta = \sqrt{1 - \sin^2 \theta} = \sqrt{\frac{N-1}{N}}$$

nên $|s\rangle$ có thể được viết lại là

$$|s\rangle = \sin \theta |w\rangle + \cos \theta |r\rangle.$$

Vì $|w\rangle$ trực giao với mọi $|x\rangle$, $x \neq w$, $x \in \{0, 1\}^n$ nên $|w\rangle$ trực giao với $|r\rangle$. Ta cũng có, $|s\rangle$ là tổ hợp tuyến tính của $|w\rangle$ và $|r\rangle$ nên $|s\rangle$ nằm trên mặt phẳng chứa $|w\rangle$ và $|r\rangle$. Hơn nữa, $\langle s|r\rangle = \cos \theta$ nên góc giữa $|s\rangle$ và $|r\rangle$ là θ . Do đó, quan hệ giữa các ket $|s\rangle, |w\rangle, |r\rangle$ có thể được mô tả trực quan như Hình 5.3(a). Lưu ý, các ket này đều đã được chuẩn hóa (là các vector đơn vị). (♣)



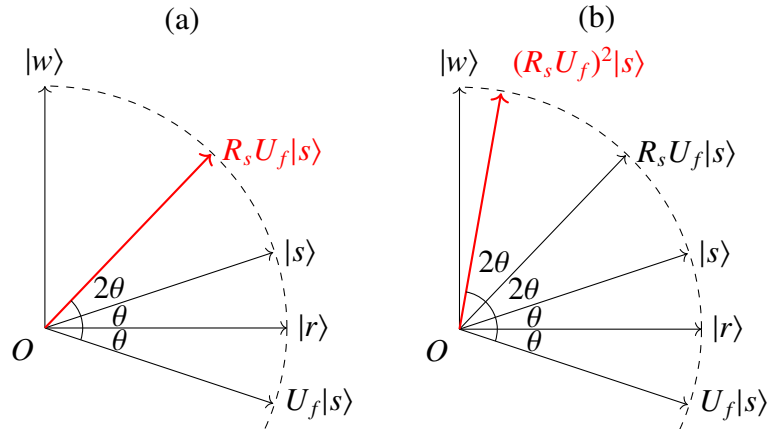
Hình 5.3: Minh họa các ket trong thuật toán Grover.

Tiếp theo, vì $f(x) = 1$ khi và chỉ khi $x = w$ nên truy vấn oracle pha U_f ta được trạng thái

$$U_f|s\rangle = (-1)^1 \sin \theta |w\rangle + (-1)^0 \cos \theta |r\rangle = -\sin \theta |w\rangle + \cos \theta |r\rangle$$

Ta thấy $U_f|s\rangle$ có cùng tọa độ trên trục $|r\rangle$ với $|s\rangle$ và đối trên trục $|w\rangle$ nên $U_f|s\rangle$ đối xứng với $|s\rangle$ qua $|r\rangle$. Trạng thái $U_f|s\rangle$ có thể được mô tả như Hình 5.3(b).

U_f chính là thao tác **phản xạ** (reflect) qua trục $|r\rangle$. Thuật toán Grover sử dụng thêm một thao tác phản xạ qua trục $|s\rangle$ (được mô tả chi tiết ở Phần dưới), kí hiệu là R_s . Kết quả tác động của U_f rồi đến R_s lên $|s\rangle$ được minh họa trong Hình 5.4(a). Rõ ràng, mỗi lần áp dụng U_f lên $|s\rangle$ rồi tiếp đó là R_s thì ta quay $|s\rangle$ một góc 2θ lại “gần” $|w\rangle$. Hình 5.4(b) minh họa kết quả khi áp dụng $R_s U_f$ 2 lần lên $|s\rangle$.



Hình 5.4: Minh họa các ket trong thuật toán Grover (tiếp Hình 5.3).

Bằng cách chọn số lần tác động phù hợp là t , ta có thể quay $|s\rangle$ lại “rất gần” $|w\rangle$

mà thực hiện phép đo khi đó thì ta được $|w\rangle$ với xác suất rất cao. Vậy chọn t bao nhiêu là phù hợp? Rõ ràng, như minh họa, ta cần chọn t sao cho

$$\theta + t(2\theta) \approx \frac{\pi}{2}$$

tức là

$$t \approx \frac{\pi}{4\theta} - \frac{1}{2}.$$

Lưu ý, t là số lần quay nên ta có thể chọn t là số nguyên làm tròn từ giá trị thực $\frac{\pi}{4\theta} - \frac{1}{2}$. Trường hợp “đẹp”, $\frac{\pi}{4\theta} - \frac{1}{2}$ nguyên, ta có thể quay $|s\rangle$ trùng với $|w\rangle$ nên xác suất thành công (đo ra w) là 1. Ngược lại, khi $\frac{\pi}{4\theta} - \frac{1}{2}$ không nguyên, ta không thể quay $|s\rangle$ trùng với $|w\rangle$ nên xác suất thành công không là 1. Tuy nhiên, vì xác suất thành công lớn nên bằng cách lặp lại vài lần chạy thì khả năng được w rất cao (Bài tập 5.32).

Tóm lại, thuật toán Grover có thể được mô như trong Thuật toán 12.

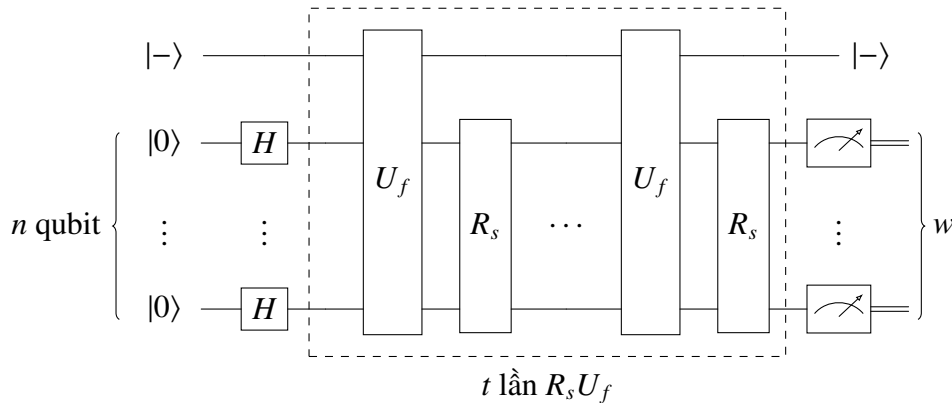
Thuật toán 12 Thuật toán Grover.

Input: oracle pha U_f

Output: w với $P(f(w) = 1)$ cao

- 1: khởi động với trạng thái $|\psi\rangle = |s\rangle = |+\rangle^{\otimes n} = H^{\otimes n} |0\rangle^{\otimes n}$
 - 2: t là giá trị nguyên làm tròn của $\frac{\pi}{4\theta} - \frac{1}{2}$ với $\theta = \arcsin \frac{1}{\sqrt{N}}$
 - 3: **for** $i = 1$ **to** t **do**
 - 4: $|\psi\rangle = R_s U_f |\psi\rangle$
 - 5: **end for**
 - 6: đo $|\psi\rangle$ được w
 - 7: **return** w
-

Thuật toán 12 cũng có thể được mô tả bằng sơ đồ mạch sau



Khi $N = 2^n$ rất lớn, ta có

$$\theta = \arcsin \frac{1}{\sqrt{N}} \approx \frac{1}{\sqrt{N}}$$

nên

$$t \approx \frac{\pi}{4} \sqrt{N} - \frac{1}{2} \approx \frac{\pi}{4} \sqrt{N}.$$

Lưu ý, mỗi lần xoay (áp dụng $R_s U_f$), ta thực hiện một lần truy vấn, nên thuật toán Grover cần $O(\sqrt{N})$ lần truy vấn. Đã có chứng minh cho thấy tính toán lượng tử không thể tìm kiếm vét cạn nhanh hơn $O(\sqrt{N})$ lần truy vấn nên thuật toán Grover đạt tối ưu tiệm cận. Thuật toán Grover cũng được vận dụng và mở rộng theo nhiều cách, chẳng hạn trong **khuếch đại biên độ** (amplitude amplification), là kỹ thuật tổng quát để tăng xác suất thành công của nhiều bài toán.

5.4.3 Phản xạ qua trạng thái tổ hợp đều

Để hoàn chỉnh thuật toán Grover, ta cần “cài đặt” R_s . Trước hết, vì R_s là thao tác phản xạ qua trục $|s\rangle$ nên từ (1.7) R_s có thể được viết là

$$R_s = 2|s\rangle\langle s| - I.$$

Từ định nghĩa của $|s\rangle$ ta có

$$|s\rangle = |+\rangle^{\otimes n} = H^{\otimes n} |0\rangle^{\otimes n}, \quad \langle s| = \langle 0|^{\otimes n} (H^\dagger)^{\otimes n} = \langle 0|^{\otimes n} H^{\otimes n}$$

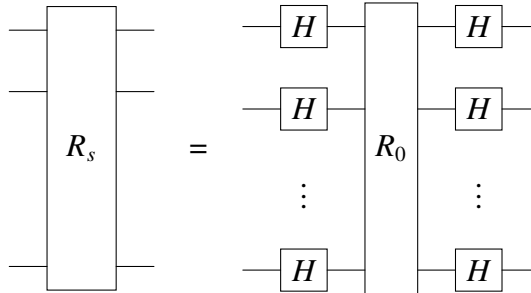
do $H^\dagger = H$. Hơn nữa, ta cũng có

$$I = I_{2^n} = I_2^{\otimes n} = (H^\dagger H)^{\otimes n} = (HH)^{\otimes n} = H^{\otimes n} H^{\otimes n}.$$

Do đó

$$R_s = 2H^{\otimes n} |0\rangle^{\otimes n} \langle 0|^{\otimes n} H^{\otimes n} - H^{\otimes n} H^{\otimes n} = H^{\otimes n} \underbrace{(2|0^n\rangle\langle 0^n| - I)}_{R_0} H^{\otimes n}.$$

Như vậy, ta có thể cài đặt R_s như mạch sau



Ta đã biết $H^{\otimes n}$ là phép chuyển cơ sở biến $|0^n\rangle$ thành $|s\rangle = |+\rangle^{\otimes n}$ và ngược lại. Do đó, một cách trực quan, ta thấy rằng để phản xạ qua $|s\rangle$, ta đưa về hệ trục $|0^n\rangle$, phản xạ qua $|0^n\rangle$, rồi đưa về hệ trục cũ.

Ta cũng thấy rằng

$$R_0 = 2|0^n\rangle\langle 0^n| - I$$

là thao tác phản xạ qua ket $|0^n\rangle$. Như vậy

$$\begin{cases} R_0 |0^n\rangle &= |0^n\rangle, \\ R_0 |x\rangle &= -|x\rangle, \forall x \in \{0, 1\}^n, x \neq 0^n. \end{cases}$$

Tức R_0 giữ nguyên $|0^n\rangle$ nhưng lật dấu (lật pha) các vector khác trong cơ sở tính toán. R_0 có thể được cài đặt như trong Bài tập 4.35.

Bài tập 5.29 cho thấy cách cài đặt khác của R_0 dùng phiên bản lượng tử của các cổng logic.

Bài tập

5.1 Vẽ mạch cụ thể và kiểm tra tính đúng đắn của mã đậm đặc khi Alice muốn gửi cho Bob thông điệp

- (a) 00 (b) 01 (c) 10 (d) 11

5.2 Mở rộng giao thức mã đậm đặc để Alice gửi thông điệp là 1 byte (chuỗi 8 bit) cho Bob. Kiểm tra tính đúng đắn khi Alice muốn gửi byte 00011011 cho Bob.

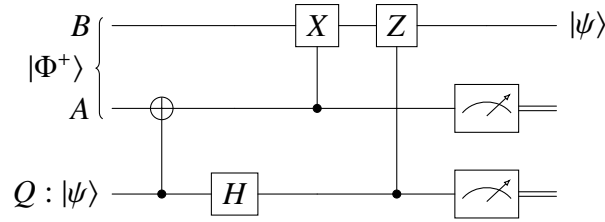
5.3 Nếu Alice và Bob chia sẻ cặp qubit ở trạng thái vướng không phải là $|\Phi^+\rangle$ mà là trạng thái Bell khác ($|\Psi^+\rangle$, $|\Phi^-\rangle$, $|\Psi^-\rangle$) thì làm thế nào để thực hiện

- (a) mã đậm đặc. (b) dịch chuyển lượng tử.

5.4 Trong giao thức dịch chuyển lượng tử, chứng minh trạng thái bắt đầu $|\pi_0\rangle$ có thể được viết lại là

$$|\pi_0\rangle = |\psi\rangle|\Phi^+\rangle = \frac{1}{2}(|\Phi^+\rangle|\psi\rangle + |\Phi^-\rangle(X|\psi\rangle) + |\Psi^+\rangle(Z|\psi\rangle) + |\Psi^-\rangle(XZ|\psi\rangle)).$$

5.5 Chứng minh rằng mạch sau



cũng thực hiện được dịch chuyển lượng tử.

Lưu ý, so với mạch cho ở Phần 5.1.3, trong mạch này, phép đo được thực hiện sau cùng (tổng quát, **nguyên lý đo trì hoãn** (deferred measurement principle) nói rằng phép đo có thể được thực hiện sau cùng).

5.6 Trong giao thức dịch chuyển lượng tử, giả sử qubit Q ở trạng thái vướng với qubit R là $|\Psi^+\rangle$. Chứng minh rằng: sau khi dịch chuyển, qubit B sẽ có “vai trò như” Q , cụ thể B, R có trạng thái vướng $|\Psi^+\rangle$.

5.7 Nghiên cứu các cách dùng trạng thái vướng 3 qubit

$$|\text{GHZ}\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

để thực hiện dịch chuyển lượng tử.

5.8 (Hệ mã Caesar) Giả sử tin nhắn là chuỗi gồm các chữ cái tiếng Anh. Ta đánh số các chữ cái theo thứ tự từ 0

$$A \leftrightarrow 0, B \leftrightarrow 1, \dots, Z \leftrightarrow 25.$$

Để mã hóa chữ cái x ta “xoay vòng phải” k vị trí, tức là

$$\text{Encrypt}(x, k) = (x + k) \bmod 26.$$

Ngược lại, để giải mã, ta “xoay vòng trái” k vị trí, tức là

$$\text{Decrypt}(x, k) = (x - k) \bmod 26.$$

Chẳng hạn, với $k = 3$, chữ cái Z (số thứ tự 25) được mã thành chữ cái C (số thứ tự 2) và C được giải mã thành Z.

Tin nhắn m gồm nhiều chữ cái được mã hóa và giải mã bằng cách trên cho từng chữ cái. Hệ thống mã hóa này được gọi là **mã Caesar** (Caesar cipher) với khóa bí mật là k .

(a) Mã hóa và giải mã tin nhắn

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

với khóa $k = 3$.

(b) Tương tự Câu (a) với khóa $k = 20$.

(c) Cho bản mã

QEB NRFZH YOLTK CLU GRJMP LSBO QEB IXWV ALD

Bản rõ là gì? Khóa k là bao nhiêu?

(d) Tìm vài cách tấn công mã Caesar (tức là tìm cách biết bản rõ từ bản mã mà không biết khóa).

5.9 Chứng minh định lý Euler: Cho số nguyên dương $n \in \mathbb{Z}$, với mọi số nguyên dương $a \in \mathbb{Z}_n$ thỏa $\gcd(a, n) = 1$ thì

$$a^{\phi(n)} = 1 \pmod{n}.$$

Do vậy, nếu $\gcd(m, n) = 1$ thì

$$m = c^d \pmod{n} = m^{ed} \pmod{n} = m^{k \cdot \phi(n) + 1} = m \pmod{n}.$$

5.10 Giả sử Eve không đo lén, nếu Alice và Bob cần đồng thuận khóa có độ dài n thì Alice cần chọn ngẫu nhiên dãy bit có độ dài khoảng bao nhiêu khi

(a) Dùng giao thức BB84.

(b) Dùng giao thức B92.

5.11 Giả sử các cặp qubit vẫn ở trạng thái vướng $|\Phi^+\rangle$ trong giao thức E91, nếu Alice và Bob cần đồng thuận khóa có độ dài n thì cần khoảng bao nhiêu cặp?

5.12 Cần sửa đổi giao thức E91 như thế nào nếu các cặp qubit ở trạng thái vướng

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$

thay vì $|\Phi^+\rangle$.

5.13 Phân tích xác suất Eve bị phát hiện nếu can thiệp trong giao thức B92 và E91.

5.14 Cho $f : \mathbb{B} \rightarrow \mathbb{B}$ với

$$f(x) = \bar{x}, \quad x \in \mathbb{B}.$$

(a) Xây dựng oracle lượng tử U_f cho f .

(b) Xây dựng oracle pha Z_f cho f .

(c) Cho biết hoạt động của U_f, Z_f trên trạng thái đầu vào là $|+\rangle$.

5.15 Cho $f : \mathbb{B}^2 \rightarrow \mathbb{B}$ với

$$f(x, y) = x \oplus y, \quad x, y \in \mathbb{B}.$$

(a) Xây dựng oracle lượng tử U_f cho f .

(b) Xây dựng oracle pha Z_f cho f .

(c) Cho biết hoạt động của U_f, Z_f trên trạng thái đầu vào là $|+\rangle^{\otimes 2}$.

5.16 Cho oracle lượng tử U_f . Điều gì xảy ra nếu ta truy vấn U_f với qubit trả lời được đặt là $|+\rangle$.

5.17 Làm lại Bài tập 5.14 cho $f : \mathbb{B}^2 \rightarrow \mathbb{B}$ với

$$f(x, y) = x \oplus y, \quad x, y \in \mathbb{B}.$$

5.18 Kiểm tra tính đúng đắn của thuật toán Deutsch ứng với các trường hợp của f cho trong Bảng 5.4

(a) f_1 .

(b) f_2 .

(c) f_3 .

(d) f_4 .

5.19 Xét chuỗi nhị phân n bit $b = b_{n-1} \dots b_1 b_0$ ta gọi

$$b_0 \oplus b_1 \oplus \dots \oplus b_{n-1}$$

là tính lẻ (parity) của b . Lưu ý, giá trị này là 1 nếu b có số lượng lẻ các bit 1.

(a) Giả sử n chẵn, đặt $c_i = b_{2i} \oplus b_{2i+1} (i = 0, \dots, n/2 - 1)$, chứng minh

$$c_0 \oplus c_1 \oplus \dots \oplus c_{n/2-1}$$

là tính lẻ của b .

(b) Từ Câu (a), thiết kế thuật toán mở rộng thuật toán Deutsch để tính tính lẻ của b với ít lần truy vấn oracle hơn thuật toán cổ điển.

5.20 Kiểm tra tính đúng đắn (và thiết kế) của thuật toán ngẫu nhiên xác định tính lẻ.

5.21 Chứng minh

$$(|+\rangle)^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle.$$

5.22 Cho $x \in \{0, 1\}^n$, chứng minh

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

trong đó, nếu $x = x_{n-1} \dots x_1 x_0$, $z = z_{n-1} \dots z_1 z_0$ thì

$$x \cdot z = x_{n-1}z_{n-1} + \dots + x_1z_1 + x_0z_0.$$

5.23 Vẽ mạch cụ thể và kiểm tra tính đúng đắn của thuật toán Deutsch-Jozsa cho các trường hợp hàm ở Bảng 5.5

- (a) f_1 . (b) f_2 . (c) f_3 . (d) f_4 .

5.24 Vẽ mạch cụ thể và kiểm tra tính đúng đắn của thuật toán Bernstein-Vazirani cho các trường hợp

- (a) $n = 2, s = 00$. (c) $n = 3, s = 010$.
(b) $n = 3, s = 000$. (d) $n = 4, s = 1111$.

5.25 Trong phần thuật toán Simon, chứng minh các đẳng thức

- (a) (5.3). (b) (5.4).

5.26 Bài tập này nghiên cứu phần xử lý dùng tính toán cổ điển của thuật toán Simon. Nhận xét

- Mỗi lần thực hiện phần xử lý dùng tính toán lượng tử của thuật toán Simon, ta chỉ thu được một $z \in \mathbb{B}^n$ thỏa $z \cdot s = 0$ (dù $s = 0$ hay $s \neq 0$). Do đó, ta cần thực hiện nhiều lần để thu được một hệ phương trình tuyến tính, từ đó giải hệ này sẽ thu được nghiệm s duy nhất (nếu nghiệm s thu được không thỏa $f(0^n) = f(s)$ thì $s = 0^n$).
- Để giải hệ phương trình tuyến tính gồm n phương trình trên n ẩn, ta có thể dùng thuật toán **khử Gauss** (Gaussian elimination) của tính toán cổ điển với độ phức tạp là $O(n^3)$.¹³ Lưu ý, ở đây, thay vì tính toán trên số thực, ta tính toán trên bit với phép toán modulo 2, tức là $1 + 1 = 0$ và $-1 = 1$.
- Để bảo đảm hệ phương trình tuyến tính có nghiệm s duy nhất, ta cần có đủ n giá trị z “độc lập tuyến tính”. Ta có được điều này với xác suất rất cao khi chạy $O(n)$ lần phần xử lý dùng tính toán lượng tử của thuật toán Simon.

¹³tham khảo https://en.wikipedia.org/wiki/Gaussian_elimination.

Ví dụ, dùng thuật toán Simon cho hàm f với $n = 3$ và $s = s_2 s_1 s_0$.

- (a) Có bao nhiêu $|z\rangle$ thỏa $z \cdot s = 0$?
- (b) Chạy 3 lần mạch Simon thu được 3 chuỗi z , tính xác suất 3 chuỗi này khác nhau?
- (c) Chạy 3 lần mạch Simon thu được 3 chuỗi là 011, 110, 010. Tìm s khi đó.

5.27 Trong bài toán tìm kiếm vết cạn, tính kì vọng số lần truy vấn để tìm thấy lời giải nếu

- (a) Mỗi lần truy vấn ta chọn ngẫu nhiên một chuỗi nhị phân (độ dài n).
- (b) Mỗi lần truy vấn ta chọn ngẫu nhiên một chuỗi nhị phân chưa được kiểm tra trước đó.

5.28 Kiểm tra tính đúng đắn và độ phức tạp của thuật toán tìm kiếm nhị phân trong Thuật toán 11.

5.29 Ta thấy rằng R_0 , thao tác phản xạ qua $|0^n\rangle$, có thể cài đặt như oracle pha U_f với $f : \{0, 1\}^n \rightarrow \{0, 1\}$ được định nghĩa cho $x \in \{0, 1\}^n$ là

$$f(x) = \begin{cases} 0 & x = 0^n \\ 1 & x \neq 0^n \end{cases}.$$

Quan sát kỹ hơn, ta thấy f chính là phép OR bit

$$\text{OR}(x_{n-1} \dots x_1 x_0) = \sum_{i=0}^{n-1} x_i \pmod{2}.$$

Như vậy bằng cách dùng các phiên bản lượng tử cho các cổng logic trong mạch logic cài đặt phép OR bit ta có thể cài đặt R_0 .

Cài đặt cụ thể R_0 theo cách này trong các trường hợp

- (a) $n = 1$.
- (b) $n = 2$.
- (c) $n = 4$.

5.30 Tính số lần lặp tối ưu t và xác suất đo ra lời giải khi đó của thuật toán Grover nếu

- (a) $N = 2^1$.
- (b) $N = 2^2$.
- (c) $N = 2^4$.
- (d) $N = 2^n$.

5.31 Vẽ biểu đồ mô tả xác suất thành công của thuật toán Grover, $P(f(w) = 1)$, theo số lần lặp (số lần áp dụng $R_s U_f$) ứng với các trường hợp ở Bài tập 5.30 và kiểm chứng số lần lặp tối ưu t .

5.32 Trong thuật toán Grover (Thuật toán 12), ta chỉ cần kết quả trả về w có xác suất cao là lời giải ($f(w) = 1$). Từ đó, bằng cách chạy thuật toán “vài lần” ta có thể tìm ra lời giải bằng 2 chiến lược sau

- (i) Nếu có thể truy vấn f , ta có thể chạy nhiều lần và chọn w cho $f(w) = 1$ nhiều lần nhất.
- (ii) Nếu không truy vấn f ta có thể chạy nhiều lần và chọn kết quả đồng thuận nhất (giống nhau nhiều lần nhất).

Xét từng chiến lược, nếu xác suất đo ra lời giải trong thuật toán Grover là 70% ($P(f(w) = 1) = 0.7$) thì

- (a) xác suất tìm được lời giải khi chạy thuật toán Grover 10 lần là bao nhiêu?
- (b) để xác suất tìm được lời giải nhỏ hơn 1 phần trăm thì cần chạy thuật toán Grover bao nhiêu lần?

5.33 Cài đặt hoàn chỉnh thuật toán Grover và thực hiện các tính toán cho thấy hoạt động của thuật toán khi lời giải là 0^n trong các trường hợp

- (a) $n = 1$.
- (b) $n = 2$.
- (c) $n = 4$.

5.34 Thuật toán Grover (Thuật toán 12) được thiết kế cho trường hợp bài toán tìm kiếm có đúng 1 lời giải. Hiệu chỉnh thuật toán cho trường hợp có số lượng lời giải

- (a) không quá 1.
- (b) đúng 2.
- (c) không biết.

Chương 6

Các thuật toán lượng tử nâng cao

Chương này trình bày các thuật toán lượng tử phức tạp hơn với nhiều ứng dụng trong thực tế. Các thuật toán này cho thấy ưu thế “đáng kể” của tính toán lượng tử so với tính toán cổ điển. Đặc biệt, thuật toán Shor cho phép giải hiệu quả bài toán phân tích số nguyên, một bài toán “được xem” là khó với tính toán cổ điển.

6.1 Biến đổi Fourier lượng tử

Phần này tìm hiểu biến đổi Fourier lượng tử (Quantum Fourier Transform, QFT) cùng cách cài đặt mạch hiệu quả. QFT là thành phần cơ bản của nhiều thuật toán khác như ước lượng pha và phân tích số nguyên. Có thể nói, QFT cùng với ước lượng pha là các thuật toán quan trọng nhất của tính toán lượng tử, mang lại ưu thế cấp mũ cho nhiều bài toán.

6.1.1 Mã pha và biến đổi Fourier lượng tử

Một cách rất quan trọng để giải nhiều bài toán là biến đổi nó về dạng mà theo đó bài toán sẽ rõ ràng và dễ giải hơn. Trong tính toán lượng tử, điều đó thường là việc chọn đúng cơ sở để làm việc. Sau khi hệ thống lượng tử được thiết lập thì cơ sở tính toán là cơ sở “tự nhiên” mà bài toán bắt đầu. Cơ sở này có thể phù hợp với vài bài toán nhưng thông thường, ta cần đưa về cơ sở trực chuẩn khác phù hợp hơn với bài toán.

Ví dụ điển hình với hệ 1 qubit là cơ sở $X = \{|+\rangle, |-\rangle\}$. Cơ sở này hay được dùng vì các vector cơ sở là “tổ hợp đều” của các vector cơ sở tính toán, hơn nữa, thông tin

về pha cũng được mã trong các vector này

$$\begin{aligned} |+\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \\ |-\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{2}(|0\rangle + e^{2\pi i \frac{1}{2}} |1\rangle). \end{aligned}$$

Để mở rộng cho trường hợp hệ n qubit,¹ đặt $N = 2^n$, trước hết nếu $\theta \in [0, 1)$ là một pha chọn trước, ta nhận thấy

$$|\phi_\theta\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i k \theta} |j\rangle$$

là trạng thái tổ hợp đều của các vector cơ sở tính toán, hơn nữa θ được mã vào pha tương đối của các amplitude. Lưu ý, $e^{2\pi i k \theta} = (e^{2\pi i \theta})^k$ là kết quả của việc quay số phức $e^{2\pi i \theta}$ đi k lần quanh đường tròn đơn vị với góc quay mỗi lần là $2\pi\theta$.

Để chọn đủ N ket dạng này làm cơ sở trực chuẩn, ta “rời rạc hóa” pha bằng cách chọn N giá trị pha chia đều trên khoảng $[0, 1)$ là

$$\left\{ \frac{j}{N} \right\}_{j=0}^{N-1} = \left\{ 0, \frac{1}{N}, \frac{2}{N}, \dots, \frac{N-1}{N} \right\}$$

đặt

$$|\phi_j\rangle = |\phi_{\frac{j}{N}}\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i \frac{jk}{N}} |k\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{jk} |k\rangle \quad (6.1)$$

trong đó $\omega = \omega_N = e^{2\pi i \frac{1}{N}}$, thì

$$\text{QFT}_N = \{|\phi_j\rangle\}_{j=0}^{N-1} = \{|\phi_0\rangle, |\phi_1\rangle, \dots, |\phi_{N-1}\rangle\}$$

là một cơ sở trực chuẩn (Bài tập 6.1).

Bây giờ, nếu sắp các ket của cơ sở QFT thành các cột của một ma trận thì ta được ma trận biến đổi cơ sở chuẩn tắc (cơ sở tính toán) thành cơ sở QFT, ma trận này cũng được ký hiệu là QFT

$$\text{QFT}_N = \begin{bmatrix} |\phi_0\rangle & |\phi_1\rangle & \dots & |\phi_{N-1}\rangle \end{bmatrix} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \dots & \omega^{(N-1)(N-1)} \end{bmatrix}.$$

¹cách mở rộng đơn giản là dùng $H^{\otimes n}$ trong (5.1). Tuy nhiên, trong cách này, thông tin pha chỉ được mã bằng các khuôn mẫu của 1 và -1 (thuận và nghịch pha), khó phân tích.

Tất cả các phần tử của ma trận này đều là căn bậc N của đơn vị (Phần 1.1.4), trong đó, $\omega = e^{2\pi i \frac{1}{N}}$ là căn nguyên thủy.

Ma trận QFT unita nên là ma trận biểu diễn cho một phép toán lượng tử, mà ta cũng kí hiệu là QFT và gọi là phép **biến đổi Fourier lượng tử** (Quantum Fourier Transform, QFT). QFT thực hiện biến đổi sau trên các vector cơ sở tính toán

$$|j\rangle \xrightarrow{\text{QFT}} |\phi_j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i \frac{jk}{N}} |k\rangle, \quad j \in \{0, 1, \dots, N-1\}.$$

Từ đó, do tính tuyến tính, QFT biến trạng thái

$$|a\rangle = \sum_{j=0}^{N-1} a_j |j\rangle$$

thành trạng thái $|\phi\rangle = \text{QFT}|a\rangle$ với

$$|\phi\rangle = \sum_{j=0}^{N-1} a_j |\phi_j\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} a_j e^{2\pi i \frac{jk}{N}} |k\rangle = \sum_{k=0}^{N-1} \left(\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} a_j e^{2\pi i \frac{jk}{N}} \right) |k\rangle. \quad (6.2)$$

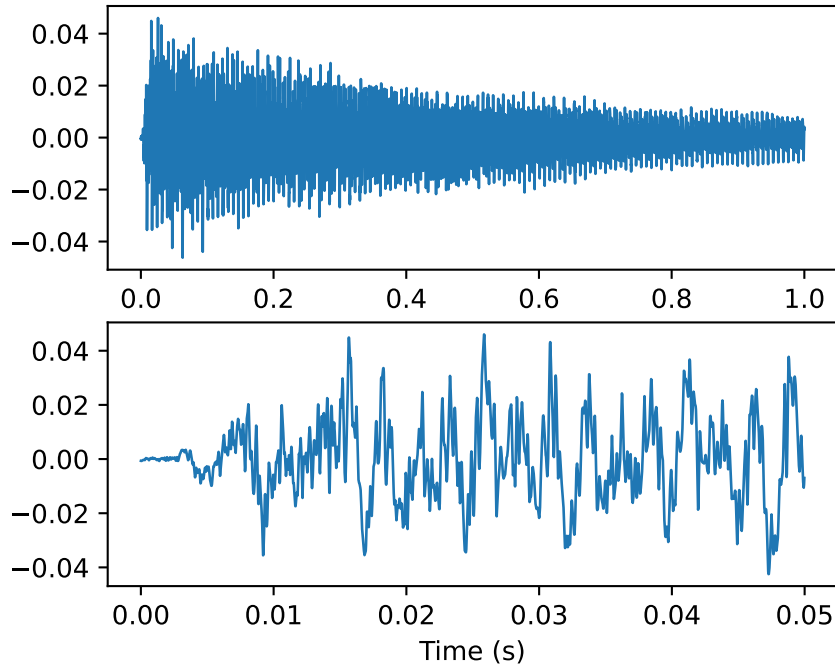
Ta cũng nói, $|\phi\rangle$ là biến đổi Fourier lượng tử của $|a\rangle$.

Ví dụ 6.1.1. Phiên bản biến đổi Fourier trong tính toán cổ điển tương tự (6.2) được gọi là **biến đổi Fourier rời rạc** (Discrete Fourier Transform, DFT). DFT được dùng nhiều trong phân tích tín hiệu và dữ liệu. Ví dụ này minh họa việc dùng DFT để phân tích tần số của một tín hiệu âm thanh.

Cụ thể, tín hiệu cần phân tích là dao động của âm thanh từ việc chơi một hợp âm Đô trưởng (C major triad) trên đàn piano, được lấy từ file âm thanh MP3 ở trang https://en.wikipedia.org/wiki/Major_chord. Hình 6.1 mô tả dạng sóng (waveform), là sự dao động của tín hiệu theo thời gian. Ở hình phía trên, ta thấy sự “tắt dần” của tín hiệu khi âm thanh nhỏ dần. Hơn nữa, trong 1 giây, tín hiệu dao động rất nhiều lần nên ta không thấy rõ dạng sóng. Hình phía dưới, trích ra 0.05 giây đầu tiên của tín hiệu, cho thấy rõ hơn sự dao động.

Do tốc độ lấy mẫu (sampling rate) là 44100 Hz, tức là li độ dao động được đo tại 44100 thời điểm chia đều trong 1 giây, nên tín hiệu có thể được mô tả bằng vector a gồm $N = 44100$ số thực với a_j ($j = 0, 1, \dots, N-1$) là li độ dao động tại thời điểm $\frac{j}{44100}$ giây. Dùng biến đổi Fourier tương tự (6.2) ta được vector phức ϕ cỡ N với

$$\phi_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} a_j e^{2\pi i \frac{jk}{N}}, \quad k = 0, 1, \dots, N-1. \quad (6.3)$$



Hình 6.1: Dạng sóng của tín hiệu âm thanh.

Khi đó, $|\phi_k|$ cho thấy mức độ đóng góp của sóng sin dao động với tần số k Hz trong tín hiệu a .

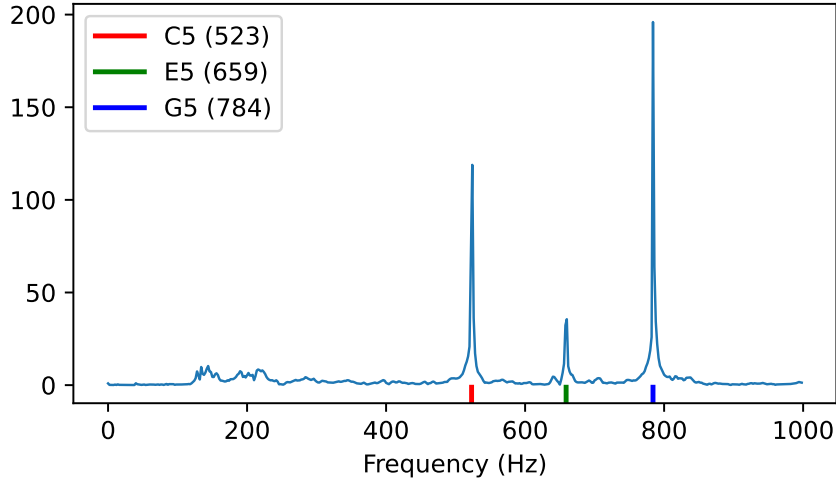
Hình 6.2 cho thấy phổ tần số (frequency spectrum) của tín hiệu a (chỉ vẽ đến tần số 1 kHz = 1000 Hz). Ta thấy tín hiệu a được tạo thành từ các sóng sin có tần số khoảng 523 Hz, 659 Hz, 784 Hz tương ứng là tần số của các nốt nhạc Đô (C), Mi (E), Sol (G) của hợp âm Đô trưởng.²

Từ vector a cỡ N , để tính ϕ_k theo (6.3) ta cần $O(N)$ phép toán. Do đó cần $O(N^2)$ để tính cả vector ϕ do ϕ có N phần tử. Rất may, tính toán cổ điển đã phát minh được một thuật toán, gọi là **biến đổi Fourier nhanh** (Fast Fourier Transform, FFT), tính ϕ từ a chỉ với $O(N \log N)$ phép toán. \square

Trạng thái $|\phi_j\rangle$ ở (6.1) là trạng thái tách được. Thật vậy (Bài tập 6.2)

$$\begin{aligned} |\phi_j\rangle &= \frac{1}{\sqrt{2^n}} (|0\rangle + e^{2\pi i \frac{j}{2}})(|0\rangle + e^{2\pi i \frac{j}{4}}) \dots (|0\rangle + e^{2\pi i \frac{j}{2^n}}) \\ &= \bigotimes_{l=1}^n \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i \frac{j}{2^l}} |1\rangle \right). \end{aligned}$$

²tham khảo tần số các nốt nhạc ở trang https://en.wikipedia.org/wiki/Piano_key_frequencies.



Hình 6.2: Phổ tần số của tín hiệu âm thanh.

Nếu dùng số nhị phân $j = j_{n-1} \dots j_1 j_0$ với lưu ý

$$e^{2\pi i \frac{j}{2^l}} = e^{2\pi i (j_{n-1} \dots j_l \cdot j_{l-1} \dots j_0)} = \underbrace{e^{2\pi i (j_{n-1} \dots j_l)}}_1 e^{2\pi i (0 \cdot j_{l-1} \dots j_0)} = e^{2\pi i (0 \cdot j_{l-1} \dots j_0)}$$

thì

$$|\phi_j\rangle = \frac{1}{\sqrt{2^n}} (|0\rangle + e^{2\pi i (0 \cdot j_0)} |1\rangle) (|0\rangle + e^{2\pi i (0 \cdot j_1 j_0)} |1\rangle) \dots (|0\rangle + e^{2\pi i (0 \cdot j_{n-1} \dots j_1 j_0)} |1\rangle). \quad (6.4)$$

Ví dụ 6.1.2. Trường hợp 1 qubit, $n = 1, N = 2^1 = 2$, ta có

$$\omega = \omega_2 = e^{2\pi i \frac{1}{2}} = -1.$$

Khi đó

$$|\phi_j\rangle = \frac{1}{\sqrt{2}} (|0\rangle + e^{2\pi i (0 \cdot j)} |1\rangle), j \in \{0, 1\}.$$

Cụ thể

$$\begin{aligned} |\phi_0\rangle &= \frac{1}{\sqrt{2}} (|0\rangle + e^{2\pi i 0} |1\rangle) = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) = |+\rangle, \\ |\phi_1\rangle &= \frac{1}{\sqrt{2}} (|0\rangle + e^{2\pi i \frac{1}{2}} |1\rangle) = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) = |-\rangle. \end{aligned}$$

Ma trận QFT khi đó là

$$\text{QFT}_2 = \begin{bmatrix} |\phi_0\rangle & |\phi_1\rangle \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & \omega \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = H.$$

□

Ví dụ 6.1.3. Trường hợp 2 qubit, $n = 2, N = 2^2 = 4$, ta có

$$\omega = \omega_4 = e^{2\pi i \frac{1}{4}} = i.$$

Khi đó

$$|\phi_j\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0,j_0)}|1\rangle) \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0,j_1j_0)}|1\rangle), j = j_1j_0 \in \{0, 1\}^2.$$

Cụ thể

$$\begin{aligned} |\phi_0\rangle &= \frac{1}{2}(|0\rangle + |1\rangle + |2\rangle + |3\rangle) = |\phi_{00}\rangle = \frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle + |1\rangle), \\ |\phi_1\rangle &= \frac{1}{2}(|0\rangle + i|1\rangle - |2\rangle - i|3\rangle) = |\phi_{01}\rangle = \frac{1}{2}(|0\rangle - |1\rangle)(|0\rangle + i|1\rangle), \\ |\phi_2\rangle &= \frac{1}{2}(|0\rangle - |1\rangle + |2\rangle - |3\rangle) = |\phi_{10}\rangle = \frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle), \\ |\phi_3\rangle &= \frac{1}{2}(|0\rangle - i|1\rangle - |2\rangle + i|3\rangle) = |\phi_{11}\rangle = \frac{1}{2}(|0\rangle - |1\rangle)(|0\rangle - i|1\rangle). \end{aligned}$$

Ma trận QFT khi đó là

$$\text{QFT}_4 = \begin{bmatrix} |\phi_0\rangle & |\phi_1\rangle & |\phi_2\rangle & |\phi_3\rangle \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 \\ 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega^9 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}.$$

□

6.1.2 Mạch biến đổi Fourier lượng tử

Làm sao cài đặt phép biến đổi Fourier lượng tử bằng các cổng 1 hay 2 qubit. Một cách tự nhiên, cổng dịch pha $R(\theta)$ (Phần 3.3.2) sẽ được dùng. Để thuận tiện, ta kí hiệu

$$R_r = R\left(\frac{2\pi}{2^r}\right)$$

là cổng 1 qubit “đơn thêm” pha $\frac{2\pi}{2^r}$, tức là

$$R_r(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle + e^{i\frac{2\pi}{2^r}}\beta|1\rangle.$$

Trường hợp $n = 1$ qubit ta thấy QFT_2 chính là H nên cổng H thực hiện phép biến đổi QFT. Trường hợp $n > 1$ qubit, với n qubit đầu vào có trạng thái $|j\rangle =$

$|j_{n-1}\rangle \dots |j_1\rangle |j_0\rangle$ ta muốn tạo trạng thái đầu ra $|\phi_j\rangle = |k_{n-1}\rangle \dots |k_1\rangle |k_0\rangle$. Từ (6.4), ta thấy qubit cao nhất (tận trái) có trạng thái

$$|k_{n-1}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0,j_0)}) = H|j_0\rangle$$

nên chỉ cần áp cổng H lên qubit thấp nhất của đầu vào. Có thể nói, H giúp thêm pha $0.j_0 = \frac{j_0}{2}$ vào $|1\rangle$ của qubit k_{n-1} .

Qubit cao thứ 2 có trạng thái

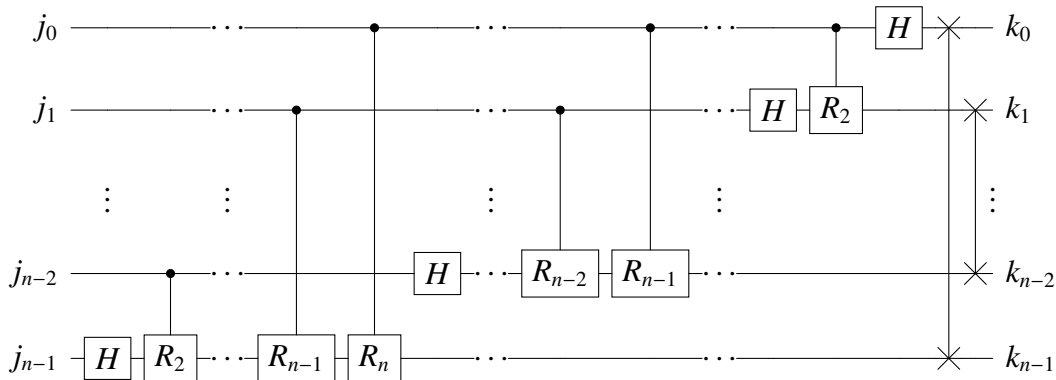
$$|k_{n-2}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0,j_1j_0)}|1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i\frac{j_1}{2}}e^{2\pi i\frac{j_0}{4}}|1\rangle)$$

Dùng H trên qubit j_1 ta có thể thêm pha $\frac{j_1}{2}$. Với pha $\frac{j_0}{4}$, nhận xét

$$\frac{j_0}{4} = \begin{cases} 0 & j_0 = 0 \\ \frac{1}{4} & j_0 = 1 \end{cases}$$

nên dùng cổng điều khiển CR_2 với qubit điều khiển là j_0 để thêm pha $\frac{1}{4}$ vào $|1\rangle$ của qubit k_{n-2} . Lưu ý, cổng H có thể được xem là cổng “tự điều khiển R_1 ”.

Tương tự cho các qubit còn lại, ta có mạch thực hiện QFT như Hình 6.3. Lưu ý, ta đã dùng cổng SWAP để đưa các qubit k_0, k_1, \dots, k_{n-1} về đúng thứ tự (qubit thấp ở trên).



Hình 6.3: Mạch QFT n qubit.

Mạch QFT trong Hình 6.3 dùng n cổng H (mỗi qubit dùng 1 cổng H). Mạch cũng dùng

$$0 + 1 + \dots + (n-2) + (n-1) = \frac{n(n-1)}{2}$$

cổng R , và $\frac{n}{2}$ cổng SWAP. Nếu xem các cổng này là các “phép toán (lượng tử) cơ bản” thì tổng số phép toán dùng trong mạch là

$$n + \frac{n(n-1)}{2} + \frac{n}{2} = \frac{2n + n^2 - n + n}{2} = \frac{2n + n^2}{2}.$$

Như vậy độ phức tạp mạch là $O(n^2) = O(\log^2 N)$. So với độ phức tạp của biến đổi Fourier nhanh FFT trong tính toán cổ điển (Ví dụ 6.1.1) là $O(N \log N)$ thì QFT nhanh hơn theo cấp mũ. Tuy nhiên, lưu ý, khi dùng QFT

- (i) Vector đầu vào $a = (a_0, a_1, \dots, a_{N-1})$ cần được chuẩn bị trong trạng thái lượng tử

$$|a\rangle = \sum_{j=0}^{N-1} a_j |j\rangle,$$

tức là các phần tử của vector cần được mã vào amplitude.

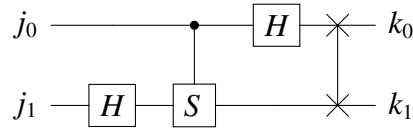
- (ii) Vector đầu ra $\phi = (\phi_0, \phi_1, \dots, \phi_{N-1})$ cũng có các phần tử được mã trong amplitude

$$|\phi\rangle = \text{QFT} |a\rangle = \sum_{k=0}^{N-1} \phi_k |k\rangle,$$

do đó việc “đọc” giá trị các phần tử này là không thể.

Có thể nói QFT không thể được dùng để tăng tốc biến đổi Fourier rời rạc DFT do khó khăn về “vào-ra” dữ liệu cổ điển. Tuy nhiên, nếu được dùng như một “thủ tục con” trong các tính toán lượng tử, nơi mà dữ liệu vào-ra vẫn ở dạng lượng tử, thì QFT có thể giúp tăng tốc (cấp độ mũ) các tính toán này. Điều này được thể hiện rõ nét nhất trong thuật toán phân tích số nguyên của Shor (Phần 6.3).

Ví dụ 6.1.4. Mạch QFT 2 qubit được mô tả như Hình 6.4.



Hình 6.4: Mạch QFT 2 qubit.

Lưu ý, cổng

$$R_2 = R\left(i\frac{\pi}{2}\right)$$

thêm pha $\frac{\pi}{2}$ chính là cổng S .

□

6.1.3 Biến đổi Fourier lượng tử ngược

Vì QFT unita nên nghịch đảo của nó, **biến đổi Fourier lượng tử ngược** (Inverse Quantum Fourier Transform, IQFT), là QFT^\dagger . QFT^\dagger “undo” QFT, tức là biến $|\phi_j\rangle$ ở (6.1) thành $|j\rangle$, cụ thể

$$|\phi_j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i \frac{jk}{N}} |k\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{jk} |k\rangle \xrightarrow{\text{QFT}^\dagger} |j\rangle, j \in \{0, 1, \dots, N-1\}.$$

Với $\omega = e^{2\pi i \frac{1}{N}}$, $\bar{\omega} = e^{2\pi i \frac{-1}{N}}$, ta có

$$\overline{\omega^{jk}} = (\bar{\omega})^{jk} = e^{2\pi i \frac{-jk}{N}}$$

nên QFT^\dagger hoạt động tương tự QFT với $\bar{\omega}$ thay cho ω .

Vì $(U_1 U_2)^\dagger = U_2^\dagger U_1^\dagger$ với U_1, U_2 unita nên chỉ cần dùng các cổng nghịch đảo theo thứ tự ngược của mạch QFT ta có mạch QFT^\dagger . Hơn nữa, nhớ là

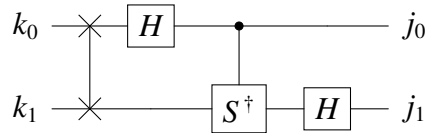
$$H^\dagger = H, \quad \text{SWAP}^\dagger = \text{SWAP}, \quad R_r^\dagger = R\left(\frac{-2\pi}{2^r}\right).$$

Do đó, độ phức tạp mạch của QFT^\dagger cũng giống như QFT là $O(n^2)$.

Ví dụ 6.1.5. Trường hợp 2 qubit, $n = 2, N = 2^2 = 4, \omega = \omega_4 = i, \bar{\omega} = -i$, QFT_4 trong Ví dụ 6.1.3 có biến đổi ngược là

$$\text{QFT}_4^\dagger = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \bar{\omega} & \bar{\omega}^2 & \bar{\omega}^3 \\ 1 & \bar{\omega}^2 & \bar{\omega}^4 & \bar{\omega}^6 \\ 1 & \bar{\omega}^3 & \bar{\omega}^6 & \bar{\omega}^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix}$$

Bằng cách dùng các cổng nghịch đảo theo thứ tự ngược của mạch QFT_4 trong Ví dụ 6.1.4 ta có mạch cho QFT_4^\dagger như sau



□

6.2 Ước lượng pha

Phần này trình bày bài toán ước lượng pha và thuật toán lượng tử hiệu quả để giải bài toán này. Thủ tục ước lượng pha là thủ tục cơ bản, hay được dùng trong nhiều

thủ tục lớn, trong đó có thuật toán Shor. Ta sẽ thấy, thủ tục cốt lõi được dùng trong ước lượng pha là biến đổi Fourier lượng tử ở Phần 6.1. Phần này cần các kiến thức về trị riêng và vector riêng của ma trận trong Phần 1.3.4.

6.2.1 Bài toán ước lượng pha

Ta đã biết các phép toán lượng tử được biểu diễn bởi các ma trận unita U . Nếu U có vector riêng $|v\rangle$ ứng với trị riêng $\lambda \in \mathbb{C}$, tức là

$$U|v\rangle = \lambda|v\rangle$$

thì do U bảo toàn chuẩn nên

$$\| |v\rangle \| = \| U|v\rangle \| = \| \lambda|v\rangle \| = |\lambda| \| |v\rangle \|$$

nên

$$|\lambda| = 1.$$

Như vậy, λ có độ lớn 1 (λ nằm trên đường tròn đơn vị phức) nên λ có thể được mô tả bằng dạng mũ là

$$\lambda = e^{i\theta}, \quad \theta \in [0, 2\pi).$$

Bài toán **ước lượng pha** (phase estimation) là bài toán: cho mạch lượng tử n qubit thực hiện phép toán U cùng với trạng thái $|v\rangle$ là một vector riêng của U , tìm $\theta \in [0, 1)$ sao cho

$$U|v\rangle = e^{2\pi i\theta}|v\rangle.$$

Lưu ý, ta đã nhân thêm 2π để đưa “pha” θ về khoảng $[0, 1)$ thay vì khoảng $[0, 2\pi)$. Hơn nữa trong nhiều trường hợp thực tế, ta chỉ cần xấp xỉ θ .

6.2.2 Ước lượng pha với 1 qubit

Ta thực ra đã dùng các tính toán liên quan đến trị riêng và vector riêng trong hiện tượng “đá pha” (phase kickback) (Phần 5.3.1). Cụ thể, cho oracle U_f của $f: \mathbb{B}^n \rightarrow \mathbb{B}$ ta có, với mọi $x \in \mathbb{B}^n$

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

Như vậy, $|x\rangle|-\rangle$ là các vector riêng của U_f với trị riêng tương ứng là $(-1)^{f(x)}$. Lưu ý, trong trường hợp này, vì $f(x)$ chỉ nhận giá trị 0 hoặc 1 nên pha chỉ có thể là 0 ($(-1)^0 = 1 = e^{2\pi i 0}$) hoặc pha là $\frac{1}{2}$ ($(-1)^1 = -1 = e^{2\pi i \frac{1}{2}}$).

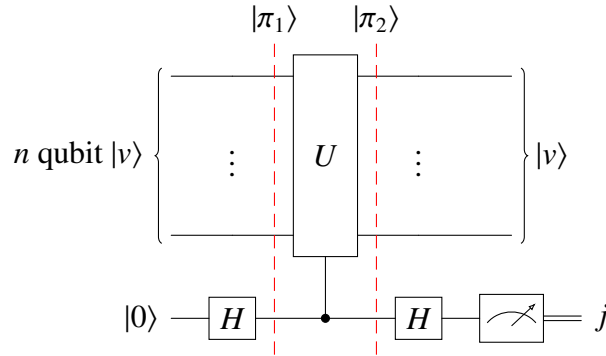
Từ đó, các thuật toán truy vấn ta đã thấy như thuật toán Deutsch, Deutsch-Jozsa, Bernstein-Vazirani, Grover đều dùng hiện tượng này trên

$$|s\rangle = |+\rangle^{\otimes n} = \sum_{x \in \{0,1\}^n} \frac{1}{\sqrt{2^n}} |x\rangle$$

là trạng thái tổ hợp đều để “khắc dấu ấn pha” lên các vector cơ sở

$$U_f |s\rangle |-\rangle = \sum_{x \in \{0,1\}^n} \frac{1}{\sqrt{2^n}} (-1)^{f(x)} |x\rangle |-\rangle.$$

Một cách tương tự, với bài toán ước lượng pha, xét mạch cho trong Hình 6.5. Mạch



Hình 6.5: Mạch ước lượng pha 1 qubit

này có dùng cổng điều khiển U thường được kí hiệu là CU (xem Phần 4.3.2). Hoạt động của CU có thể được tóm tắt là “tác động U lên các qubit mục tiêu (n qubit trên) khi qubit điều khiển (qubit dưới cùng) có giá trị 1”, cụ thể

$$CU(|0\rangle |x\rangle) = |0\rangle |x\rangle, CU(|1\rangle |x\rangle) = |1\rangle U|x\rangle, \forall x \in \mathbb{B}^n. \quad (6.5)$$

Ta phân tích hoạt động của mạch trên. Trước hết, vì $H|0\rangle = |+\rangle$ nên trạng thái chuẩn bị là

$$|\pi_1\rangle = |+\rangle |v\rangle = \frac{1}{\sqrt{2}}(|0\rangle |v\rangle + |1\rangle |v\rangle).$$

Sau đó qua cổng điều khiển U trạng thái là

$$\begin{aligned} |\pi_2\rangle &= CU(|+\rangle |v\rangle) = \frac{1}{\sqrt{2}} CU(|0\rangle |v\rangle + |1\rangle |v\rangle) = \frac{1}{\sqrt{2}}(|0\rangle |v\rangle + |1\rangle U|v\rangle) \\ &= \frac{1}{\sqrt{2}}(|0\rangle |v\rangle + e^{2\pi i \theta} |1\rangle |v\rangle) = \underbrace{\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i \theta} |1\rangle)}_{|o\rangle} |v\rangle. \end{aligned}$$

Tương tự hiện tượng đá pha trên oracle pha, ta thấy trạng thái của n bit trên (có thể gọi là “thanh ghi vector riêng”) không thay đổi (vẫn là $|v\rangle$). Trạng thái của qubit dưới cùng là

$$|o\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i \theta} |1\rangle).$$

Ta thấy pha θ đã được “khắc” thành pha tương đối của $|0\rangle, |1\rangle$. Nhiệm vụ của mạch sau đó (cổng H và phép đo) là “đọc được” thông tin này để có θ .

Nhận xét

- $\theta = 0$ thì $|o\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle$,
- $\theta = \frac{1}{2}$ thì $|o\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$.

Nếu θ chỉ có thể là 0 hoặc $\frac{1}{2}$ thì $|o\rangle$ chỉ có thể là $|+\rangle$ hoặc $|-\rangle$. Do đó cổng H sẽ biến $|o\rangle$ thành $|0\rangle$ hoặc $|1\rangle$ và phép đo (trong cơ sở tính toán) sẽ được bit j tương ứng là 0 hoặc 1. Trường hợp θ có thể là các giá trị khác trong khoảng $[0, 1)$ thì

$$\begin{aligned} H|o\rangle &= H \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i\theta}|1\rangle) = \frac{1}{\sqrt{2}}(|+\rangle + e^{2\pi i\theta}|-\rangle) \\ &= \frac{1}{2}(|0\rangle + |1\rangle + e^{2\pi i\theta}(|0\rangle - |1\rangle)) = \frac{1 + e^{2\pi i\theta}}{2}|0\rangle + \frac{1 - e^{2\pi i\theta}}{2}|1\rangle \end{aligned}$$

nên xác suất mạch trên cho ra $j = 1$ là

$$P_1(\theta) = \left| \frac{1 - e^{2\pi i\theta}}{2} \right|^2$$

và xác suất mạch cho ra $j = 0$ là

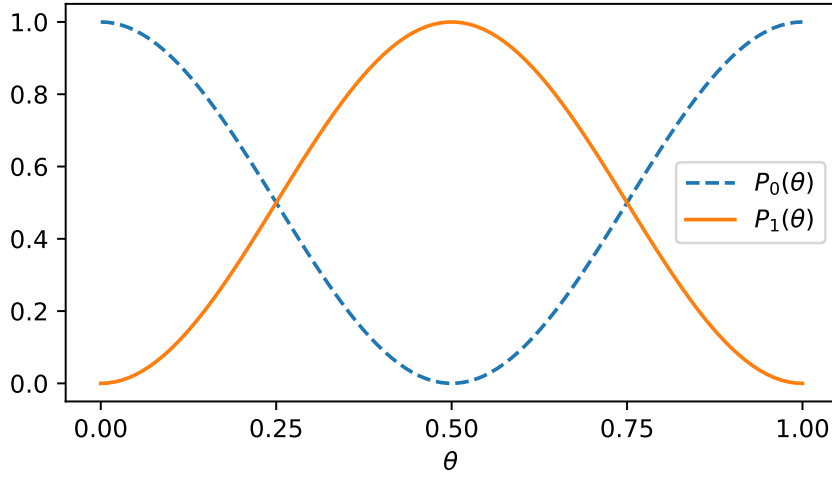
$$P_0(\theta) = \left| \frac{1 + e^{2\pi i\theta}}{2} \right|^2 = 1 - P_1(\theta).$$

Hình 6.6 minh họa các xác suất này theo pha $\theta \in [0, 1)$. Khi $\theta = 0.5$ mạch trên chắc chắn cho ra đúng bit $j = 1$ vì $P_1(0.5) = 1$. Nếu $\theta \approx 0.5$ thì mạch cho ra kết quả là 1 với xác suất rất cao. Khi $\theta \in (0.25, 0.75)$ thì mạch cho ra kết quả 1 với xác suất cao hơn kết quả 0, $P_1(\theta) > P_0(\theta)$. Ta có thể nói $(0.25, 0.75)$ là vùng “ủng hộ” 1. Ngược lại, $[0, 0.25)$ và $(0.75, 1]$ là vùng “ủng hộ” 0 vì $P_0(\theta) > P_1(\theta)$, nhất là khi $\theta \approx 0$ hoặc $\theta \approx 1$ thì mạch cho ra kết quả là 0 với xác suất rất cao. Lưu ý, vì pha xác định “điểm trên đường tròn” nên 0 hay 1 là như nhau (0.01 gần 0.99 trên đường tròn). Hình 6.9(a) minh họa điều này.

Nếu muốn kết quả trả ra của mạch trên là 0 hoặc $\frac{1}{2}$, ta có thể “hậu xử lý” bằng cách chia j cho 2

$$\frac{j}{2} = \begin{cases} 0 & j = 0 \\ 0.5 & j = 1 \end{cases}$$

để được thủ tục trả về 0 hoặc 0.5.



Hình 6.6: Các xác suất được 1 và 0 theo θ của mạch 6.5.

Như vậy, có thể nói, thủ tục trên “cố gắng làm tròn” pha θ với sai số 0.5. Khi pha đúng bằng 0 hoặc 0.5, kết quả trả ra chắc chắn chính xác. Trường hợp khác, thủ tục trả ra một trong 2 giá trị làm tròn có thể của θ là 0 hoặc 0.5 với xác suất điều chỉnh theo “độ gần” của θ với các giá trị này.

6.2.3 Ước lượng pha với 2 qubit

Để ước lượng pha với sai số nhỏ hơn, trước hết, ta nhận xét, nếu $|v\rangle$ là vector riêng của U với trị riêng có pha θ , tức là

$$U|v\rangle = e^{2\pi i\theta}|v\rangle$$

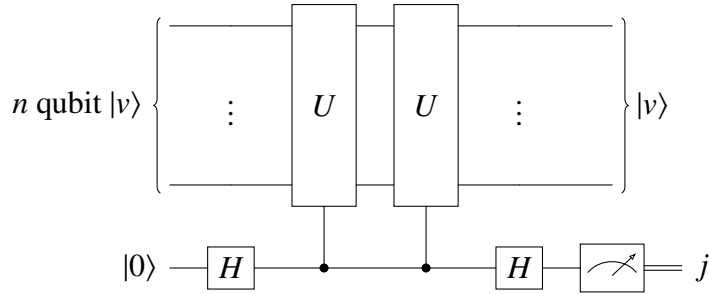
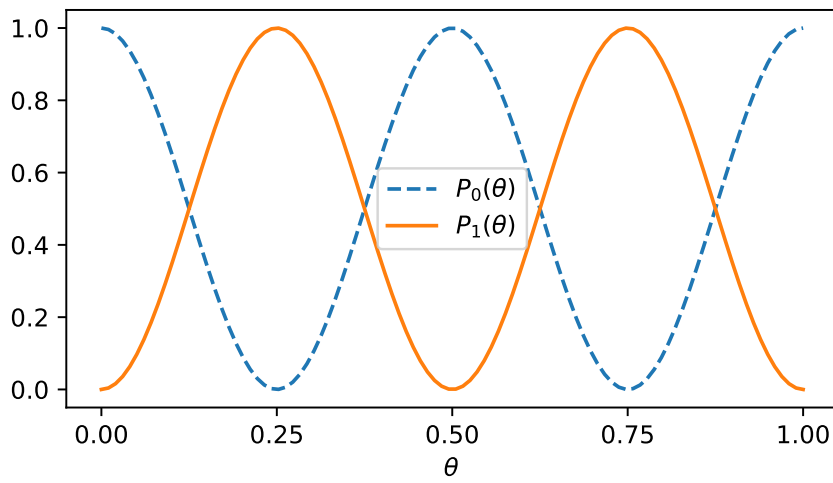
thì $|v\rangle$ cũng là vector riêng của $U^2 = UU$ với trị riêng có pha 2θ vì

$$U^2|v\rangle = U(U|v\rangle) = Ue^{2\pi i\theta}|v\rangle = e^{2\pi i\theta}U|v\rangle = e^{2\pi i\theta}e^{2\pi i\theta}|v\rangle = e^{2\pi i(2\theta)}|v\rangle.$$

Như vậy, nếu ta dùng 2 cổng điều khiển U như mạch trong Hình 6.7 (thay vì 1 như mạch 6.5) thì ta sẽ ước lượng 2θ thay vì θ .

Cách hoạt động của mạch 6.7 tương tự như mạch 6.5 với 2θ thay cho θ . Hình 6.8 minh họa các xác suất đo được j là 1 hay 0 theo θ . So với Hình 6.6, ta thấy xác suất “ủng hộ” 1 thay đổi nhanh gấp đôi. Khi $\theta = 0.25$ hoặc $\theta = 0.75$ mạch trên chắc chắn cho ra đúng bit $j = 1$, ngược lại, khi $\theta = 0$ hoặc $\theta = 0.5$ mạch trên chắc chắn cho ra đúng bit $j = 0$. Các vùng “ưu tiên” 1 hay 0 được minh họa trong Hình 6.9(b).

Nếu xem biểu diễn nhị phân của pha θ là $0.\theta_1\theta_2\theta_3\dots$ thì mạch 6.5 ước lượng bit θ_1 còn mạch 6.7 ước lượng bit θ_2 . Tuy nhiên, dưới góc nhìn cổ điển, việc dùng đồng

Hình 6.7: Mạch ước lượng pha 1 qubit dùng 2 cổng điều khiển U Hình 6.8: Các xác suất được 1 và 0 theo θ của mạch 6.7.

thời 2 mạch không giúp ta ước lượng được pha đến 2 chữ số (tức là ước lượng $0.\theta_1\theta_2$). Trong Hình 6.9(c), ta thấy “lẫn lộn” 01 và 11 trong vùng 0.25 cũng như 0.75. Giả như vùng 0.25 là 01 còn vùng 0.75 là 11 thì tốt?!

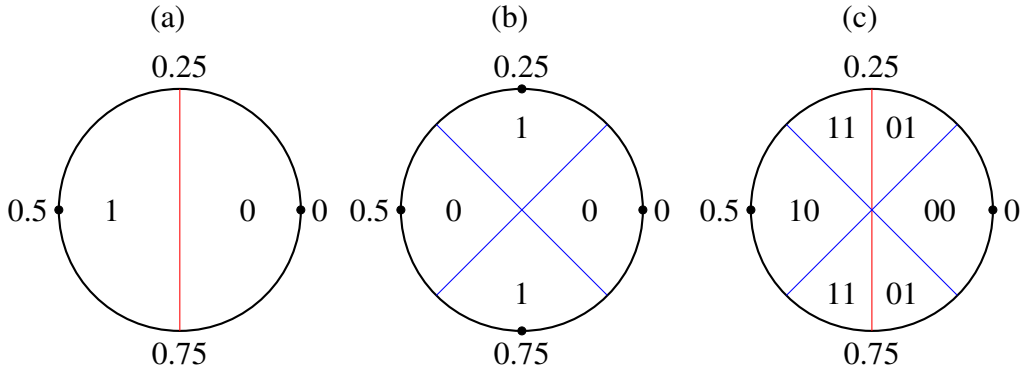
Tuy nhiên, với tính toán lượng tử ta có thể ước lượng pha đến 2 chữ số bằng cách kết hợp 2 mạch trên để có mạch ước lượng pha 2 qubit như trong Hình 6.10.

Trước hết, trạng thái chuẩn bị là

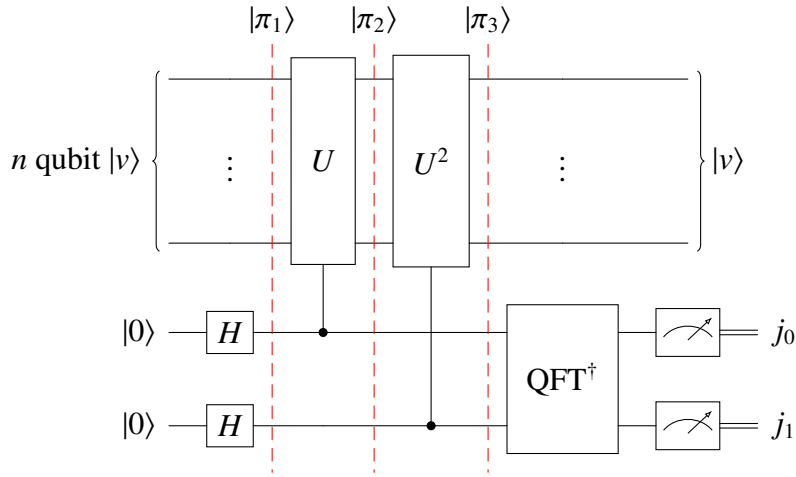
$$|\pi_1\rangle = |+\rangle^{\otimes 2} |v\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) |v\rangle.$$

Sau đó, cổng điều khiển U sẽ “đá” pha θ vào các amplitude nơi qubit dưới thứ nhất là 1, được trạng thái

$$|\pi_2\rangle = \frac{1}{2}(|00\rangle + e^{2\pi i\theta} |01\rangle + |10\rangle + e^{2\pi i\theta} |11\rangle) |v\rangle.$$



Hình 6.9: Kết hợp cổng điển thông tin ước lượng 2 bit.



Hình 6.10: Mạch ước lượng pha 2 qubit.

Tiếp theo, cổng điều khiển U^2 (chính là 2 cổng điều khiển U ở mạch 6.7) sẽ “đá” pha 2θ vào các amplitude nơi qubit dưới thứ hai (dưới cùng) là 1, được trạng thái

$$\begin{aligned}
 |\pi_3\rangle &= \frac{1}{2}(|00\rangle + e^{2\pi i\theta} |01\rangle + e^{2\pi i(2\theta)} |10\rangle + e^{2\pi i\theta} e^{2\pi i(2\theta)} |11\rangle) |v\rangle \\
 &= \frac{1}{2} \underbrace{(e^{2\pi i(0\theta)} |00\rangle + e^{2\pi i(1\theta)} |01\rangle + e^{2\pi i(2\theta)} |10\rangle + e^{2\pi i(3\theta)} |11\rangle)}_{|o\rangle} |v\rangle
 \end{aligned}$$

Nếu xem chuỗi 2 bit là số nhị phân ta có thể viết gọn $|o\rangle$ là

$$|o\rangle = \frac{1}{2} \sum_{k=0}^3 e^{2\pi i k \theta} |k\rangle.$$

Ta thấy pha θ đã được “khắc” vào pha tương đối của các vector cơ sở $|k\rangle$ bằng bội

số tương ứng. Nếu θ chỉ có thể là các điểm chia phần tư là 0, 0.25, 0.5, 0.75 tức là

$$\theta = \frac{j}{4}, j \in \{0, 1, 2, 3\}$$

thì $|o\rangle$ tương ứng là

$$|o\rangle = \frac{1}{2} \sum_{k=0}^3 e^{2\pi i \frac{jk}{4}} |k\rangle$$

cụ thể

- $\theta = \frac{0}{4}$ thì $|o\rangle = \frac{1}{2}(|0\rangle + |1\rangle + |2\rangle + |3\rangle) = |\phi_0\rangle$,
- $\theta = \frac{1}{4}$ thì $|o\rangle = \frac{1}{2}(|0\rangle + i|1\rangle - |2\rangle - i|3\rangle) = |\phi_1\rangle$,
- $\theta = \frac{2}{4}$ thì $|o\rangle = \frac{1}{2}(|0\rangle - |1\rangle + |2\rangle - |3\rangle) = |\phi_2\rangle$,
- $\theta = \frac{3}{4}$ thì $|o\rangle = \frac{1}{2}(|0\rangle - i|1\rangle - |2\rangle + i|3\rangle) = |\phi_3\rangle$.

Ta thấy $\{|\phi_j\rangle\}_{j=0}^3$ chính là cơ sở QFT₄ (Ví dụ 6.1.3). Do đó nếu dùng phép biến đổi Fourier ngược tức là QFT[†] thì ta có thể đưa $|\phi_j\rangle$ thành $|j\rangle$ mà khi đó, đo trong cơ sở tính toán ta sẽ được j mà nếu hậu xử lý bằng cách chia j cho 4 thì ta tính được chính xác pha θ .

6.2.4 Thủ tục ước lượng pha

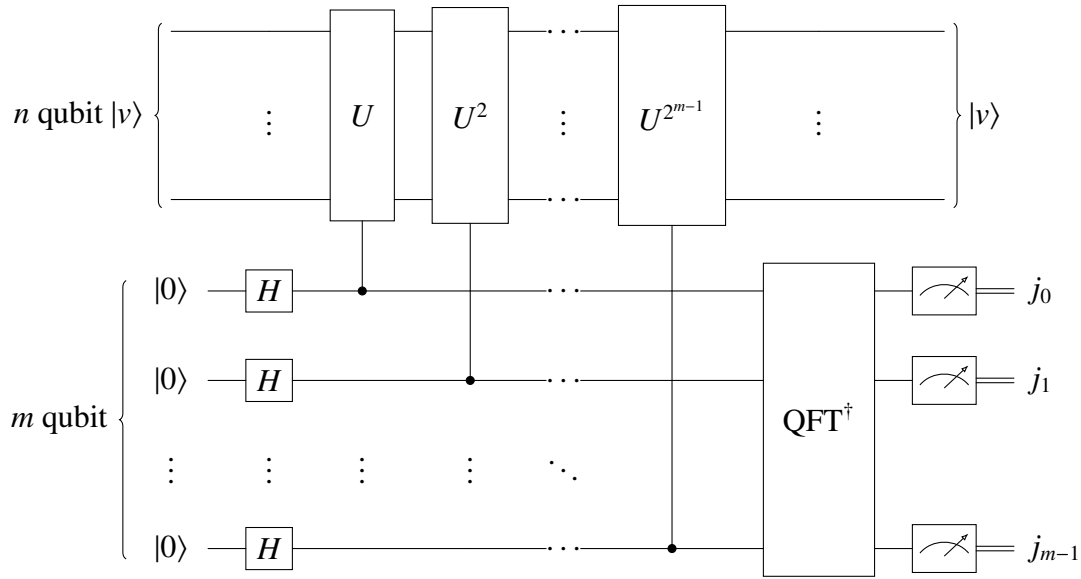
Tổng quát từ mạch ước lượng pha 2 qubit (Hình 6.10), ta dễ dàng có được mạch ước lượng pha dùng m qubit như Hình 6.11. Kết quả đo $j = j_{m-1} \dots j_1 j_0$ được hậu xử lý để trả về kết quả của thủ tục ước lượng pha là

$$\frac{j}{2^m} = 0.j_{m-1} \dots j_1 j_0.$$

Xét “thanh ghi trị riêng” gồm m qubit dưới (“thanh ghi vector riêng” gồm n qubit trên có trạng thái $|v\rangle$ không đổi), các bước của thuật toán ước lượng pha là

1. Các cổng H giúp tạo trạng thái $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ cho mỗi qubit.
2. Các cổng điều khiển U^{2^k} ($k = 0, 1, \dots, m-1$) khắc bội pha $2^k\theta$ vào $|1\rangle$ của các qubit tương ứng được

$$|\phi_\theta\rangle = \frac{1}{\sqrt{2^m}}(|0\rangle + e^{2\pi i 2^{m-1}\theta} |1\rangle) \dots (|0\rangle + e^{2\pi i 2\theta} |1\rangle)(|0\rangle + e^{2\pi i \theta} |1\rangle)$$

Hình 6.11: Mạch ước lượng pha m qubit.

3. Bây giờ, nếu $\theta = \frac{j}{2^m}$ với $j = j_{m-1} \dots j_1 j_0$ thì

$$|\phi_\theta\rangle = |\phi_{\frac{j}{2^m}}\rangle = (|0\rangle + e^{2\pi i(0.j_0)} |1\rangle) \dots (|0\rangle + e^{2\pi i(0.j_{m-2} \dots j_0)} |1\rangle) (|0\rangle + e^{2\pi i(0.j_{m-1} j_{m-2} \dots j_0)} |1\rangle).$$

Đối chiếu với 6.4 ta thấy đây chính là vector thứ j trong cơ sở QFT. Do đó dùng biến đổi Fourier ngược QFT^\dagger ta có j . Hậu xử lý $\frac{j}{2^m}$ ta có θ .

Nếu pha θ không biểu diễn được bằng m bit ($\theta \neq \frac{j}{2^m}, \forall j \in \{0, 1, \dots, 2^m - 1\}$) thì mạch sẽ cho ra kết quả “làm tròn m bit” của θ với xác suất cao. Bây giờ, bằng cách chạy thuật toán nhiều lần và chọn kết quả đồng thuận ta có được giá trị làm tròn của θ đến m bit nhị phân. Dĩ nhiên, nếu muốn ước lượng θ chính xác hơn, ta có thể tăng số qubit m của “thanh ghi trị riêng”.

Mạch ước lượng pha m qubit cần m cổng H và $O(m^2)$ cổng cơ bản của mạch QFT^\dagger . Quan trọng hơn, ta cần chuẩn bị được trạng thái riêng $|v\rangle$ cho mạch U và cài đặt được hiệu quả dây cổng điều khiển U^{2^k} ($k = 0, 1, \dots, m-1$). Như vậy, thủ tục ước lượng pha ở Hình 6.11 là một “khung thuật toán” (framework) mà các thủ tục khác có thể dùng để giải các bài toán cụ thể bằng cách chuẩn bị $|v\rangle$ và cài đặt dây cổng điều khiển U^{2^k} hiệu quả. Ta sẽ thấy một thuật toán rất quan trọng như vậy, thuật toán phân tích số nguyên của Shor ở Phần 6.3.

Để hiểu rõ hơn tác động của các cổng điều khiển U , ta thấy nếu $|z\rangle, z \in \{0, 1\}$ là qubit điều khiển U tác động lên $|x\rangle$ thì (6.5) có thể được viết gọn là

$$CU(|z\rangle|x\rangle) = |z\rangle U^z|x\rangle,$$

với $U^0 = I$ và $U^1 = U$. Từ đó ta có tác động của dãy cổng điều khiển U^{2^k} ($k = 0, 1, \dots, m-1$) trong mạch ước lượng pha là biến $|z\rangle |x\rangle$, với $z = z_{m-1} \dots z_1 z_0 \in \{0, 1\}^m$ thành

$$\begin{aligned} |z_{m-1}\rangle \dots |z_1\rangle |z_0\rangle U^{2^{m-1}z_{m-1}} \dots U^{2z_1} U^{z_0} |x\rangle &= |z_{m-1} \dots z_1 z_0\rangle U^{2^{m-1}z_{m-1} + \dots + 2z_1 + z_0} |x\rangle \\ &= |z\rangle U^z |x\rangle. \end{aligned} \quad (6.6)$$

Trong vài trường hợp, việc chuẩn bị vector riêng $|v\rangle$ cho U là việc khó. Khi đó nếu ta dùng

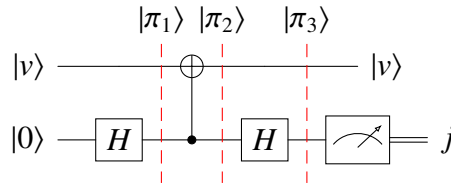
$$|\psi\rangle = \sum_{s=0}^l \alpha_s |v_s\rangle$$

là tổ hợp tuyến tính của các vector riêng $|v_s\rangle$ của U làm trạng thái của thanh ghi vector riêng thì mạch 6.11 sẽ “chọn” ước lượng pha tương ứng cho vector riêng $|v_s\rangle$ với xác suất là $|\alpha_s|^2$ (Bài tập 6.8).

Ví dụ 6.2.1. Ta đã biết cổng X có 2 cặp vector riêng và trị riêng tương ứng là $|v_1\rangle = |+\rangle, \lambda_1 = 1$ và $|v_2\rangle = |-\rangle, \lambda_2 = -1$ vì

$$X|+\rangle = |+\rangle, \quad X|-\rangle = -|-\rangle.$$

Hơn nữa, $\lambda_1 = 1 = e^{2\pi i 0}$ có pha tương ứng là $\theta_1 = 0$ và $\lambda_2 = -1 = e^{2\pi i \frac{1}{2}}$ có pha tương ứng là $\theta_2 = 0.5$ nên có thể được ước lượng chính xác bằng mạch ước lượng pha 1 qubit như sau



Ở đây, U là X nên cổng điều khiển CU chính là CX tức là CNOT. Hơn nữa, QFT_2^\dagger chính là H .

Với $|v\rangle = |v_1\rangle = |+\rangle$ ta có

$$\begin{aligned} |\pi_1\rangle &= |+\rangle |+\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle), \\ |\pi_2\rangle &= \frac{1}{2}(|00\rangle + |01\rangle + |11\rangle + |10\rangle) = |+\rangle |+\rangle, \\ |\pi_3\rangle &= |0\rangle |+\rangle. \end{aligned}$$

Khi đo, chắc chắn được $j = 0$ nên được pha $\theta = \frac{j}{2} = 0 = \theta_1$. Tương tự, với $|v\rangle = |v_2\rangle = |-\rangle$ ta có

$$\begin{aligned} |\pi_1\rangle &= |+\rangle|-\rangle = \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle), \\ |\pi_2\rangle &= \frac{1}{2}(|00\rangle - |01\rangle + |11\rangle - |10\rangle) = |-\rangle|-\rangle, \\ |\pi_3\rangle &= |1\rangle|-\rangle. \end{aligned}$$

Khi đo, chắc chắn được $j = 1$ nên được pha $\theta = \frac{j}{2} = 0.5 = \theta_2$. □

6.3 Thuật toán Shor *

Phần này trình bày thuật toán Shor, một thuật toán lượng tử hiệu quả để phân tích số nguyên (integer factorization). Đây có thể nói là hòn ngọc của tính toán lượng tử. Về lý thuyết, thuật toán Shor cho thấy ưu thế cấp mũ của tính toán lượng tử so với tính toán cổ điển. Về thực tế, thuật toán Shor cho thấy tiềm năng ứng dụng rất lớn của tính toán lượng tử. Có thể nói, thuật toán Shor (và các thuật toán tương tự) là động lực phát triển của điện toán lượng tử.

Trọng tâm của thuật toán Shor là thủ tục tìm chu kỳ của phép toán lũy thừa modulo. Như ta sẽ thấy, thuật toán này có thể được giải một cách khéo léo bằng khung thuật toán ước lượng pha ở Phần 6.2.

6.3.1 Phép toán lũy thừa modulo

Tập số nguyên

$$\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$$

là tập vô hạn. Các phép toán trên \mathbb{Z} có thể cho kết quả lớn tùy ý. Để giới hạn phạm vi các số nguyên, phép toán modulo hay được dùng. Cụ thể, cho số nguyên dương N , khi số nguyên a chia cho N dư r ta nói

$$a = r \pmod{N}.$$

Chẳng hạn

$$2^3 = 8 \pmod{5}$$

vì $2^3 = 8$ chia cho 5 dư 3.

Lưu ý, kí hiệu modulo (\pmod{N}) có thể không được ghi rõ. Hơn nữa, khi lấy kết quả modulo N , tức là lấy phần dư khi chia cho N , ta được số nguyên không âm nhỏ hơn N , nên để thuận tiện, ta kí hiệu

$$\mathbb{Z}_N = \{0, 1, \dots, N-1\}.$$

Trên \mathbb{Z}_N , một phép toán quan trọng hay được dùng là phép **lũy thừa modulo** (modular exponentiation). Cụ thể, cho $a \in \mathbb{Z}_N$ và x là số nguyên không âm, ta cần tính

$$a^x \pmod{N}.$$

Với a, N, x nhỏ ta có thể tính “như bình thường”, chẳng hạn $2^3 = 2 \times 2 \times 2 = 8$ chia cho 5 dư 3 nên $2^3 = 3 \pmod{5}$. Tuy nhiên, với a, N, x lớn ta cần cách tính nhanh hơn, chẳng hạn $91^{43} \pmod{131}$ là bao nhiêu?

Rất may, tính toán cổ điển có thuật toán tính rất nhanh phép lũy thừa modulo. Nếu x được biểu diễn bằng số nhị phân n bit là $x = x_{n-1} \dots x_1 x_0$, tức là

$$x = \sum_{i=0}^{n-1} x_i 2^i$$

thì Thuật toán 13 giúp tính nhanh $a^x \pmod{N}$.

Thuật toán 13 Thuật toán tính nhanh lũy thừa modulo.

Input: các số nguyên dương a, N, x với $x = x_{n-1} \dots x_1 x_0$

Output: $a^x \pmod{N}$

```

1:  $y = 1$ 
2: for  $i = n - 1$  downto 0 do
3:    $y = y \cdot y \pmod{N}$ 
4:   if  $x_i = 1$  then
5:      $y = y \cdot a \pmod{N}$ 
6:   end if
7: end for
8: return  $y$ 
```

Nếu biểu diễn nhị phân của x gồm n bit thì Thuật toán 13 lặp n lần. Mỗi lần lặp, thuật toán thực hiện 1 phép nhân modulo ở dòng 3 và thêm 1 phép nhân modulo ở dòng 5 nếu bit đang xét là 1. Như vậy, số phép nhân modulo của thuật toán là $O(n)$. Hơn nữa, nếu các số a, N cũng có biểu diễn nhị phân khoảng n bit thì phép nhân modulo có thể được thực hiện với $O(n^2)$ phép toán trên bit. Khi đó, tổng số phép toán trên bit của thuật toán là $O(n^3)$.

Thuật toán 13 rất hiệu quả (tính rất nhanh) vì $n \approx \log x$ do đó $x \approx 2^n$ nên nếu tính “như bình thường” (nhân x lần để tính a^x) ta cần $O(2^n)$ phép toán nhân modulo.

Ví dụ 6.3.1. Để tính $91^{43} \pmod{131}$, tức $a = 91, x = 43, N = 131$, ta có $x = 43$ được biểu diễn bằng số nhị phân $n = 6$ bit là

$$\underbrace{1}_{x_5} \underbrace{0}_{x_4} \underbrace{1}_{x_3} \underbrace{0}_{x_2} \underbrace{1}_{x_1} \underbrace{1}_{x_0}$$

Chạy Thuật toán 13 ta có y lần lượt là (kết quả modulo $N = 131$)

1. $y = a^0 = 1$
2. $i = 5$: $y = y \cdot y = a^0 = 1$, do $x_5 = 1$ nên $y = y \cdot a = a^1 = 91$,
3. $i = 4$: $y = y \cdot y = a^2 = 28$,
4. $i = 3$: $y = y \cdot y = a^4 = 129$, do $x_3 = 1$ nên $y = y \cdot a = a^5 = 80$,
5. $i = 2$: $y = y \cdot y = a^{10} = 112$,
6. $i = 1$: $y = y \cdot y = a^{20} = 99$, do $x_1 = 1$ nên $y = y \cdot a = a^{21} = 101$,
7. $i = 0$: $y = y \cdot y = a^{42} = 114$, do $x_0 = 1$ nên $y = y \cdot a = a^{43} = 25$.

Vậy $91^{43} = 25 \pmod{131}$. □

6.3.2 Chu kỳ lũy thừa modulo

Với phép toán lũy thừa modulo, một bài toán rất quan trọng được đặt ra là: cho các số nguyên dương $a < N$ nguyên tố cùng nhau, $\gcd(a, N) = 1$, tìm số nguyên dương r nhỏ nhất thỏa

$$a^r = 1 \pmod{N}.$$

Số nguyên r thường được gọi là **cấp** (order) của a modulo N . Ta cũng thấy r là số nguyên dương nhỏ nhất thỏa

$$a^{x+r} = a^x \pmod{N}, \forall x \in \mathbb{Z}$$

nên r cũng được gọi là **chu kỳ** (period) của phép lũy thừa cơ số a modulo N . Do đó bài toán trên thường được gọi là bài toán tìm cấp hay tìm chu kỳ lũy thừa modulo. Lưu ý, $0 < r < N$ (Bài tập 6.11).

Mặc dù tính được lũy thừa modulo rất nhanh nhưng tính toán cổ điển vẫn chưa có cách nào tìm được chu kỳ nhanh. Một cách để tìm chu kỳ r là ta thử tính $a^x \pmod{N}$ với $x = 1, 2, \dots, N-1$ cho đến khi được $a^x = 1 \pmod{N}$. Số lần thử như vậy là $O(N)$.

Ví dụ 6.3.2. Với $N = 5, a = 2$ ta có (kết quả modulo 5)

$$2^0 = 1, \quad 2^1 = 2, \quad 2^2 = 4, \quad 2^3 = 3, \quad 2^4 = 1, \quad 2^5 = 2, \quad \dots$$

nên cấp của 2 modulo 5 là $r = 4$. Ta cũng thấy, $2^x \pmod{5}$ lần lượt có giá trị là

$$\underbrace{1, 2, 4, 3}_{\text{chu kỳ 1}}, \underbrace{1, 2, 4, 3}_{\text{chu kỳ 2}}, \dots$$

Các giá trị 1, 2, 4, 3 được lặp lại tuần hoàn với chu kỳ 4 (gồm 4 số).

Với $N = 7, a = 2$ ta có (kết quả modulo 7)

$$2^0 = 1, \quad 2^1 = 2, \quad 2^2 = 4, \quad 2^3 = 1, \quad 2^4 = 2, \quad \dots$$

nên cấp của 2 modulo 7 là 3. Ta cũng thấy, $2^x \bmod 7$ có các giá trị 1, 2, 4 được lặp lại tuần hoàn với chu kỳ 3. \square

6.3.3 Mạch lượng tử nhân modulo

Tính toán lượng tử có thể tìm chu kỳ của $a^x \pmod{N}$ một cách hiệu quả bằng cách vận dụng khéo léo khung thuật toán ước lượng pha ở Phần 6.2. Trước hết, để làm việc với $Z_N = \{0, 1, \dots, N-1\}$, ta cần các số nhị phân có kích thước là $n = \lceil \log N \rceil$. Sau đó, xét phép toán lượng tử M_a thao tác trên n qubit được định nghĩa bởi tác động trên các vector cơ sở tính toán $|x\rangle, x \in \{0, 1, \dots, 2^n - 1\}$ là

$$M_a |x\rangle = \begin{cases} |ax \bmod N\rangle & 0 \leq x < N, \\ |x\rangle & N \leq x < 2^n. \end{cases} \quad (6.7)$$

Vì a, N nguyên tố cùng nhau nên M_a unita, do đó M_a là phép toán lượng tử hợp lệ (Bài tập 6.12).

Gọi r là chu kỳ của $a^x \bmod N$, xét trạng thái (các kết quả modulo N)

$$|v_0\rangle = \frac{1}{\sqrt{r}}(|1\rangle + |a\rangle + \dots + |a^{r-1}\rangle).$$

Vì $M_a |a^x\rangle = |aa^x\rangle = |a^{x+1}\rangle$ và $a^r = 1$ nên ta có

$$M_a |v_0\rangle = \frac{1}{\sqrt{r}}(|a\rangle + |a^2\rangle + \dots + |a^r\rangle) = \frac{1}{\sqrt{r}}(|1\rangle + |a\rangle + |a^2\rangle + \dots + |a^{r-1}\rangle) = |v_0\rangle.$$

Như vậy, $|v_0\rangle$ là một vector riêng của M_a với trị riêng 1. Tương tự, xét trạng thái

$$|v_1\rangle = \frac{1}{\sqrt{r}}(|1\rangle + \omega_r^{-1} |a\rangle + \dots + \omega_r^{-(r-1)} |a^{r-1}\rangle)$$

với $\omega_r = e^{2\pi i \frac{1}{r}}$, ta có

$$\begin{aligned} M_a |v_1\rangle &= \frac{1}{\sqrt{r}}(|a\rangle + \omega_r^{-1} |a^2\rangle + \dots + \omega_r^{-(r-1)} |a^r\rangle) \\ &= \frac{\omega_r}{\sqrt{r}}(\omega_r^{-1} |a\rangle + \omega_r^{-1} \omega_r^{-1} |a^2\rangle + \dots + \omega_r^{-1} \omega_r^{-(r-1)} |1\rangle) \\ &= \frac{\omega_r}{\sqrt{r}}(|1\rangle + \omega_r^{-1} |a\rangle + \dots + \omega_r^{-(r-1)} |a^{r-1}\rangle) = \omega_r |v_1\rangle. \end{aligned}$$

Lưu ý, hàm ω_r^x cũng tuần hoàn với chu kỳ r (xem lại Phần 1.1.4) nên

$$\omega_r^{-1} \omega_r^{-(r-1)} = \omega_r^{-r} = 1.$$

Như vậy, $|v_1\rangle$ là một vector riêng của M_a với trị riêng $\omega_r = e^{2\pi i \frac{1}{r}}$. Một cách tổng quát, với $s = 0, 1, \dots, r-1$, ta có (Bài tập 6.13)

$$|v_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \omega_r^{-sk} |a^k\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i \frac{sk}{r}} |a^k\rangle$$

là các vector riêng của M_a với trị riêng tương ứng là $\omega_r^s = e^{2\pi i \frac{s}{r}}$. Hơn nữa

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |v_s\rangle = |1\rangle.$$

Vì M_a thực hiện phép nhân modulo nên để thiết kế mạch cho M_a ta có thể chuyển mạch logic thực hiện phép nhân và chia lấy dư thành mạch lượng tử theo kỹ thuật ở Phần 4.4. Hơn nữa, vì các phép nhân và chia các số n bit cần $O(n^2)$ cổng logic nên mạch lượng tử cho M_a cũng chỉ cần $O(n^2)$ cổng lượng tử cơ bản.

Ví dụ 6.3.3. Cho $a = 2, N = 7$ nguyên tố cùng nhau, phép toán lượng tử $M_a = M_2$ trên $n = \lceil \log 7 \rceil = 3$ qubit được định nghĩa là

$$M_2 |x\rangle = \begin{cases} |2x \bmod 7\rangle & 0 \leq x < 7, \\ |x\rangle & 7 \leq x < 8. \end{cases}$$

Cụ thể

$$\begin{array}{c|c|c|c|c|c|c|c|c|c} |x\rangle & |0\rangle & |1\rangle & |2\rangle & |3\rangle & |4\rangle & |5\rangle & |6\rangle & |7\rangle \\ \hline M_2 |x\rangle & |0\rangle & |2\rangle & |4\rangle & |6\rangle & |1\rangle & |3\rangle & |5\rangle & |7\rangle \end{array}$$

Chẳng hạn $M_2 |4\rangle = |2 \cdot 4 \bmod 7\rangle = |1\rangle$ và $M_2 |7\rangle = |7\rangle$ do $7 \geq N = 7$.

Ta nhận thấy M_2 hoán vị các vector cơ sở tính toán nên M_2 unita. Cụ thể hơn, M_2 có ma trận biểu diễn là ma trận unita như sau (♣)

$$M_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Chu kỳ của $2^x \pmod{7}$ là $r = 3$, với $\omega_3 = e^{i2\pi\frac{1}{3}} = e^{i\frac{2\pi}{3}}$, M_2 có các vector riêng và trị riêng tương ứng là

$$\begin{aligned} |v_0\rangle &= \frac{1}{\sqrt{3}}(|1\rangle + |2\rangle + |4\rangle), & \lambda_1 &= \omega_3^0 = 1, \\ |v_1\rangle &= \frac{1}{\sqrt{3}}(|1\rangle + e^{i\frac{-2\pi}{3}}|2\rangle + e^{i\frac{-4\pi}{3}}|4\rangle), & \lambda_2 &= \omega_3^1 = e^{i\frac{2\pi}{3}}, \\ |v_2\rangle &= \frac{1}{\sqrt{3}}(|1\rangle + e^{i\frac{-4\pi}{3}}|2\rangle + e^{i\frac{-8\pi}{3}}|4\rangle), & \lambda_2 &= \omega_3^2 = e^{i\frac{4\pi}{3}}. \end{aligned}$$

Ta có

$$\begin{aligned} \frac{|v_0\rangle + |v_1\rangle + |v_2\rangle}{\sqrt{3}} &= \frac{1+1+1}{3}|1\rangle + \frac{1+e^{i\frac{-2\pi}{3}}+e^{i\frac{-4\pi}{3}}}{3}|2\rangle + \frac{1+e^{i\frac{-4\pi}{3}}+e^{i\frac{-8\pi}{3}}}{3}|4\rangle \\ &= |1\rangle. \end{aligned}$$

□

6.3.4 Thuật toán lượng tử tìm chu kỳ lũy thừa modulo

Bây giờ, cho các số nguyên dương a, N nguyên tố cùng nhau, ta đã sẵn sàng vận dụng khung thuật toán ước lượng pha ở Phần 6.2 để tìm chu kỳ của $a^x \pmod{N}$ một cách hiệu quả.

Trước hết, mạch U trong ước lượng pha chính là mạch lượng tử cho phép nhân a modulo N , là M_a định nghĩa ở (6.12). Khung ước lượng pha cần mạch cho dãy cổng điều khiển $U, U^2, \dots, U^{2^{m-1}}$. Áp dụng (6.6), với $U = M_a$, ta thấy dãy cổng điều khiển biến $|z\rangle|x\rangle$, với $z = z_{m-1} \dots z_1 z_0 \in \mathbb{B}^m$ thành

$$|z\rangle U^z |x\rangle = |z\rangle |a^z x \pmod{N}\rangle. \quad (6.8)$$

Bằng cách biến mạch cổ điển tính lũy thừa và nhân modulo thành mạch lượng tử, ta có mạch cho dãy cổng điều khiển cần trong ước lượng pha với độ phức tạp $O(n^3)$ nếu số qubit thanh ghi trị riêng là $m = O(n)$.

Tiếp theo ta cần chuẩn bị vector riêng $|v\rangle$ cho $U = M_a$. Nếu dùng $|v_1\rangle$ thì trị riêng tương ứng là $\omega_r = e^{2\pi i\frac{1}{r}}$, tức pha là $\theta = \frac{1}{r}$. Ước lượng được θ thì tính được r . Tuy nhiên ta không biết r nên không chuẩn bị được trạng thái $|v_1\rangle$. Tương tự ta cũng không chuẩn bị được các vector riêng $|v_s\rangle$ khác ($s = 1, 2, \dots, r-1$). Thay vì dùng một vector riêng $|v_s\rangle$ nào đó, ta sẽ dùng trạng thái

$$|1\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |v_s\rangle$$

là trạng thái tổ hợp đều của các vector riêng $|v_s\rangle$ (với trị riêng tương ứng $\omega_r^s = e^{2\pi i\frac{s}{r}}$, tức pha tương ứng $\theta_s = \frac{s}{r}$). Dĩ nhiên trạng thái $|1\rangle$ rất dễ chuẩn bị (chỉ cần dùng

cổng X để chuyển $|0\rangle$ thành $|1\rangle$ cho qubit thấp nhất của thanh ghi vector riêng). Mặt khác, nếu dùng $|1\rangle$ cho thanh ghi vector riêng thì thuật toán ước lượng pha sẽ “chọn” ước lượng một trong r pha $\theta_0, \theta_1, \dots, \theta_{r-1}$ với xác suất $\frac{1}{r}$ (Bài tập 6.8).

Còn một vấn đề “nhỏ” là nếu có ước lượng cho $\theta_s = \frac{s}{r}$ làm sao tìm được r khi không biết cả r lẫn s . Rất may, ta có thể dùng kĩ thuật khai triển **liên phân số** (continued fraction). Một ví dụ cụ thể của thuật này được minh họa trong Ví dụ 6.3.4. Kĩ thuật này đảm bảo, với $m = O(n)$, từ kết quả ước lượng pha $\theta \approx \frac{s}{r}$ ta tìm được phân số $\frac{s'}{r'}$ tối giản sao cho $\frac{s'}{r'} = \frac{s}{r}$. Lưu ý, khi đó r' vẫn có thể nhỏ hơn r nếu s, r có ước chung lớn hơn 1. Tuy nhiên, ta có thể kiểm tra r' có phải là chu kỳ hay không bằng cách kiểm tra $a^{r'} = 1 \pmod{N}$ hay không. Nếu r' không phải là chu kỳ, ta có thể chạy lại thủ tục ước lượng pha.

Một cách tốt hơn, ta có thể chạy thủ tục ước lượng pha hai lần để được s'_1, r'_1 ở lần 1 và s'_2, r'_2 ở lần 2. Nếu s'_1, s'_2 không nguyên tố cùng nhau thì ta lấy r là bội chung nhỏ nhất của r'_1, r'_2 . Người ta chứng minh được xác suất để việc này xảy ra ít nhất là $\frac{1}{4}$. Do đó, bằng cách chạy “vài lần” ta sẽ có r . Tóm lại, nếu dùng số qubit ước lượng $m = O(n)$ thì mạch ước lượng pha có độ phức tạp $O(n^3)$, cùng với độ phức tạp $O(n^3)$ của khai triển liên phân số và $O(1)$ lần chạy lại ta có thể tìm được chu kỳ r với tổng cộng $O(n^3)$ thao tác cơ bản.

Ví dụ 6.3.4. Liên phân số (continued fractions) là số có dạng

$$a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\ddots + \frac{1}{a_K}}}} \quad (6.9)$$

với a_1, a_2, \dots, a_K là các số nguyên dương và a_0 là số nguyên. Liên phân số (6.9) còn được mô tả gọn bằng danh sách $[a_0, a_1, a_2, \dots, a_K]$. Với $0 \leq k \leq K$, ta nói liên phân số $[a_0, \dots, a_k]$ là số tụ (convergent) thứ k của liên phân số (6.9).

Ví dụ, để biểu diễn phân số $\frac{3}{8}$ bằng liên phân số, trước hết ta tách phần nguyên và phần dư để được

$$\frac{3}{8} = 0 + \frac{3}{8}.$$

Sau đó ta nghịch đảo phần dư để được

$$\frac{3}{8} = 0 + \frac{1}{\frac{8}{3}}.$$

Ta lặp lại việc tách (bây giờ là cho phân số $\frac{3}{8}$)

$$\frac{3}{8} = 0 + \frac{1}{2 + \frac{2}{3}}$$

và nghịch đảo cho đến khi được phân số có tử là 1

$$\frac{3}{8} = 0 + \frac{1}{2 + \frac{1}{1 + \frac{1}{2}}}.$$

Như vậy $\frac{3}{8}$ có biểu diễn liên phân số là $[0, 2, 1, 2]$ với các số tự

- thứ 0: $[0] = 0$,
- thứ 1: $[0, 2] = 0 + \frac{1}{2} = \frac{1}{2} = 0.5$,
- thứ 2: $[0, 2, 1] = 0 + \frac{1}{2 + \frac{1}{1}} = \frac{1}{3} = 0.33333\dots$,
- thứ 3: $[0, 2, 1, 2] = \frac{3}{8} = 0.375$.

Bây giờ, giả sử ta dùng ước lượng pha để tìm chu kỳ của $a^x \pmod{N}$ với $a = 2, N = 7$ dùng $m = 3$ qubit ước lượng. Ta biết rằng chu kỳ cần tìm là $r = 3$ nên mạch sẽ chọn ngẫu nhiên để đo pha $\frac{s}{r}$, $s = 0, \dots, r-1$ tức là $0, \frac{1}{3}, \frac{2}{3}$. Vì dùng 3 qubit ước lượng nên thủ tục sẽ cho ra các xấp xỉ 3 bit của các giá trị này (với xác suất cao) là $0, \frac{3}{8} = 0.375, \frac{5}{8} = 0.625$.

Nếu thủ tục cho ra $\frac{3}{8}$ thì dùng khai triển liên phân số ở trên cho $\frac{3}{8}$ ta có các số tự dạng $\frac{s'}{r'}$ vì chu kỳ $r < N$ nên ta chọn số tự cao nhất (xấp xỉ tốt nhất) với mẫu $r' < N$ và kiểm tra lại r' có là chu kỳ, trong trường hợp này là $\frac{1}{3}$ với $r' = 3$ và là chu kỳ. Tương tự, nếu thủ tục cho ra $\frac{5}{8}$ thì vì khai triển liên phân số của $\frac{5}{8}$ là $[0, 1, 1, 2]$ với các số tự lần lượt là $0, \frac{1}{1}, \frac{1}{2}, \frac{2}{3}, \frac{5}{8}$ nên ta chọn $r' = 3$ cũng là chu kỳ.

Trường hợp thủ tục cho ra 0 ta không có manh mối gì để tìm r nên ta cần chạy lại thủ tục. Hơn nữa, thủ tục cũng có thể cho ra các xấp xỉ không phải tốt nhất (xác suất nhỏ), chẳng hạn nếu thủ tục cho ra $\frac{6}{8}$ (là một xấp xỉ 3 bit khác của $\frac{2}{3}$) thì vì $\frac{6}{8}$ có biểu diễn liên phân số là $[0, 1, 3]$ với các số tự lần lượt là $0, \frac{1}{1}, \frac{3}{4}$ nên $r' = 4$ tuy nhiên 4 không phải chu kỳ nên ta cũng cần chạy lại thủ tục. Tóm lại, giả sử thủ tục ước lượng pha trả về xấp xỉ 3 bit tốt nhất của pha cần tính thì các trường hợp chạy thủ tục tìm chu kỳ được mô tả như trong Bảng 6.1.

□

s	s/r	Ước lượng tốt nhất	Số tự liên phân số	r
0	$0/3 = 0$	$0/8 = 0$	-	(chạy lại)
1	$1/3 = 0.33...$	$3/8 = 0.375$	$1/3$	3
2	$2/3 = 0.66...$	$5/8 = 0.625$	$2/3$	3

Bảng 6.1: Các trường hợp chạy thử tục tìm chu kỳ với $a = 2, N = 7, m = 3$.

6.3.5 Phân tích số nguyên

Hệ mã RSA ở Ví dụ 5.2.2 được thiết lập dựa trên con số $N = pq$ với p, q là các số nguyên tố. Trong hệ mã này, N được công khai nhưng p, q phải bí mật. Thuật toán tốt nhất của tính toán cổ điển phân tích được N thành p, q có độ phức tạp trên đa thức nên việc phân tích số nguyên là “bài toán khó” với tính toán cổ điển (xem Ví dụ 2.3.5). Đây là lý do mà hệ mã RSA được dùng trong thực tế.

Tuy nhiên, năm 1994, Peter Shor phát minh một thuật toán phân tích N thành các thừa số p, q có độ phức tạp đa thức theo n . Đây là một bước ngoặt mà đóng góp chủ đạo chính là thuật toán lượng tử tìm chu kỳ lũy thừa modulo. Cụ thể, để phân tích $N = pq$, thuật toán Shor được tiến hành qua 3 bước

1. *Bước 1.* Chọn ngẫu nhiên số nguyên $1 < a < N$.

(a) Nếu $\gcd(a, N) \neq 1$ thì

$$p = \gcd(a, N), \quad q = \frac{N}{p}$$

ta may mắn phân tích xong!

(b) Nếu $\gcd(a, N) = 1$, qua Bước 2.

2. *Bước 2.* Dùng thuật toán lượng tử ở phần trước tìm chu kỳ r của $a^x \pmod{N}$.

(a) Nếu r lẻ, quay về Bước 1 để chọn số a khác.

(b) Nếu r chẵn, tính $a^{r/2} \pmod{N}$.

i. Nếu $a^{r/2} = N - 1 \pmod{N}$, quay về Bước 1 để chọn số a khác.

ii. Nếu $a^{r/2} \neq N - 1 \pmod{N}$, qua Bước 3.

3. *Bước 3.* Tính p, q từ a, N, r . Vì $a^r = 1 \pmod{N}$ nên $a^r - 1$ là bội của N . Hơn nữa, r chẵn nên

$$a^r - 1 = (a^{r/2})^2 - 1^2 = (a^{r/2} - 1)(a^{r/2} + 1).$$

Do đó $(a^{r/2} - 1)(a^{r/2} + 1)$ là bội của N . Tới đây ta có (Bài tập 6.14)

$$p = \gcd(a^{r/2} - 1, N), \quad q = \gcd(a^{r/2} + 1, N).$$

Ta thấy, ngoại trừ Bước 2 là tìm chu kỳ r của $a^x \pmod{N}$, các bước khác đều có thể được tiến hành bằng tính toán cổ điển rất hiệu quả (phép tính ước chung lớn nhất gcd hay phép mũ modulo đều có độ phức tạp đa thức theo n). Dùng tính toán lượng tử để tính chu kỳ thì Bước 2 cũng rất hiệu quả. Hơn nữa, Shor cho thấy số lần thực hiện Bước 2, tức là số lần “quay lui” ở 2.(a) và 2.(b)i là không nhiều nên Thuật toán trên rất hiệu quả.

Do phân tích $N = pq$ rất hiệu quả nên thuật toán Shor có khả năng “bẻ gãy” RSA là hệ mã hóa khóa công khai đang được dùng phổ biến hiện nay. Để chuẩn bị cho sự ra đời của máy tính lượng tử, nhiều hệ mã và giao thức mã đã được nghiên cứu để chống lại các thuật toán lượng tử dạng này, gọi chung là **mật mã kháng lượng tử** (quantum-resistant cryptography) hay **mật mã hậu lượng tử** (Post-Quantum Cryptography, PQC).³

6.4 Mã Shor

6.4.1 Mã sửa lỗi

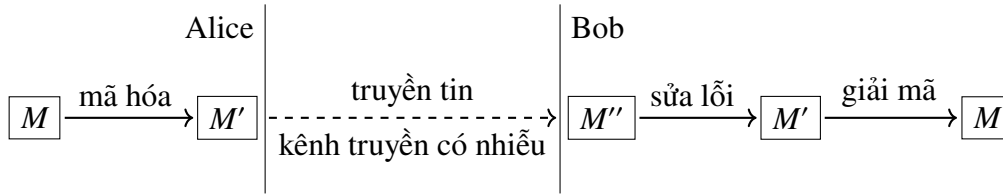
Ta đã biết, trong tính toán cổ điển, thông tin được biểu diễn bằng các chuỗi bit. Khi truyền tin, lưu trữ hay xử lý, các bit này có thể gặp **lỗi** (error) và bị thay đổi giá trị (lật bit, $0 \leftrightarrow 1$).⁴ Chẳng hạn, nếu bên gửi dùng còi để truyền tin (0 là tit, 1 là te) thì do khiếm khuyết của thiết bị (còi bị rè) hay nhiễu của môi trường (tiếng ồn) mà bên nhận có thể nhận sai bit được gửi. Phương tiện cùng với môi trường truyền tin có thể tạo nên lỗi khi truyền được gọi chung là **kênh truyền có nhiễu** (noisy channel).

Giả sử Alice muốn truyền thông điệp là chuỗi bit M cho Bob qua một kênh truyền có nhiễu. Một cách hệ thống, các bước của một quy trình truyền thông tin cậy được mô tả trong Hình 6.12. Đầu tiên, Alice mã hóa chuỗi bit M thành chuỗi bit M' . Quá trình truyền tin M' có thể có lỗi nên chuỗi bit mà Bob nhận là M'' có thể khác M' . Bob sau khi nhận M'' thực hiện việc sửa lỗi để có M' rồi giải mã để có M là thông điệp mà Alice muốn truyền. Hệ thống này được gọi là hệ **mã sửa lỗi** (error-correction code). Trong trường hợp việc truyền tin không tổn kém, Bob có thể chỉ cần dùng hệ thống **mã phát hiện lỗi** (error-detection code). Trong đó, khi nhận M'' , Bob có khả năng phát hiện được M'' bị lỗi hay không. Nếu có lỗi, Bob yêu cầu Alice gửi lại M' ; nếu không có lỗi (M'' là M'), Bob giải mã M'' để được M . Dĩ nhiên, nếu đảm bảo việc truyền tin không có lỗi thì hệ thống chỉ cần mã

³thuật ngữ chơi chữ Y2Q đề cập đến thời điểm mà máy tính lượng tử đủ năng lực để phá các hệ thống mật mã không có tính kháng lượng tử hiện tại như RSA.

⁴lỗi lật bit là lỗi “mức thấp”, các lỗi khác như mất bit hoặc thêm bit có thể được xem là lỗi “mức cao”. Hơn nữa, việc truyền tin có thể xem là theo từng bit nên chỉ có một loại lỗi là lật bit.

hóa/giải mã như Hình 2.1.



Hình 6.12: Quy trình của hệ thống mã sửa lỗi.

Ví dụ 6.4.1. (Mã chẵn lẻ) Trong Ví dụ 2.1.3, mỗi kí tự được mã bằng mã ASCII là chuỗi gồm 1 byte để dễ truyền tin (đơn vị truyền tin thường là byte chứ không phải bit). Thật ra, mã gốc ASCII chỉ dùng chuỗi 7 bit cho mỗi kí tự. Để có khả năng phát hiện lỗi, ta có thể dùng thêm **bit chẵn lẻ** (parity bit) cho mỗi chuỗi 7 bit. Cụ thể, quá trình mã hóa kí tự c gồm các bước

1. Tra bảng mã ASCII để có 7 bit cho c là $b = b_6 \dots b_1 b_0$.
2. Tính bit chẵn lẻ

$$p = \text{PARITY}(b) = b_6 \oplus \dots \oplus b_1 \oplus b_0.$$

Lưu ý, $p = 1$ nếu số bit 1 của b là lẻ, ngược lại $p = 0$.

3. Tạo mã $w = bp = b_6 \dots b_1 b_0 p$ gồm 8 bit cho kí tự c .

Nhận xét là mã w của mỗi kí tự đều có $\text{PARITY}(w) = 0$. (♣)

Để truyền thông điệp là chuỗi kí tự, Alice mã mỗi kí tự thành 1 byte như trên và truyền đi cho Bob. Chẳng hạn, thông điệp “Ok!” được mã bằng 3 byte sau (bit chẵn lẻ được bôi đậm)

$$\underbrace{10011111}_{0} \underbrace{11010111}_{k} \underbrace{01000010}_{!}$$

Giả sử khi truyền tin, mỗi byte bị lỗi không quá 1 bit thì Bob có khả năng phát hiện lỗi. Cụ thể, nếu byte w Bob nhận được có $\text{PARITY}(w) = 1$ thì w bị lỗi (♣) và Bob có thể yêu cầu Alice gửi lại. Nếu $\text{PARITY}(w) = 0$ thì Bob có thể giải mã w bằng cách lấy 7 bit trái và tra kí tự trong bảng mã ASCII. Lưu ý, mặc dù có thể phát hiện lỗi nhưng Bob không thể sửa lỗi. Hơn nữa, hệ thống mã này chỉ cho phép phát hiện lỗi khi có không quá 1 bit bị lỗi trong mỗi byte. (♣) \square

6.4.2 Mã lặp

Cách đơn giản nhất để phát hiện và sửa lỗi là sử dụng **mã lặp** (repetition code), trong đó, mỗi bit được lặp lại k lần khi gửi. Cụ thể,

$$0 \text{ được mã thành } 0^k = \underbrace{00\dots0}_k, \quad 1 \text{ được mã thành } 1^k = \underbrace{11\dots1}_k.$$

Các chuỗi $0^k, 1^k$ còn được gọi là **từ mã** (codeword). Khi truyền tin, nếu chuỗi bit nhận được không thuần (khác 0^n hay 1^n) thì có lỗi. Hơn nữa, nếu số bit lỗi trong mỗi từ mã nhỏ hơn $\frac{k}{2}$, ta có thể sửa lỗi bằng cách dùng giá trị bit chiếm số đông (majority vote).

Chẳng hạn, với $k = 3$, ta dùng 3 bit (thường được gọi là bit vật lý) để mã cho 1 bit (thường được gọi là bit logic) theo qui tắc

$$0 \text{ được mã thành } 000, \quad 1 \text{ được mã thành } 111.$$

Nếu trong quá trình truyền tin, mỗi từ mã bị lỗi không quá 1 bit, ta có thể sửa lỗi như sau

- $\{100, 010, 100\}$ được sửa thành 000 và giải mã thành 0.
- $\{011, 101, 110\}$ được sửa thành 111 và giải mã thành 1.

Lưu ý, ta không thể sửa lỗi được nếu có từ 2 lỗi trở lên, chẳng hạn, ta không biết được chuỗi 100 là do 000 bị 1 lỗi biến thành hay do 111 bị 2 lỗi biến thành. Cũng lưu ý, nếu $k = 2$ thì ta không sửa lỗi được dù chỉ có 1 bit lỗi.

Đặc biệt, ta có thể sử dụng tính chẵn lẻ của chuỗi bit để phát hiện và sửa lỗi mà không cần đọc giá trị của các bit! Cụ thể, với $k = 3$, gọi 3 bit nhận được khi truyền tin là $b_2b_1b_0$, ta có cách phát hiện lỗi và sửa lỗi như Bảng 6.2.

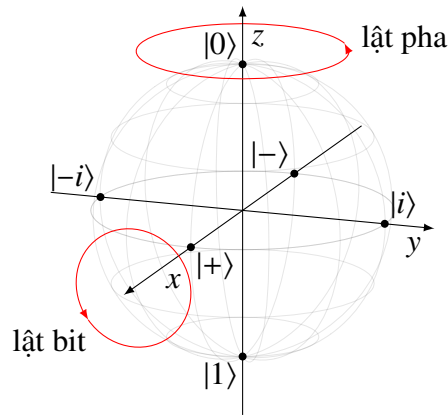
Chuỗi bit $b_2b_1b_0$	$b_2 \oplus b_1$	$b_1 \oplus b_0$	Sửa lỗi
000	0	0	(không lỗi)
111	0	0	
001	0	1	Lật bit phải b_0
110	0	1	
011	1	0	Lật bit trái b_2
100	1	0	
010	1	1	Lật bit giữa b_1
101	1	1	

Bảng 6.2: Phát hiện và sửa lỗi của mã lặp 3 bit.

Mã lặp có khả năng sửa nhiều lỗi hơn khi dùng nhiều bit lặp hơn (tăng k). Tuy nhiên, việc dùng nhiều bit vật lý lại làm tăng khả năng bị lỗi. Do đó ta cũng cần phân tích khả năng sửa lỗi cho các kênh truyền tin khác nhau để chọn dùng k hợp lý (Bài tập 6.17). Hơn nữa, việc dùng nhiều bit vật lý sẽ tăng dung lượng khi truyền. Ở các khía cạnh này, mã lặp không phải là mã hiệu quả.

6.4.3 Mã sửa lỗi trong tính toán lượng tử

Trong tính toán cổ điển, một bit chỉ có thể nhận một trong 2 giá trị là 0 hoặc 1. Khi bit bị lỗi, nó chỉ có thể bị đổi giá trị (lật bit, $0 \leftrightarrow 1$). Trong tính toán lượng tử, một qubit có thể nhận vô số trạng thái khác nhau ứng với các điểm khác nhau trên mặt cầu Bloch. Khi qubit bị lỗi, nó đã “di chuyển” đến điểm khác trên mặt cầu Bloch.



Hình 6.13: Phép biến đổi lật bit và lật pha.

Trên mặt cầu Bloch, bit cổ điển ứng với 1 trong 2 cực là cực Bắc ($|0\rangle$) hoặc cực Nam ($|1\rangle$) và chỉ có thể bị lỗi lật bit hoàn toàn, ứng với phép quay quanh trục x góc π (chính là tác động của cổng X). Tổng quát hơn, các qubit cũng có thể bị lỗi lật bit, nghĩa là bị quay quanh trục x với góc θ có thể khác π , gọi là **lật bit một phần** (partial bit flip) mà cũng được gọi chung là lỗi **lật bit** (bit flip).

Khác với bit cổ điển, các qubit cũng có thể bị lỗi dạng khác, chẳng hạn bị **lật pha** (phase flip), ứng với phép quay quanh trục z một góc θ . Nếu $\theta = \pi$, phép lật pha là hoàn toàn (chính là tác động của cổng Z), biến $|+\rangle$ thành $|-\rangle$ và ngược lại. Nếu $\theta \neq \pi$, phép lật pha là một phần, chẳng hạn cổng S thực hiện phép quay quanh trục z góc $\theta = \frac{\pi}{2}$, biến $|+\rangle$ thành $|i\rangle$ và $|-\rangle$ thành $|-i\rangle$.

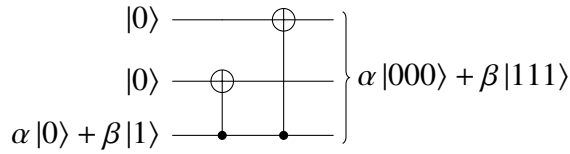
Rõ ràng, qubit rất nhạy cảm so với bit cổ điển, chỉ cần một tác động nhỏ từ môi trường cũng có thể làm cho qubit thay đổi trạng thái. Việc bị thay đổi trạng thái

theo cách không mong muốn do chịu tác động của nhiễu từ môi trường thường được gọi là sự **mất gắn kết lượng tử** (quantum decoherence). Quá trình này có thể làm mất đi tính chồng chất hay vướng của các qubit nên là trở ngại lớn nhất khi xây dựng các máy tính lượng tử quy mô lớn.

Phần tiếp theo trình bày cách phát hiện và sửa lỗi lượng tử. Lưu ý, lật pha và lật bit được nghiên cứu kĩ vì các phép toán lượng tử khác có thể được qui về như $Y = iXZ$ (vừa lật pha vừa lật bit chính là lật pha rồi lật bit) và đây là 2 biến đổi “độc lập” theo nghĩa cái này không ảnh hưởng đến cái kia (có thể hình dung lật bit chính là thay đổi vĩ độ còn lật pha là thay đổi kinh độ trên mặt cầu Bloch).

6.4.4 Mã lật bit

Để sửa lỗi lật bit, qubit logic $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ được mã bằng 3 qubit vật lý q_2, q_1, q_0 có trạng thái $\alpha|000\rangle + \beta|111\rangle$. Việc này có thể được thực hiện bằng mạch sau (♣)



Lưu ý, tương tự như mã lặp 3 bit, các ket cơ sở $|0\rangle, |1\rangle$ được mã tương ứng thành $|000\rangle, |111\rangle$ nhưng ket $|\psi\rangle$ nói chung không được mã thành $|\psi\rangle|\psi\rangle|\psi\rangle$; việc này, như ta đã biết là bị cấm bởi định lý không nhân bản (Phần 4.5).

Giả sử khi truyền tin, qubit q_2 bị lật bit hoàn toàn, ta có sự thay đổi trạng thái

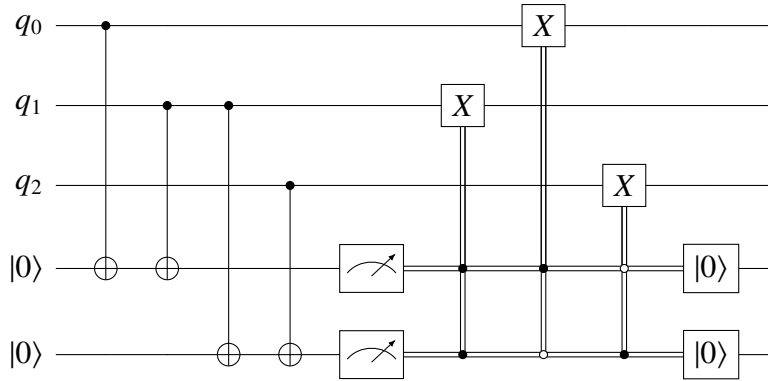
$$\alpha|000\rangle + \beta|111\rangle \xrightarrow{X \otimes I \otimes I} \alpha|100\rangle + \beta|011\rangle.$$

Khi đó, nếu biết q_2 bị lật bit, ta sửa lỗi bằng cách đơn giản là lật lại bit q_2

$$\alpha|100\rangle + \beta|011\rangle \xrightarrow{X \otimes I \otimes I} \alpha|000\rangle + \beta|111\rangle.$$

Tương tự, nếu biết qubit bị lật bit là q_1, q_0 ta thực hiện thao tác sửa lỗi tương ứng là $I \otimes X \otimes I, I \otimes I \otimes X$.

Tuy nhiên, làm sao ta biết được có qubit nào bị lật bit hay không và nếu có thì là qubit nào? Ta không được phép đo các qubit vì khi đo trạng thái tổ hợp bị sụp đổ nên ta không còn thông tin trạng thái ban đầu (không còn α, β). Rất may, tương tự mã lặp, ta có thể phát hiện lỗi nhờ tính chẵn lẻ mà không cần “đọc” trực tiếp các qubit. Nhớ rằng, $\text{CNOT } |a\rangle|b\rangle = |a\rangle|a \oplus b\rangle$, phiên bản lượng tử của thuật toán phát hiện và sửa lỗi ở Bảng 6.2 được cho như mạch ở Hình 6.14.



Hình 6.14: Mạch sửa mã lật bit lượng tử 3.

Khi 2 qubit dưới cùng được đo ta biết được tính chẵn lẻ của q_2, q_1 và q_1, q_0 . Lưu ý, phép đo không ảnh hưởng đến trạng thái của 3 qubit trên. Nếu kết quả đo là 11 ta biết q_1 bị lật bit nên ta áp cổng X lên q_1 để sửa lỗi. Các trường hợp khác tương tự, đặc biệt nếu kết quả đo là 00 ta biết không có lỗi. Sau cùng ta đặt lại trạng thái cho 2 qubit dưới là $|0\rangle$ để tái sử dụng (chẳng hạn để sửa lỗi các bộ 3 qubit khác).

Nếu lỗi không phải là phép lật bit hoàn toàn mà là phép lật bit một phần, tức là phép quay qubit quanh trục x một góc θ nào đó, thì việc phát hiện và sửa lỗi diễn ra như thế nào? Trước hết, phép quay qubit quanh trục x một góc θ có tác động trên các ket cơ sở là (Bài tập 6.20)

$$\begin{aligned} |0\rangle &\rightarrow i\sqrt{1-\varepsilon^2}|0\rangle + \varepsilon|1\rangle, \\ |1\rangle &\rightarrow \varepsilon|0\rangle + i\sqrt{1-\varepsilon^2}|1\rangle, \end{aligned}$$

với $\varepsilon = \sin(\frac{\theta}{2})$. Phép lật bit hoàn toàn ứng với $\theta = \pi$, tức $\varepsilon = 1$, $|0\rangle \leftrightarrow |1\rangle$.

Rất thú vị, mạch 6.14 cũng có thể được dùng để sửa lỗi trong trường hợp này. Chẳng hạn, khi truyền tin, nếu qubit q_2 bị lật bit một phần thì 3 qubit q_2, q_1, q_0 có trạng thái $\alpha|000\rangle + \beta|111\rangle$ bị biến thành

$$\alpha i\sqrt{1-\varepsilon^2}|000\rangle + \alpha\varepsilon|100\rangle + \beta\varepsilon|011\rangle + \beta i\sqrt{1-\varepsilon^2}|111\rangle. \quad (6.10)$$

Khi đưa trạng thái này vào mạch 6.14, cùng với trạng thái $|00\rangle$ của 2 qubit dưới, ta có trạng thái đầu vào của mạch là

$$\alpha i\sqrt{1-\varepsilon^2}|00000\rangle + \alpha\varepsilon|00100\rangle + \beta\varepsilon|00011\rangle + \beta i\sqrt{1-\varepsilon^2}|00111\rangle.$$

Trạng thái của mạch trước khi đo là (♣)

$$\alpha i\sqrt{1-\varepsilon^2}|00000\rangle + \alpha\varepsilon|10100\rangle + \beta\varepsilon|10011\rangle + \beta i\sqrt{1-\varepsilon^2}|00111\rangle.$$

Bây giờ, khi đo 2 qubit dưới, ta có 2 trường hợp

(i) đo được 00 với xác suất

$$|\alpha i \sqrt{1 - \varepsilon^2}|^2 + |\beta i \sqrt{1 - \varepsilon^2}|^2 = 1 - \varepsilon^2$$

và 3 qubit trên sụp về trạng thái

$$Ai \sqrt{1 - \varepsilon^2}(\alpha|000\rangle + \beta|111\rangle) \equiv \alpha|000\rangle + \beta|111\rangle.$$

Phép đo đã đưa qubit về đúng trạng thái được gửi!

(ii) đo được 10 với xác suất

$$|\alpha \varepsilon|^2 + |\beta \varepsilon|^2 = \varepsilon^2$$

và 3 qubit trên sụp về trạng thái

$$B\varepsilon(\alpha|100\rangle + \beta|011\rangle) \equiv \alpha|100\rangle + \beta|011\rangle.$$

Phép đo này đưa qubit q_2 về dạng lật bit toàn phần. Do đó, việc áp dụng cổng $(X \otimes I \otimes I)$ đưa về đúng trạng thái được gửi.

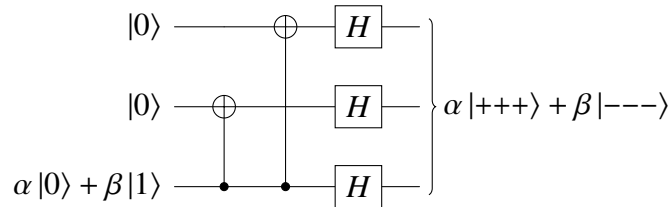
Các trường hợp lật bit khác cũng áp dụng được mạch 6.14.

6.4.5 Mã lật pha

Từ Hình 6.13, ta thấy phép lật pha có thể xem như phép lật bit nếu ta quay trục z thành trục x . Đó chính là biến cơ sở Hadamard thành cơ sở tính toán và cổng H thực hiện việc này. Như vậy ta có thể đưa việc sửa lỗi lật pha về sửa lỗi lật bit khi đổi cơ sở. Cụ thể, nếu như phép lật bit toàn phần X thực hiện biến đổi $|0\rangle \leftrightarrow |1\rangle$ thì phép lật pha toàn phần Z thực hiện biến đổi $|+\rangle \leftrightarrow |-\rangle$. Do đó, một cách tự nhiên, trong mã lật pha, trạng thái $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ được mã bằng trạng thái sau dùng 3 qubit vật lý

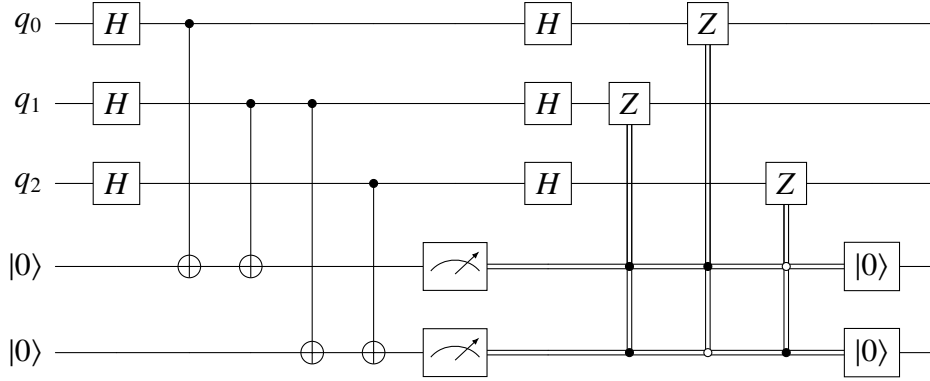
$$\alpha|+++ \rangle + \beta|--- \rangle.$$

Để thực hiện mã hoá ta có thể dùng mạch sau (♣)



Vì H chuyển đổi giữa $\{|0\rangle, |1\rangle\}$ và $\{|+\rangle, |-\rangle\}$ nên mạch này chính là mạch mã hóa trong lật bit nhưng có thêm “lớp” các cổng H ở cuối.

Vì tác động của lật pha lên $|+\rangle, |-\rangle$ tương tự như tác động của lật bit lên $|0\rangle, |1\rangle$ nên mạch sửa mã lật bit ở Hình 6.14 có thể được sửa đổi để làm mạch sửa mã lật pha ở Hình 6.15. Lớp cổng H đầu tiên thực hiện việc đổi cơ sở từ $\{|0\rangle, |1\rangle\}$ sang $\{|+\rangle, |-\rangle\}$ và lớp cổng H sau đó thực hiện việc đổi ngược lại. Việc sửa lỗi áp dụng các cổng lật pha toàn phần Z thay cho lật bit toàn phần X .



Hình 6.15: Mạch sửa mã lật pha lượng tử 3.

Ta kiểm tra để thấy mạch 6.15 phát hiện và sửa được lỗi lật pha. Trước hết, phép quay qubit quanh trục z một góc θ có tác động trên các ket cơ sở là (Bài tập 6.21)

$$\begin{aligned} |0\rangle &\rightarrow (i\sqrt{1-\varepsilon^2} + \varepsilon)|0\rangle, \\ |1\rangle &\rightarrow (i\sqrt{1-\varepsilon^2} - \varepsilon)|1\rangle, \end{aligned}$$

với $\varepsilon = \sin(\frac{\theta}{2})$. Phép lật pha hoàn toàn ứng với $\theta = \pi, \varepsilon = 1, |0\rangle \rightarrow |0\rangle, |1\rangle \rightarrow -|1\rangle$. Từ đó, tác động của phép quay trên các ket $|+\rangle, |-\rangle$ là

$$\begin{aligned} |+\rangle &\rightarrow i\sqrt{1-\varepsilon^2}|+\rangle + \varepsilon|-\rangle, \\ |-\rangle &\rightarrow \varepsilon|+\rangle + i\sqrt{1-\varepsilon^2}|-\rangle, \end{aligned}$$

Phép lật pha hoàn toàn ứng với $\theta = \pi, \varepsilon = 1, |+\rangle \leftrightarrow |-\rangle$.

Bây giờ, giả sử khi truyền tin, qubit q_2 bị lật pha một phần thì 3 qubit q_2, q_1, q_0 có trạng thái $\alpha|+++ \rangle + \beta|--- \rangle$ bị biến thành

$$\alpha i\sqrt{1-\varepsilon^2}|+++ \rangle + \alpha\varepsilon|+-- \rangle + \beta\varepsilon|+-+ \rangle + \beta i\sqrt{1-\varepsilon^2}|--- \rangle.$$

Khi đưa trạng thái này vào mạch 6.15 thì lớp cổng H đầu tiên biến trạng thái trên thành trạng thái (6.10). Hoạt động tiếp đó tương tự như mạch sửa lỗi lật bit. Khi đo 2 qubit dưới, ta có 2 trường hợp

- (i) đo được 00 với xác suất $1-\varepsilon^2$ và 3 qubit trên sụp về trạng thái $\alpha|000\rangle + \beta|111\rangle$ mà qua lớp cổng H sau đó biến thành $\alpha|+++ \rangle + \beta|--- \rangle$. Phép đo đã đưa qubit về đúng trạng thái được gửi!
- (ii) đo được 10 với xác suất ε^2 và 3 qubit trên sụp về trạng thái $\alpha|100\rangle + \beta|011\rangle$ mà qua lớp cổng H sau đó biến thành $\alpha| -++ \rangle + \beta| +-- \rangle$. Phép đo này đưa qubit q_2 về dạng lật pha toàn phần. Do đó, việc áp dụng cổng $(Z \otimes I \otimes I)$ đưa về đúng trạng thái được gửi.

6.4.6 Mã Shor

Năm 1995, Peter Shor đề xuất hệ mã kết hợp mã lật bit và mã lật pha sửa được cả hai loại lỗi, gọi là mã Shor. Mã Shor đã đặt nền tảng cho hệ thống các kĩ thuật **sửa lỗi lượng tử** (Quantum Error Correction, QEC).

Để mã trạng thái $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, trước hết, 3 qubit với trạng thái sau được dùng như mã lật pha ($|0\rangle \rightarrow |+++ \rangle, |1\rangle \rightarrow |--- \rangle$)

$$\begin{aligned} \alpha|+++ \rangle + \beta|--- \rangle &= \frac{\alpha}{2\sqrt{2}}(|0\rangle + |1\rangle)(|0\rangle + |1\rangle)(|0\rangle + |1\rangle) \\ &\quad + \frac{\beta}{2\sqrt{2}}(|0\rangle - |1\rangle)(|0\rangle - |1\rangle)(|0\rangle - |1\rangle). \end{aligned}$$

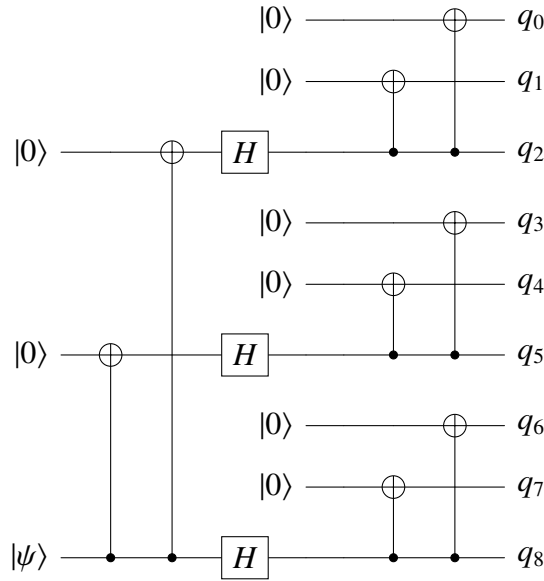
Sau đó, mỗi qubit trong 3 qubit lại được mã bằng 3 qubit như mã lật bit ($|0\rangle \rightarrow |000\rangle, |1\rangle \rightarrow |111\rangle$). Như vậy trạng thái một qubit logic $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ được mã bằng 9 qubit vật lý với trạng thái

$$\begin{aligned} \frac{\alpha}{2\sqrt{2}}(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \\ + \frac{\beta}{2\sqrt{2}}(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle). \end{aligned} \quad (6.11)$$

Lưu ý, 9 qubit được phân tách thành 3 khối với mỗi khối gồm 3 qubit. Ta kí hiệu dãy 9 qubit này là $(q_8q_7q_6)(q_5q_4q_3)(q_2q_1q_0)$.

Từ qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ và dùng thêm 8 qubit có trạng thái $|0\rangle$, mạch mã hóa cho mã Shor biến đổi trạng thái $|\psi\rangle|00\rangle|000\rangle|000\rangle$ thành trạng thái (6.11) ở trên. Để làm việc này, trước hết, ta dùng mạch mã lật pha cho $q_8q_5q_2$ (lưu ý q_8 có đầu vào là $|\psi\rangle$). Tiếp đến ta dùng mạch mã lật bit cho mỗi khối $q_8q_7q_6, q_5q_4q_3, q_2q_1q_0$. Chi tiết mạch được cho trong Hình 6.16.

Có thể nói, cách triển khai mã Shor ở mạch 6.16 là “nối” mã lật pha với mã lật bit. Mã lật pha là “mã ngoài” (thông tin mã “trải” lên các khối với q_8, q_5, q_2 đại diện



Hình 6.16: Mạch mã hóa cho mã Shor 9 qubit.

mỗi khối). Mã lật bit là “mã trong” (thông tin mã “đề” trong riêng mỗi khối). Mã Shor cũng có thể được triển khai theo cách nổi ngược lại với mã lật bit là mã ngoài và mã lật pha là mã trong (Bài tập 6.28).

Với mã Shor, việc sửa lỗi lật bit khá đơn giản. Vì thông tin mã lật bit được để riêng trong mỗi khối nên ta chỉ cần dùng mạch 6.14 để sửa lỗi lật bit (cả một phần lẫn toàn phần) cho mỗi khối (lúc này ta cũng thấy tác dụng của việc đặt lại $|0\rangle$ cho 2 qubit đo). Việc thiết kế chi tiết mạch được yêu cầu trong Bài tập 6.25.

Dĩ nhiên, nếu mỗi khối có nhiều nhất là 1 qubit bị lỗi lật bit thì mạch ở trên mới sửa được. Chẳng hạn, nếu q_8 và q_3 bị lật bit thì trạng thái (6.11) bị biến thành

$$\begin{aligned} & \frac{\alpha}{2\sqrt{2}}(|100\rangle + |011\rangle)(|001\rangle + |110\rangle)(|000\rangle + |111\rangle) \\ & + \frac{\beta}{2\sqrt{2}}(|100\rangle - |011\rangle)(|001\rangle - |110\rangle)(|000\rangle - |111\rangle). \end{aligned}$$

Với mỗi khối, mạch tính parity của 2 qubit kế bên nhau để xác định vị trí của lỗi và thực hiện việc sửa lỗi phù hợp bằng cách lật bit cho qubit tương ứng. Cụ thể, với trạng thái trên

- Khối $q_8q_7q_6$ đo được parity là 10, q_8 bị lỗi, áp dụng X cho q_8 .
- Khối $q_5q_4q_3$ đo được parity là 01, q_3 bị lỗi, áp dụng X cho q_3 .

- Khối $q_2q_1q_0$ đo được parity là 00, không có qubit nào trong khối bị lỗi.

Việc sửa lỗi lật pha với mã Shor khó hơn vì mã lật pha là mã ngoài. Ban đầu, thông tin mã pha được để ở đại diện mỗi khối là q_8, q_5, q_2 nhưng sau đó mỗi khối được áp dụng tiếp mã lật bit. Do đó, ta cần “hồi mã” cho mã lật bit ở mỗi khối bằng cách dùng mạch nghịch đảo của mạch mã lật bit (nhớ là nghịch đảo của CNOT là CNOT). Sau đó, ta đo parity và sửa lỗi như trong mạch sửa lỗi lật pha rồi “mã lại” mã lật bit cho mỗi khối. Việc thiết kế chi tiết mạch được yêu cầu trong Bài tập 6.26.

Dĩ nhiên, nếu có nhiều nhất một khối bị lỗi lật pha thì mạch ở trên mới sửa được. Chẳng hạn, nếu q_3 bị lật pha (hay qubit nào trong khối ở giữa bị lật pha cũng vậy) thì trạng thái (6.11) bị biến thành

$$\begin{aligned} & \frac{\alpha}{2\sqrt{2}}(|000\rangle + |111\rangle)(|000\rangle - |111\rangle)(|000\rangle + |111\rangle) \\ & + \frac{\beta}{2\sqrt{2}}(|000\rangle - |111\rangle)(|000\rangle + |111\rangle)(|000\rangle - |111\rangle). \end{aligned}$$

Sau khi “hồi mã” lật bit ta có trạng thái (♣)

$$\begin{aligned} & \frac{\alpha}{2\sqrt{2}}(|0\rangle + |1\rangle)|00\rangle(|0\rangle - |1\rangle)|00\rangle(|0\rangle + |1\rangle)|00\rangle \\ & + \frac{\beta}{2\sqrt{2}}(|0\rangle - |1\rangle)|00\rangle(|0\rangle + |1\rangle)|00\rangle(|0\rangle - |1\rangle)|00\rangle. \end{aligned}$$

Sau đó kết quả đo parity pha là 11 cho biết khối ở giữa bị lật pha. Để sửa lỗi, ta áp dụng cổng Z vào q_3 (hay qubit nào trong khối ở giữa cũng được) được trạng thái

$$\begin{aligned} & \frac{\alpha}{2\sqrt{2}}(|0\rangle + |1\rangle)|00\rangle(|0\rangle + |1\rangle)|00\rangle(|0\rangle + |1\rangle)|00\rangle \\ & + \frac{\beta}{2\sqrt{2}}(|0\rangle - |1\rangle)|00\rangle(|0\rangle - |1\rangle)|00\rangle(|0\rangle - |1\rangle)|00\rangle. \end{aligned}$$

Sau đó ta mã lại mã lật bit cho các khối thì được lại trạng thái (6.11).

Với mã Shor, việc sửa lỗi lật bit không ảnh hưởng đến pha và ngược lại nên bằng cách dùng mạch sửa lật pha rồi đến sửa lật bit thì ta sửa được lỗi vừa bị lật pha vừa bị lật bit. Thật ra, mã Shor có thể sửa được mọi loại lỗi (giả sử có nhiều nhất một qubit bị lỗi).⁵

Một máy tính lượng tử tích lũy lỗi “đủ chậm” để có thể sửa lỗi được gọi là có khả năng **chịu lỗi** (fault tolerant). Tùy thuộc vào mã sửa lỗi được sử dụng, “hiệu năng”

⁵chi tiết về mã sửa lỗi lượng tử, độc giả có thể xem thêm trong [4].

sửa lỗi có thể thay đổi. Đây là lĩnh vực nghiên cứu đang được quan tâm và được xem là “chén thánh” của tính toán lượng tử. Hiện giờ, máy tính lượng tử có khả năng chịu lỗi vẫn chưa xây dựng được. Thời kì hiện nay của điện toán lượng tử thường được gọi là **NISQ era** (noisy intermediate-scale quantum, NISQ).

Bài tập

6.1 Chứng minh các vector $|\phi_j\rangle, j = 0, 1, \dots, N - 1$ với

$$|\phi_j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{jk} |k\rangle, \quad \omega = e^{2\pi i \frac{1}{N}}$$

tạo thành một cơ sở trực chuẩn của \mathbb{C}^N .

6.2 Chứng minh trạng thái $|\phi_j\rangle$ ở (6.1) là trạng thái tách được, cụ thể

$$\begin{aligned} |\phi_j\rangle &= \frac{1}{\sqrt{2^n}} (|0\rangle + e^{2\pi i \frac{j}{2}} |1\rangle) (|0\rangle + e^{2\pi i \frac{j}{4}} |1\rangle) \dots (|0\rangle + e^{2\pi i \frac{j}{2^n}} |1\rangle) \\ &= \bigotimes_{l=1}^n \frac{1}{\sqrt{2}} (|0\rangle + e^{2\pi i \frac{j}{2^l}} |1\rangle). \end{aligned}$$

6.3 Thực hiện phân tích phổ tần số bằng biến đổi Fourier rời rạc (DFT) cho tín hiệu là dao động của âm thanh từ việc chơi một hợp âm La thứ (A minor triad) trên đàn piano, được lấy từ file âm thanh MP3 ở trang https://en.wikipedia.org/wiki/Minor_chord.

Gợi ý. Thực hiện tương tự Ví dụ 6.1.1. Dùng các thư viện như librosa, numpy của Python để phân tích. (Xem Jupyter Notebook ở GitHub của tài liệu.)

6.4 Tính toán từng bước để cho thấy mạch QFT 2 qubit trong Ví dụ 6.1.4 biến vector cơ sở tính toán $|j\rangle, j = 0, 1, 2, 3$ thành các vector $|\phi_j\rangle$ cho trong Ví dụ 6.1.3.

- (a) 3. (b) 4. (c) n .

6.5 Thiết kế mạch QFT 3 và 4 qubit. Tương tự Bài tập 6.4, tính toán từng bước để thấy mạch biến các vector cơ sở tính toán $|j\rangle$ thành các vector $|\phi_j\rangle$ tương ứng.

6.6 Thiết kế mạch QFT[†] với số qubit là

- (a) 3. (b) 4. (c) n .

6.7 Vẽ mạch ước lượng pha và phân tích tương tự Ví dụ 6.2.1 cho mạch U là các cổng sau (dùng các trị riêng và vector riêng tương ứng đã biết trong các bài trước).

- (a) Z . (c) H . (e) T .
 (b) Y . (d) S . (f) U ở Bài tập 4.13.

6.8 Chứng minh, trong mạch ước lượng pha 6.11, nếu ta dùng

$$|\psi\rangle = \sum_{s=0}^l \alpha_s |v_s\rangle$$

là tổ hợp tuyến tính của các vector riêng $|v_s\rangle$ của U làm trạng thái của thanh ghi vector riêng thì mạch sẽ “chọn” ước lượng pha tương ứng cho vector riêng $|v_s\rangle$ với xác suất là $|\alpha_s|^2$.

6.9 Tính

- (a) $4^5 \pmod{15}$. (b) $91^{53} \pmod{131}$. (c) $87^{38} \pmod{197}$.

6.10 Tìm chu kỳ của $a^x \pmod{N}$ với

- (a) $a = 4, N = 15$. (b) $a = 91, N = 131$. (c) $a = 87, N = 197$.

6.11 Cho các số nguyên dương a, N nguyên tố cùng nhau. Chứng minh chu kỳ r của $a^x \pmod{N}$ thỏa $0 < r < N$.

6.12 Xét M_a thao tác trên n qubit được định nghĩa bởi tác động trên các vector cơ sở tính toán $|x\rangle, x \in \{0, 1, \dots, 2^n - 1\}$ là

$$M_a |x\rangle = \begin{cases} |ax \bmod N\rangle & 0 \leq x < N, \\ |x\rangle & N \leq x < 2^n. \end{cases} \quad (6.12)$$

Chứng minh, nếu a, N nguyên tố cùng nhau thì M_a unita.

6.13 Với M_a ở Bài tập 6.12, chứng minh, với $s = 0, 1, \dots, r - 1$, ta có

$$|v_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \omega_r^{-sk} |a^k\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i \frac{sk}{r}} |a^k\rangle$$

là các vector riêng của M_a với trị riêng tương ứng là $\omega_r^s = e^{2\pi i \frac{s}{r}}$. Hơn nữa

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |v_s\rangle = |1\rangle.$$

6.14 Cho N là tích của 2 số nguyên tố p, q và $1 < a < N$ nguyên tố cùng nhau với N . Giả sử $a^x \pmod{N}$ có chu kỳ r chẵn và $a^{r/2} \not\equiv N-1 \pmod{N}$. Chứng minh

$$p = \gcd(a^{r/2} - 1, N), \quad q = \gcd(a^{r/2} + 1, N).$$

Lưu ý, p, q có “vai trò như nhau”.

6.15 Tương tự Ví dụ 6.3.4, phân tích các trường hợp có thể khi chạy thủ tục tìm chu kỳ $a^x \pmod{N}$ dùng m qubit ước lượng, giả sử thủ tục ước lượng pha luôn cho kết quả xấp xỉ m bit tốt nhất, với

(a) $a = 3, N = 7, m = 5$.

(b) $a = 2, N = 15, m = 8$.

6.16 Viết mã Python để cài đặt thuật toán Shor trong đó việc tính chu kỳ modulo ở Bước 2 có thể được cài cổ điển (sẽ thay bằng module lượng tử khi có máy tính lượng tử!). Sau đó chạy từng bước để hiểu rõ thuật toán trong các trường hợp

(a) $N = 15$.

(b) $N = 35$.

(c) $N = 209$.

(Xem Jupyter Notebook ở GitHub của tài liệu.)

6.17 Giả sử khi truyền, các bit có thể bị lỗi (lật bit) độc lập nhau với xác suất là p .⁶ Như vậy, nếu không sửa lỗi thì xác suất được bit đúng là $1 - p$. Xét mã lặp 3 bit (dùng 3 bit vật lý khi truyền cho mỗi bit logic)

- (a) Xác suất không có bit vật lý nào bị lỗi là bao nhiêu?
- (b) Xác suất chỉ có 1 bit vật lý bị lỗi là bao nhiêu?
- (c) Xác suất mã lặp 3 bit giải mã ra bit logic sai là bao nhiêu?
- (d) Ta nên dùng mã lặp 3 bit khi nào? (phân tích theo p)

6.18 Xét mã lặp 5 bit ($k = 5$).

- (a) Mã này sửa được tối đa bao nhiêu lỗi?

⁶kênh truyền có nhiều dạng này thường được gọi là kênh truyền đối xứng nhị phân (binary symmetric channel, BSC).

- (b) Lập bảng mô tả việc phát hiện và sửa lỗi mà không cần đọc giá trị các bit (tương tự Bảng 6.2).
- (c) Làm lại Bài tập 6.17 cho trường hợp này ($k = 5$).
- (d) Với $p = 0.1$, so sánh xác suất mã này sửa được lỗi với xác suất sửa được lỗi của mã lặp 3 bit.
- (e) Giữa mã lặp 3 bit và 5 bit, nên dùng mã nào hơn?

6.19 Chứng minh các trạng thái dùng trong mã lật bit

$$\alpha |000\rangle + \beta |111\rangle$$

và mã lật pha

$$\alpha |+++\rangle + \beta |--\rangle$$

là các trạng thái vướng.

6.20 Chứng minh, phép quay qubit quanh trục x một góc θ có tác động trên các ket cơ sở là

$$\begin{aligned} |0\rangle &\rightarrow i\sqrt{1-\varepsilon^2}|0\rangle + \varepsilon|1\rangle, \\ |1\rangle &\rightarrow \varepsilon|0\rangle + i\sqrt{1-\varepsilon^2}|1\rangle, \end{aligned}$$

với $\varepsilon = \sin(\frac{\theta}{2})$.

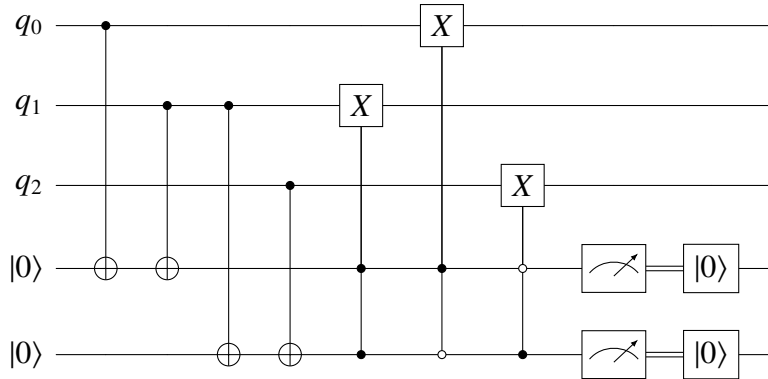
6.21 Chứng minh, phép quay qubit quanh trục z một góc θ có tác động trên các ket cơ sở là

$$\begin{aligned} |0\rangle &\rightarrow (i\sqrt{1-\varepsilon^2} + \varepsilon)|0\rangle, \\ |1\rangle &\rightarrow (i\sqrt{1-\varepsilon^2} - \varepsilon)|1\rangle, \end{aligned}$$

với $\varepsilon = \sin(\frac{\theta}{2})$. Từ đó, tác động của phép quay trên các ket $|+\rangle, |-\rangle$ là

$$\begin{aligned} |+\rangle &\rightarrow i\sqrt{1-\varepsilon^2}|+\rangle + \varepsilon|-\rangle, \\ |-\rangle &\rightarrow \varepsilon|+\rangle + i\sqrt{1-\varepsilon^2}|-\rangle. \end{aligned}$$

6.22 Từ nguyên lý đo trễ (principle of deferred measurement), mạch sửa mã lật bit 6.14 có thể được thiết kế lại là



Chứng minh mạch trên tương đương với mạch 6.14 bằng cách cho thấy chúng cho cùng đầu ra khi nhận cùng đầu vào.

6.23 Dùng nguyên lý đo trễ, thiết kế mạch tương tự ở Bài tập 6.22 cho mạch sửa mã lật pha 6.15.

6.24 Từ trạng thái mã $\alpha|000\rangle + \beta|111\rangle$, cho biết kết quả chạy mạch sửa lỗi lật bit 6.14 khi

- (i) qubit q_2 và q_1 đều bị lật toàn phần,
- (ii) cả 3 qubit đều bị lật toàn phần.

Từ đó cho biết mạch 6.14 có sửa được lỗi hay không khi có hơn 1 lỗi lật bit.

6.25 Thiết kế chi tiết mạch sửa lỗi lật bit cho mã Shor và kiểm tra kết quả khi q_8 và q_3 bị lật bit để thấy mạch sửa được lỗi.

6.26 Thiết kế chi tiết mạch sửa lỗi lật pha cho mã Shor và kiểm tra kết quả khi q_3 bị lật pha để thấy mạch sửa được lỗi.

6.27 Dùng mạch sửa lỗi lật bit (Bài tập 6.25) rồi đến mạch sửa lỗi lật pha (Bài tập 6.26) hoặc theo thứ tự ngược lại, ta có thể sửa được lỗi vừa lật bit vừa lật pha. Kiểm tra kết quả khi q_3 bị lỗi $Y = iXZ$ để thấy điều này.

6.28 Với mã Shor trình bày trong Phần 6.4.6 ta đã nối mã lật pha với lật bit, do đó lật pha là mã ngoài (outer) còn lật bit là mã trong (inner). Thiết kế và phân tích mạch nếu ta chọn thứ tự ngược lại, lật bit là mã ngoài còn lật pha là mã trong. Cách nối nào tốt hơn?

Phụ lục A

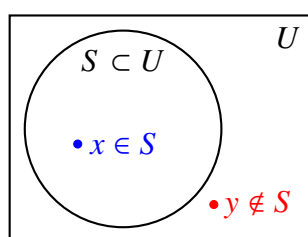
Tập hợp, tích Descartes và ánh xạ

A.1 Tập hợp

A.1.1 Khái niệm

Một **tập hợp** (set) là một bộ sưu tập các đối tượng. Nếu một đối tượng x nằm trong tập S thì ta nói x thuộc S hay x là một **phần tử** (element, member) của S , kí hiệu $x \in S$. Ta cũng kí hiệu $y \notin S$ để chỉ đối tượng y không là phần tử của S .

Thông thường, với mỗi bài toán, ta xác định tập tất cả các đối tượng liên quan U và chỉ làm việc với các **tập con** (subset) S của U , là các tập có các phần tử thuộc U , kí hiệu $S \subset U$.¹ Hình A.1 là **sơ đồ Venn** (Venn diagram) minh họa các khái niệm này.



Hình A.1: Sơ đồ Venn minh họa các khái niệm của tập hợp.

Nếu 2 tập A, B chứa cùng các phần tử, nghĩa là $A \subset B$ và $B \subset A$, thì ta nói A bằng B , kí hiệu $A = B$. Tập không có phần tử nào cả được gọi là **tập rỗng** (empty set) và được kí hiệu là \emptyset . Tập rỗng là tập con của mọi tập hợp, $\emptyset \subset A$ với mọi tập A .

¹ U thường được gọi là “phạm vi bàn luận” (domain of discourse) hay **vũ trụ** (universe).

Tập hợp có thể được cho bằng cách liệt kê các phần tử, chẳng hạn

$$\begin{aligned}\mathbb{B} &= \{0, 1\}, \\ \mathbb{N} &= \{0, 1, 2, \dots\}, \\ \mathbb{Z} &= \{\dots, -2, -1, 0, 1, 2, \dots\},\end{aligned}$$

lần lượt là tập các **bit** (binary digit, bit), tập các **số tự nhiên** (natural number) và tập các **số nguyên** (integer).

Tập hợp cũng có thể được cho bằng cách mô tả các tính chất của các phần tử, chẳng hạn

$$\begin{aligned}\mathbb{Q} &= \left\{ \frac{a}{b} : a, b \in \mathbb{Z}, b \neq 0 \right\}, \\ \mathbb{R} &= \{x : x \text{ là một điểm trên đường thẳng thực (real line)}^2\}, \\ \mathbb{C} &= \{a + bi : a, b \in \mathbb{R}, i^2 = -1\},\end{aligned}$$

lần lượt là tập các **số hữu tỉ** (rational number), tập các **số thực** (real number) và tập các **số phức** (complex number).

Ta có quan hệ giữa các tập số này như sau

$$\mathbb{B} \subsetneq \mathbb{N} \subsetneq \mathbb{Z} \subsetneq \mathbb{Q} \subsetneq \mathbb{R} \subsetneq \mathbb{C}.$$

Kí hiệu $A \subsetneq B$ chỉ A là **tập con “thực sự”** (proper subset) của B , tức là $A \subset B$ và $A \neq B$. Chẳng hạn,

$$\begin{cases} \forall x \in \mathbb{N}, x \in \mathbb{Z} \\ \exists x \in \mathbb{Z}, x \notin \mathbb{N} \text{ } (-1 \in \mathbb{Z} \text{ và } -1 \notin \mathbb{N}) \end{cases} \implies \mathbb{N} \subsetneq \mathbb{Z}.$$

Các kí hiệu logic $\forall, \exists, \implies$ có nghĩa lần lượt là “với mọi”, “tồn tại” và “suy ra”.

Ta cũng thường dùng các tập con của các tập số trên như

$$\begin{aligned}\mathbb{N}_{>0} &= \{x \in \mathbb{N} : x > 0\} = \{1, 2, \dots\}, \\ \mathbb{Z}_{<0} &= \{x \in \mathbb{Z} : x < 0\}, \\ \mathbb{R}_{\geq 0} &= \{x \in \mathbb{R} : x \geq 0\}.\end{aligned}$$

A.1.2 Các phép toán

Trong một bài toán, ta thường được cho các tập hợp đơn giản rồi dùng các phép toán trên tập hợp để xác định các tập phức tạp hơn. Các phép toán (thao tác) cơ bản trên tập hợp là

- **Giao** (intersection) của hai tập là tập các đối tượng thuộc cả hai tập đó

$$A \cap B = \{x \in U : x \in A \wedge x \in B\}.$$

- **Hợp** (union) của hai tập là tập các đối tượng thuộc ít nhất một trong hai tập đó

$$A \cup B = \{x \in U : x \in A \vee x \in B\}.$$

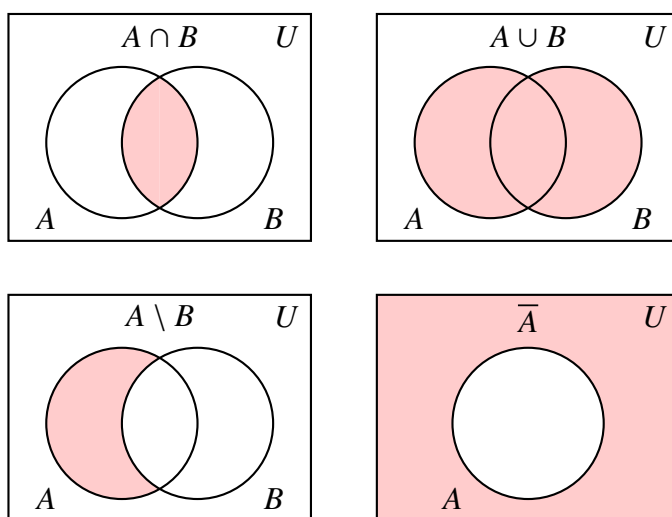
- **Hiệu** (set difference) của tập này với tập kia là tập các đối tượng thuộc tập này mà không thuộc tập kia

$$A \setminus B = \{x \in U : x \in A \wedge x \notin B\}.$$

- **Bù** (complement) của một tập là tập các đối tượng không thuộc tập đó

$$\overline{A} = A^c = U \setminus A = \{x \in U : x \notin A\}.$$

Các kí hiệu logic \wedge, \vee có nghĩa lần lượt là “và”, “hoặc”. Hình A.2 minh họa các phép toán này. Ví dụ, vì $\mathbb{Q} \subset \mathbb{R}$ nên $\mathbb{Q} \cap \mathbb{R} = \mathbb{Q}$ và $\mathbb{Q} \cup \mathbb{R} = \mathbb{R}$. Hơn nữa, $\mathbb{R} \setminus \mathbb{Q} \neq \emptyset$ thường được gọi là tập các **số vô tỉ** (irrational number). Trong ngữ cảnh chỉ làm việc với các số thực, tức $U = \mathbb{R}$, tập các số vô tỉ có thể được viết gọn là \mathbb{R}^c .



Hình A.2: Các phép toán cơ bản trên tập hợp.

A.2 Tích Descartes

Cho a, b là hai đối tượng, ta gọi (a, b) là một **cặp có thứ tự** (ordered pair) hay gọn hơn là một cặp, a được gọi là thành phần thứ nhất và b là thành phần thứ hai của cặp. Sở dĩ ta gọi là cặp có thứ tự vì thứ tự (thứ nhất, thứ hai) của các thành phần là quan trọng. Chẳng hạn ta nói cặp (a, b) bằng cặp (c, d) , kí hiệu $(a, b) = (c, d)$, nếu $a = c \wedge b = d$. Ví dụ, ta có $(1, 2) \neq (2, 1)$.

Cho hai tập A, B , **tích Descartes** (Cartesian product) của A với B là tập

$$A \times B = \{(a, b) : a \in A \wedge b \in B\}.$$

Trường hợp $A = B$, ta kí hiệu A^2 thay cho $A \times A$, tức là

$$A^2 = \{(a, b) : a \in A \wedge b \in A\}.$$

Khái niệm cặp có thể được mở rộng một cách tự nhiên, chẳng hạn, cho $n \in \mathbb{N}_{>0}$, ta kí hiệu

$$A^n = \{(a_1, a_2, \dots, a_n) : a_i \in A, \forall i = 1, 2, \dots, n\}$$

là tập tất cả các **bộ** (tuple) n thành phần với tất cả các thành phần đều thuộc A .

Tích Descartes cho phép mô tả các đối tượng “ghép” từ các đối tượng khác. Có thể tưởng tượng cặp (a, b) là đối tượng có phần đầu là a ghép với phần sau là b . Như vậy $A \times B$ là tập tất cả các đối tượng có thể ghép được với phần đầu lấy từ A và phần sau lấy từ B . Chẳng hạn, một điểm trên mặt phẳng hai chiều có thể được mô tả bằng cặp số thực (x, y) với phần đầu x là hoành độ và phần sau y là tung độ. Như vậy $\mathbb{R}^2 = \mathbb{R} \times \mathbb{R}$ là tập tất cả các điểm trên mặt phẳng. Một cách tự nhiên, \mathbb{R}^n là tập tất cả các điểm trong “không gian n chiều”.

Với tập bit $\mathbb{B} = \{0, 1\}$, ta thường kí hiệu cặp $(0, 0)$ là 00, $(0, 1)$ là 01, ... nên

$$\mathbb{B}^2 = \{00, 01, 10, 11\}$$

và thường được gọi là tập các chuỗi 2 bit. Tổng quát

$$\mathbb{B}^n = \{b_1 b_2 \dots b_n : b_i \in \mathbb{B}, \forall i = 1, \dots, n\}$$

là tập các **chuỗi n bit** (n -bit string).

Cho tập S , một tập con của $S^2 = S \times S$ mô tả một **quan hệ** (relation) hay liên kết nào đó giữa các phần tử của S . Chẳng hạn, một số nguyên $a \neq 0$ được gọi là chia hết một số nguyên b nếu a là ước của b , tức là có số nguyên q để $b = qa$. Quan hệ chia hết này là một tập con của \mathbb{Z}^2 và có thể được định nghĩa là

$$| = \{(a, b) \in \mathbb{Z}^2 : a \neq 0 \wedge \exists q \in \mathbb{Z}, b = qa\}.$$

Khi $(a, b) \in |$ ta còn kí hiệu là $a \mid b$. Chẳng hạn, 2 chia hết 10 vì $10 = 5 \times 2$ nên ta kí hiệu $2 \mid 10$ mà viết rõ là $(2, 10) \in |$.

A.3 Ảnh xạ

A.3.1 Khái niệm

Cho 2 tập X, Y , một **ảnh xạ** (map, mapping) hay **hàm** (function) f từ X vào Y , kí hiệu $f : X \rightarrow Y$, là một phép gán mỗi phần tử x của X một và chỉ một phần tử y của Y , kí hiệu $x \mapsto y = f(x)$. Ta thường gọi X là **miền xác định** (domain) và Y là **miền giá trị** (codomain) của f . Một hàm từ X vào chính nó thường được gọi là một **biến đổi** (transformation) trên X .

Cho $f, g : X \rightarrow Y$ là 2 hàm từ tập X vào tập Y , ta nói f bằng g , kí hiệu $f = g$, nếu

$$f(x) = g(x), \forall x \in X.$$

Ví dụ, bình phương là một biến đổi trên \mathbb{R} (hàm từ \mathbb{R} vào \mathbb{R})

$$x \in \mathbb{R} \mapsto x^2 \in \mathbb{R}. \quad (\text{A.1})$$

“Gấp đôi” cũng là một biến đổi trên \mathbb{R}

$$x \in \mathbb{R} \mapsto 2x \in \mathbb{R}. \quad (\text{A.2})$$

Cho chuỗi n bit $s = s_1 s_2 \cdots s_n$, đặt

$$\text{sum}(s) = \text{sum}(s_1 s_2 \cdots s_n) = \sum_{i=1}^n s_i \quad (\text{A.3})$$

là tổng các bit của s thì $\text{sum} : \mathbb{B}^n \rightarrow \mathbb{N}$ (sum là hàm từ \mathbb{B}^n vào \mathbb{N}).

Lưu ý, để được gọi là hàm thì mọi phần tử của miền xác định cần phải được gán giá trị và chỉ được gán một giá trị từ miền giá trị. Chẳng hạn, cho số thực x , phép gán x với số thực y sao cho $y^2 = x$ không phải là hàm trên \mathbb{R} vì với $x < 0$ ta không thể tìm được y , hơn nữa, với $x > 0$ ta có tới 2 giá trị y khác nhau để $y^2 = x$ (ví dụ, $(-2)^2 = 2^2 = 4$). Tuy nhiên, nếu xét x trên tập số thực không âm và chọn y cũng là số không âm thì phép gán này là một hàm hợp lệ, chính là “hàm căn”

$$x \in \mathbb{R}_{\geq 0} \mapsto \sqrt{x} \in \mathbb{R}_{\geq 0}. \quad (\text{A.4})$$

Cho $f : X \rightarrow Y$, đặt

$$G = \{(x, f(x)) : x \in X\}$$

thì G được gọi là **đồ thị** (graph) của f .³ Ta thấy $G \subset X \times Y$ nên hàm là một quan hệ. Ngược lại, không phải quan hệ nào cũng là hàm. Chẳng hạn, “quan hệ căn của” trên \mathbb{R}

$$R = \{(x, y) \in \mathbb{R}^2 : y^2 = x\}$$

không phải là hàm như đã thấy ở trên.

³trường hợp $X = Y = \mathbb{R}$ thì đồ thị G thường được vẽ trên mặt phẳng Oxy với x là hoành độ, $y = f(x)$ là tung độ như ta vẫn thấy.

A.3.2 Đơn ánh, toàn ánh và song ánh

Cho $f : X \rightarrow Y$, f được gọi là

- **đơn ánh** (injective, one-to-one) nếu $f(a) \neq f(b)$ với mọi $a \neq b \in X$.
- **toàn ánh** (surjective, onto) nếu với mọi $y \in Y$, có $x \in X$ sao cho $y = f(x)$.
- **song ánh** (bijective, one-to-one correspondence) nếu f vừa đơn ánh vừa toàn ánh, tức là với mọi $y \in Y$, có duy nhất $x \in X$ sao cho $y = f(x)$.

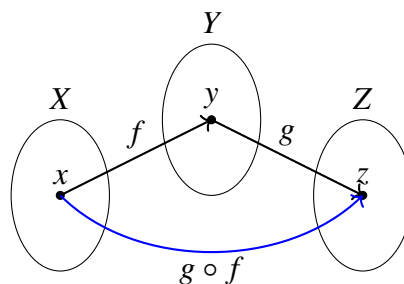
Ví dụ, hàm bình phương (A.1) không đơn ánh vì $(-2)^2 = 2^2 = 4$ và cũng không toàn ánh vì không có số nào bình phương bằng -1. Hàm sum (A.3) đơn ánh khi $n = 1$ nhưng không đơn ánh khi $n > 1$. Nếu miền giá trị là \mathbb{N} thì hàm sum không toàn ánh nhưng nếu miền giá trị là $\{0, 1, \dots, n\}$ thì hàm sum toàn ánh với mọi $n \in \mathbb{N}_{>0}$. Ngược lại, hàm gấp đôi (A.2) và hàm căn (A.4) vừa đơn ánh vừa toàn ánh nên là song ánh.

A.3.3 Hàm hợp

Cho $f : X \rightarrow Y$ và $g : Y \rightarrow Z$, hàm **hợp** (composition) của g với f , kí hiệu $g \circ f$, là hàm từ X vào Z được định nghĩa là

$$x \in X \mapsto (g \circ f)(x) = g(f(x)) \in Z.$$

Một cách hình ảnh, cho $x \in X$, để tìm $z = (g \circ f)(x)$, ta tìm $y = f(x) \in Y$ và sau đó tìm $z = g(y) \in Z$ như minh họa ở Hình A.3.



Hình A.3: Minh họa hàm hợp.

Ví dụ, với f, g lần lượt là hàm bình phương (A.1) và hàm gấp đôi (A.2) thì

$$(g \circ f)(x) = g(f(x)) = g(x^2) = 2x^2,$$

còn

$$(f \circ g)(x) = f(g(x)) = f(2x) = 4x^2.$$

A.3.4 Hàm ngược

Cho $f : X \rightarrow Y$ là một song ánh. Khi đó, với mỗi $y \in Y$ có duy nhất một $x \in X$ để $y = f(x)$ nên phép gán y với x này là một hàm hợp lệ từ Y vào X . Ta gọi hàm này là **hàm ngược** (inverse function) của f , kí hiệu f^{-1} . Nhận xét

$$f^{-1}(f(x)) = x, \forall x \in X$$

nên f^{-1} “undo” f . Nếu đặt $\text{id}_X : X \rightarrow X$ là **hàm đơn vị** (identity function), được định nghĩa là $\text{id}_X(x) = x, \forall x \in X$ thì

$$f^{-1} \circ f = \text{id}_X \wedge f \circ f^{-1} = \text{id}_Y.$$

Ví dụ, hàm gấp đôi (A.2) là song ánh trên \mathbb{R} nên có hàm ngược trên \mathbb{R} là “hàm chia đôi”

$$x \in \mathbb{R} \mapsto \frac{x}{2} \in \mathbb{R}.$$

Hàm bình phương (A.1) không là song ánh trên \mathbb{R} nên không có hàm ngược trên \mathbb{R} . Tuy nhiên, nếu “giới hạn” hàm bình phương trên $\mathbb{R}_{\geq 0}$ (nghĩa là đặt miền xác định và miền giá trị đều là $\mathbb{R}_{\geq 0}$) thì hàm bình phương là song ánh với hàm ngược chính là hàm căn (A.4).

Rõ ràng $f : \mathbb{B}^n \rightarrow \mathbb{B}^m$ không thể là song ánh nếu $n \neq m$. Nếu $n < m$ thì f không thể toàn ánh còn nếu $n > m$ thì f không thể đơn ánh. Nếu $f : \mathbb{B}^n \rightarrow \mathbb{B}^n$ là song ánh thì mỗi $x \in \mathbb{B}^n$ tương ứng với một và chỉ một $y = f(x) \in \mathbb{B}^n$ nên có thể nói, f là một **hoán vị** (permutation) trên \mathbb{B}^n , tức là một cách sắp xếp các phần tử của \mathbb{B}^n .

Ví dụ, trên $\mathbb{B}^1 = \mathbb{B}$ chỉ có 2 hoán vị (2 song ánh) là ánh xạ đơn vị $\text{id}_{\mathbb{B}}$ và ánh xạ “đảo” biến 0 thành 1 và 1 thành 0. Trên \mathbb{B}^2 có 24 hoán vị khác nhau mà vài trường hợp được cho ở Bảng A.1. Nếu xem 00, 01, 10, 11 là thứ tự liệt kê “tự nhiên” các phần tử của \mathbb{B}^2 thì: f_1 là id giữ nguyên thứ tự, f_2 đổi chỗ 2 phần tử đầu, f_3 “dịch phải” và f_4 “dịch trái” một vị trí.

$x \in \mathbb{B}^2$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
00	00	01	11	01
01	01	00	00	10
10	10	10	01	11
11	11	11	10	00

Bảng A.1: Một vài hoán vị trên \mathbb{B}^2 .

A.3.5 Lực lượng của tập hợp

Các khái niệm đơn ánh, toàn ánh, song ánh cũng liên quan mật thiết đến việc **đếm** (counting) hay **liệt kê** (enumeration) vì khi đếm (hay liệt kê) các phần tử của một tập S là ta đang gán tương ứng các phần tử này với các số $1, 2, 3, \dots$. Cụ thể, nếu có một song ánh từ tập A đến tập B , ta nói A, B cùng **lực lượng** (cardinality), kí hiệu $|A| = |B|$; ngược lại, ta nói $|A| \neq |B|$. Nếu có một đơn ánh từ A đến B , ta nói A có lực lượng không quá B , kí hiệu $|A| \leq |B|$. Ta cũng kí hiệu $|A| < |B|$ để chỉ $|A| \leq |B|$ và $|A| \neq |B|$.

Cho tập S , ta cũng nói

- S **hữu hạn** (finite) nếu S cùng lực lượng với tập $\{1, 2, \dots, n\}$ với $n \in \mathbb{N}$ nào đó, ta cũng kí hiệu $|S| = n$. Tập \emptyset cũng được xem là hữu hạn với $|\emptyset| = 0$.
- S **vô hạn** (infinite) nếu S không hữu hạn, kí hiệu $|S| = \infty$.
- S vô hạn **đếm được** (countable) nếu $|S| = |\mathbb{N}|$.
- S vô hạn **không đếm được** (uncountable) nếu $|S| > |\mathbb{N}|$ (tức là $|\mathbb{N}| < |S|$).

Ví dụ: tập \mathbb{B}^n là hữu hạn với $|\mathbb{B}^n| = 2^n$, các tập $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$ đều vô hạn đếm được, các tập \mathbb{R}, \mathbb{C} hay $[0, 1] = \{x \in \mathbb{R} : 0 \leq x \leq 1\}$ đều vô hạn không đếm được và

$$|\mathbb{C}| = |[0, 1]| = |\mathbb{R}|.$$

Phụ lục B

Xác suất

B.1 Xác suất

Lý thuyết xác suất (probability theory) là ngành Toán học giúp định lượng, tính toán và suy diễn trên các hiện tượng **ngẫu nhiên** (random) và/hoặc **không chắc chắn** (uncertain).

B.1.1 Không gian mẫu và biến cố

Thí nghiệm ngẫu nhiên (random experiment) là các quá trình/hoạt động/thử nghiệm/công việc/thao tác không biết chắc chắn **kết quả** (outcome) nhưng xác định được tập tất cả các kết quả có thể. Tập tất cả các kết quả có thể này được gọi là **không gian mẫu** (sample space) của thí nghiệm, kí hiệu là Ω (omega).

Nếu việc xảy ra hay không của một tình huống E được xác định hoàn toàn khi biết kết quả của thí nghiệm T thì E được gọi là **biến cố** (event) **liên quan** đến T . Biến cố được xác định bởi các **kết quả thuận lợi** cho nó

$$E = \{\omega \in \Omega : \omega \text{ làm cho } E \text{ xảy ra}\} \subset \Omega.$$

Như vậy, “lý thuyết biến cố” được hình thức hóa bằng “lý thuyết tập hợp”. Xét thí nghiệm T với không gian mẫu Ω và các biến cố $E, F \subset \Omega$, ta nói

- $\{\omega\} \equiv \omega \in \Omega$: **biến cố sơ cấp** (elementary event),
- Ω : **biến cố chắc chắn** (certain event),
- \emptyset : **biến cố không thể** (impossible event),
- $E^c = \Omega \setminus E$: biến cố **đối** (complement) của E , biến cố “ E không xảy ra”,

- $E \cup F$: biến cố E **hoặc** (or) F , biến cố “ E xảy ra hoặc F xảy ra”,
- $E \cap F$: biến cố E **và** (and) F , biến cố “ E xảy ra và F xảy ra”,
- $E \setminus F$: biến cố E **không** (not) F , biến cố “ E xảy ra nhưng F không xảy ra”,
- $E \subset F$: E **kéo theo** (imply) F , E xảy ra thì F xảy ra,
- $E = F$: E **là** (is) F , E và F cùng xảy ra hoặc cùng không xảy ra,
- $E \cap F = \emptyset$: E, F **rời nhau** (disjoint) hay **xung khắc** (mutually exclusive), E và F không thể đồng thời xảy ra.

B.1.2 Độ đo xác suất

Xét thí nghiệm T với không gian mẫu Ω , một hàm P gán mỗi biến cố $E \subset \Omega$ với số thực $P(E)$ được gọi là một **độ đo xác suất** (probability measure) trên Ω nếu P thỏa mãn 3 tiên đề

1. Với mọi biến cố $E \subset \Omega$, $0 \leq P(E) \leq 1$.
2. Với mọi dãy biến cố E_1, E_2, \dots đôi một *xung khắc* ($E_i \cap E_j = \emptyset, \forall i \neq j$):

$$P\left(\bigcup_{i=1}^{\infty} E_i\right) = \sum_{i=1}^{\infty} P(E_i),$$

tức là $P(E_1 \cup E_2 \cup \dots) = P(E_1) + P(E_2) + \dots$

3. $P(\Omega) = 1$.

$P(E)$ được gọi là **xác suất** (probability) của E và là số đo khả năng xảy ra của biến cố E khi *không biết kết quả* của thí nghiệm T .

Từ 3 tiên đề, ta suy ra được các tính chất cơ bản sau của xác suất

1. $P(\overline{E}) = 1 - P(E)$
2. $P(\emptyset) = 0$
3. Nếu $E_1 \subset E_2$ thì $P(E_1) \leq P(E_2)$ và $P(E_2 \setminus E_1) = P(E_2) - P(E_1)$
4. Nếu $E_1 \cap E_2 = \emptyset$ thì $P(E_1 \cup E_2) = P(E_1) + P(E_2)$
5. $P(E_1 \cup E_2) = P(E_1) + P(E_2) - P(E_1 \cap E_2)$ (**addition law of probability**)
6. $P(E_1 \cup E_2 \cup E_3) = P(E_1) + P(E_2) + P(E_3) - P(E_1 \cap E_2) - P(E_1 \cap E_3) - P(E_2 \cap E_3) + P(E_1 \cap E_2 \cap E_3)$
7. $P(\bigcup_{i=1}^{\infty} E_i) \leq \sum_{i=1}^{\infty} P(E_i)$ (**union bound**)

B.1.3 Xác suất rời rạc

Khi không gian mẫu hữu hạn hoặc vô hạn đếm được, $\Omega = \{\omega_1, \omega_2, \dots\}$, ta có **mô hình xác suất rời rạc** (discrete probability model). Trong đó độ đo xác suất được xác định bởi xác suất của các biến cố sơ cấp $p_i = P(\omega_i)$

- $p_i \geq 0, i = 1, 2, \dots$ với $\sum_i p_i = 1$,
- $P(E) = \sum_{\omega_i \in E} p_i, \forall E \subset \Omega$.

Khi không gian mẫu hữu hạn và các **kết quả đồng khả năng** (equiprobable outcomes), $\Omega = \{\omega_1, \dots, \omega_n\}$, ta có **mô hình xác suất đơn giản** (simple/classical probability model)

- $p_i = \frac{1}{n}, i = 1, \dots, n$,
- $P(E) = \frac{|E|}{|\Omega|}, \forall E \subset \Omega$, ($|X|$ là số lượng phần tử của tập X)
- Xác suất là tỉ lệ và việc tính xác suất được đưa về việc **đếm** (counting).

Ví dụ B.1.1. Hoạt động “tung 2 xúc xắc đồng chất” là một thí nghiệm ngẫu nhiên vì ta không biết chắc kết quả được mặt nào. Kí hiệu 6 mặt của xúc xắc bằng các số tương ứng và đặt

$$S = \{1, 2, 3, 4, 5, 6\}.$$

Gọi a, b lần lượt là mặt ra của xúc xắc 1 và 2, ta có không gian mẫu (tập tất cả các kết quả có thể) là

$$\Omega = \{(a, b) : a, b \in S\} = S^2.$$

Khi biết kết quả tung là (a, b) , ta có thể xác định được các tình huống

- E : tổng 2 mặt bằng 10 ($a + b \stackrel{?}{=} 10$)
- F : tổng 2 mặt bằng 11 ($a + b \stackrel{?}{=} 11$)

là có xảy ra hay không nên E, F là các biến cố liên quan đến thí nghiệm. E, F có thể được xác định bằng tập các kết quả thuận lợi như sau

$$E = \{(4, 6), (5, 5), (6, 4)\}, \quad F = \{(5, 6), (6, 5)\}.$$

Vì các xúc xắc đồng chất nên mỗi mặt của mỗi xúc xắc đều có cùng khả năng ra. Hơn nữa, 2 xúc xắc “không liên quan nhau” nên mỗi kết quả (a, b) của Ω đều có cùng khả năng. Như vậy, dùng mô hình xác suất đơn giản ta có

$$P(E) = \frac{|E|}{|\Omega|} = \frac{3}{36}, \quad P(F) = \frac{|F|}{|\Omega|} = \frac{2}{36}.$$

Xác suất tổng 2 mặt bằng 11 nhỏ hơn xác suất tổng 2 mặt bằng 10. □

Ví dụ B.1.2. Bài toán sinh nhật (birthday problem): tính xác suất p của biến cố có ít nhất 2 người cùng sinh nhật (cùng ngày và tháng sinh) trong nhóm k người? Giả sử ngày sinh của mỗi người là một ngày ngẫu nhiên trong một năm gồm 365 ngày và “không liên quan nhau”.

Giải. Không mất tính tổng quát, ta có thể gọi tập tất cả các ngày trong năm là

$$\mathcal{Y} = \{1, 2, \dots, 365\}.$$

Không gian mẫu

$$\Omega = \{(d_1, d_2, \dots, d_k) : d_i \in \mathcal{Y}, i = 1, \dots, k\} = \mathcal{Y}^k.$$

Đặt các biến cố

- A : “có ít nhất 2 người cùng sinh nhật”,
- B : “không có người nào cùng sinh nhật”.

Ta thấy $A = \overline{B}$ và $B = \{\text{các chỉnh hợp chọn } k \text{ của } \mathcal{Y}\}$ với

$$|B| = \frac{365!}{(365 - k)!}.$$

Ta có $|\Omega| = 365^k$ và dùng mô hình xác suất đơn giản ta có

$$p = P(A) = 1 - P(B) = 1 - \frac{|B|}{|\Omega|} = 1 - \frac{365!}{(365 - k)!365^k}.$$

Hình B.1 minh họa xác suất p tính theo k . Lưu ý, chỉ cần $k = 23$ thì p đã lớn hơn 50%. Đây là một kết quả đáng ngạc nhiên với trực giác của nhiều người (nhiều người nghĩ k phải lớn hơn nữa) nên thường được gọi là một “nghịch lý”.¹ \square

B.2 Xác suất có điều kiện

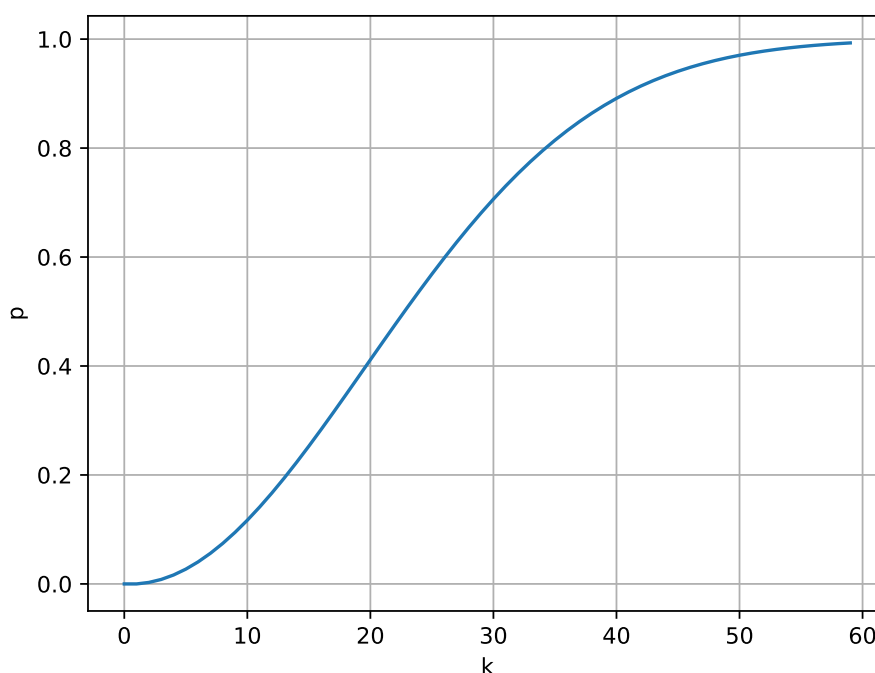
B.2.1 Định nghĩa

Vì xác suất đánh giá khả năng xảy ra của các biến cố nên ta cần điều chỉnh hay *cập nhật xác suất* của các biến cố liên quan đến thí nghiệm T khi có thêm thông tin về T mà thường là khi biết (các) biến cố nào đó đã xảy ra.

Xác suất của biến cố A khi biết biến cố B đã xảy ra được gọi là **xác suất có điều kiện** (conditional probability) của A khi biết B xảy ra, kí hiệu là $P(A|B)$ và được tính bằng định nghĩa

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad (\text{với } P(B) > 0).$$

¹https://en.wikipedia.org/wiki/Birthday_problem.



Hình B.1: Xác suất p tính theo k trong bài toán sinh nhật.

- $A \cap B$ là “ A trong B ”,
- Chia $P(A \cap B)$ cho $P(B)$ giúp chuẩn hóa xác suất,
- $P(.|B)$ có thể hiểu là xác suất “tính trong không gian mẫu mới” B và là một độ đo xác suất hợp lệ.

B.2.2 Công thức nhân, toàn phần và công thức Bayes

Từ định nghĩa của xác suất có điều kiện ta có **công thức nhân xác suất** (multiplication rule)

$$P(A \cap B) = P(B)P(A|B) \text{ (với } P(B) > 0\text{)}.$$

Tổng quát, cho n biến cố A_1, \dots, A_n với $P(A_1 A_2 \dots A_n) > 0$, ta có

$$P(A_1 A_2 \dots A_n) = P(A_1)P(A_2|A_1)P(A_3|A_1 A_2) \dots P(A_n|A_1 A_2 \dots A_{n-1}).$$

Lưu ý, khi có nhiều biến cố, ta thường viết gọn $P(A_1 \cap A_2 \cap \dots \cap A_n)$ là

$$P(A_1, A_2, \dots, A_n) \text{ hay } P(A_1 A_2 \dots A_n).$$

Cho các biến cố B_1, B_2, \dots, B_n là một **họ đầy đủ** của Ω , nghĩa là

1. $P(B_i) > 0, \forall i = 1, \dots, n$,

2. $B_i \cap B_j = \emptyset, \forall i \neq j,$
3. $\Omega = B_1 \cup B_2 \cup \dots \cup B_n.$

Khi đó, với mọi biến cố A , **công thức xác suất toàn phần** (law of total probability) cho

$$P(A) = \sum_{i=1}^n P(A \cap B_i) = \sum_{i=1}^n P(B_i)P(A|B_i).$$

Đặc biệt, với mọi biến cố B với $P(B) > 0$, ta có

$$P(A) = P(B)P(A|B) + P(B^c)P(A|B^c).$$

Cho B_1, \dots, B_n là một **họ đầy đủ** của Ω và biến cố A với $P(A) > 0$, **công thức Bayes** (Bayes' theorem, Bayes's rule) cho

$$P(B_i|A) = \frac{P(B_i)P(A|B_i)}{P(A)} = \frac{P(B_i)P(A|B_i)}{\sum_{j=1}^n P(B_j)P(A|B_j)}, \quad i = 1, \dots, n.$$

- $P(B_i)$: **xác suất tiên nghiệm** (prior probability) của B_i ,
- $P(B_i|A)$: **xác suất hậu nghiệm** (posterior probability) của B_i khi biết A ,
- $P(A|B_i)$: **xác suất hợp lý** (likelihood) của A theo B_i .

B.2.3 Các biến cố độc lập

Hai biến cố $\{A, B\}$ được gọi là **độc lập** (independent, statistically independent, stochastically independent) nhau nếu

$$P(A \cap B) = P(A)P(B).$$

Một cách tương đương: $P(A|B) = P(A)$ hay $P(B|A) = P(B)$.

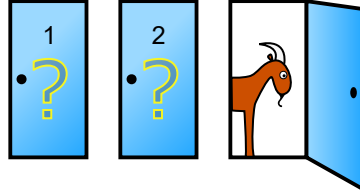
Họ các biến cố $\{A_1, A_2, \dots\}$ được gọi là **độc lập** nếu với mọi tập con khác rỗng và hữu hạn $\{B_1, B_2, \dots, B_k\}$ của họ ta có

$$P(B_1 B_2 \dots B_k) = P(B_1)P(B_2) \dots P(B_k).$$

Nếu một thí nghiệm T được thực hiện lặp lại nhiều lần *một cách độc lập* và A_i là biến cố “*liên quan đến lần thực hiện thứ i* ” thì ta có

$$P(A_1 A_2 \dots A_n) = P(A_1)P(A_2) \dots P(A_n).$$

Ví dụ B.2.1. Monty Hall problem: trong một trò chơi truyền hình, trên sân khấu có 3 cửa #1, #2, #3, có 1 cửa chứa xe do ban tổ chức đặt ngẫu nhiên và 2 cửa còn lại chứa dê. Người chơi chọn 1 cửa (chẳng hạn cửa #1). Người dẫn chương trình biết cửa nào có gì. Người dẫn chọn và mở cửa có dê trong 2 cửa còn lại (chẳng hạn cửa #3). Người dẫn hỏi người chơi có muốn đổi lựa chọn không (vẫn giữ cửa #1 hay chọn cửa #2). Người chơi nên giữ hay đổi (để khả năng được xe cao hơn)?



(https://en.wikipedia.org/wiki/Monty_Hall_problem)

Giải. Do “tính đối xứng” nên ta có thể giả sử kịch bản như mô tả: người chơi chọn cửa #1, người dẫn mở cửa #3. Các trường hợp khác cho ra cùng kết quả.

Đặt A_i là biến cố “xe được đặt ở cửa # i ” và B_j là biến cố “người dẫn mở cửa # j ” ($i, j = 1, 2, 3$). Từ bài toán và kịch bản đã cho (người chơi chọn cửa #1), ta có

- $P(A_1) = P(A_2) = P(A_3) = \frac{1}{3}$.
- $P(B_3|A_1) = \frac{1}{2}$ (người chơi đã chọn cửa #1, xe cũng được đặt ở cửa #1 nên người dẫn có thể mở 1 trong 2 cửa #2, #3).
- $P(B_3|A_2) = 1$ (người chơi đã chọn cửa #1, xe được đặt ở cửa #2 nên người dẫn chỉ có thể mở cửa #3).
- $P(B_3|A_3) = 0$ (xe được đặt ở cửa #3 nên người dẫn không được mở cửa #3).

Công thức Bayes cho xác suất người chơi được xe khi không đổi cửa và đổi cửa là

$$\begin{aligned}
 P(A_1|B_3) &= \frac{P(A_1)P(B_3|A_1)}{P(A_1)P(B_3|A_1) + P(A_2)P(B_3|A_2) + P(A_3)P(B_3|A_3)} \\
 &= \frac{\frac{1}{3} \frac{1}{2}}{\frac{1}{3} \frac{1}{2} + \frac{1}{3} 1 + \frac{1}{3} 0} = \frac{1}{3}, \\
 P(A_2|B_3) &= \frac{P(A_2)P(B_3|A_2)}{P(A_1)P(B_3|A_1) + P(A_2)P(B_3|A_2) + P(A_3)P(B_3|A_3)} \\
 &= \frac{\frac{1}{3} 1}{\frac{1}{3} \frac{1}{2} + \frac{1}{3} 1 + \frac{1}{3} 0} = \frac{2}{3}.
 \end{aligned}$$

Vậy người chơi nên chọn đổi cửa! Các tính toán cũng có thể được trực quan bằng sơ đồ cây (tree diagram) sau.

	Car location:	Host opens:	Total probability:	Stay:	Switch:
1/3	Door 1	1/2 Door 2	1/6	Car	Goat
		1/2 Door 3	1/6	Car	Goat
	Door 2	1 Door 3	1/3	Goat	Car
1/3	Door 3	1 Door 2	1/3	Goat	Car

(https://en.wikipedia.org/wiki/Monty_Hall_problem#Conditional_probability_by_direct_calculation)

□

B.3 Biến ngẫu nhiên rời rạc

B.3.1 Định nghĩa

Nếu giá trị của một đại lượng X được xác định hoàn toàn khi biết kết quả ω của thí nghiệm T thì X được gọi là một **biến ngẫu nhiên** (random variable) liên quan đến T . Trước khi biết kết quả, ta chỉ biết X có thể nhận một giá trị nào đó trong tập S . Sau khi biết kết quả, ta biết X nhận một giá trị cụ thể thuộc S . Như vậy

$$X : \omega \in \Omega \mapsto X(\omega) \in S.$$

X là một hàm từ không gian mẫu Ω vào **tập giá trị** (range) hay **không gian trạng thái** (state space) S . Thông thường, S là tập con của \mathbb{R} (hoặc \mathbb{R}^d).

Biến ngẫu nhiên là phương tiện hay được dùng để mô tả các biến cố. Xét biến ngẫu nhiên X liên quan đến thí nghiệm T có không gian mẫu là Ω . Cho $C \subset S$, ta kí hiệu biến cố “ X nhận giá trị trong C ” là

$$(X \in C) = \{\omega \in \Omega : X(\omega) \in C\}.$$

X được gọi là **biến ngẫu nhiên rời rạc** (discrete random variable) nếu tập giá trị của nó là hữu hạn hoặc vô hạn đếm được.

Cho X là biến ngẫu nhiên (rời rạc),² **hàm xác suất** (probability function, proba-

²ngoài biến ngẫu nhiên rời rạc còn có các loại biến ngẫu nhiên khác. Tài liệu này chỉ làm việc với biến ngẫu nhiên rời rạc nên ta nói gọn là biến ngẫu nhiên.

bility mass function) của X là hàm $f : S \rightarrow \mathbb{R}$, được xác định bởi

$$f(x) = P(X = x), x \in S.$$

- Hàm xác suất f cho biết khả năng X nhận một giá trị cụ thể.
- Hàm xác suất có tính chất: $f(x) \geq 0, \forall x \in S$ và $\sum_{x \in S} f(x) = 1$.

Hàm xác suất xác định **phân phối** (distribution) của X

$$P(X \in C) = \sum_{x \in C} f(x), \quad C \subset S.$$

B.3.2 Kỳ vọng và phương sai

Cho biến ngẫu nhiên X có tập giá trị $S \subset \mathbb{R}$ và hàm xác suất f , **kỳ vọng** (mean) của X là số thực được tính bởi

$$\mu = E(X) = \sum_x xP(X = x) = \sum_x xf(x).$$

Như vậy, kỳ vọng của X là giá trị trung bình của các giá trị mà X có thể nhận với trọng số là xác suất để X nhận các giá trị tương ứng đó. Kỳ vọng phản ánh **trọng tâm** của phân phối của X .

Cho biến ngẫu nhiên $X : \Omega \rightarrow S_X$ và hàm $r : S_X \rightarrow S_Y$, ta nói $Y = r \circ X : \Omega \rightarrow S_Y$ là biến ngẫu nhiên **biến đổi** (transformation) từ X qua hàm r , kí hiệu $Y = r(X)$. Nếu $S_Y \subset \mathbb{R}$, ta có

$$E(Y) = E(r(X)) = \sum_x r(x)f_X(x).$$

Cho biến ngẫu nhiên X có tập giá trị $S \subset \mathbb{R}$, hàm xác suất f và kỳ vọng $\mu = E(X)$, **phương sai** (variance) của X là số thực được tính bởi

$$\sigma^2 = \text{Var}(X) = E((X - \mu)^2) = \sum_x (x - \mu)^2 f(x).$$

Ta cũng nói $\sigma = \sqrt{\sigma^2} = \sqrt{\text{Var}(X)}$ là **độ lệch chuẩn** (standard deviation) của X . Phương sai (và độ lệch chuẩn) phản ánh sự **phân tán** của phân phối của X .

Mệnh đề B.3.1. $\text{Var}(X) = E(X^2) - (E(X))^2$. △

Mệnh đề B.3.2. Cho X_1, X_2, \dots, X_n là các biến ngẫu nhiên (có kỳ vọng), ta có

$$E\left(\sum_{i=1}^n X_i\right) = \sum_{i=1}^n E(X_i) \quad (\text{linearity of expectation}).$$

△

Ví dụ B.3.1. (tiếp Ví dụ B.1.1) Trong thí nghiệm tung 2 xúc xắc đồng chất, gọi X là “chênh lệch giữa 2 mặt ra”, tức là

$$X = |a - b|$$

với a, b lần lượt là mặt ra của xúc xắc 1 và 2. Ta có X là biến ngẫu nhiên (rời rạc) với tập giá trị

$$S = \{0, 1, 2, 3, 4, 5\}.$$

Hàm xác suất f của X được cho như bảng sau

x	0	1	2	3	4	5
$f(x)$	$\frac{6}{36}$	$\frac{10}{36}$	$\frac{8}{36}$	$\frac{6}{36}$	$\frac{4}{36}$	$\frac{2}{36}$

Chẳng hạn

$$f(0) = P(X = 0) = \frac{| \{(0, 0), (1, 1), \dots, (6, 6) \} |}{|\Omega|} = \frac{6}{36}.$$

Từ đây ta có

$$P(X \geq 3) = P(X = 3) + P(X = 4) + P(X = 5) = f(3) + f(4) + f(5) = \frac{12}{36}.$$

Ta cũng có

$$E(X) = \sum_{x \in S} x f(x) = 0 \times \frac{6}{36} + 1 \times \frac{10}{36} + \dots + 5 \times \frac{2}{36} = \frac{35}{18} \approx 1.94,$$

$$E(X^2) = \sum_{x \in S} x^2 f(x) = 0^2 \times \frac{6}{36} + 1^2 \times \frac{10}{36} + \dots + 5^2 \times \frac{2}{36} = \frac{35}{6} \approx 5.83,$$

$$\text{Var}(X) = E(X^2) - (E(X))^2 \approx 2.0524,$$

$$\sigma = \sqrt{\text{Var}(X)} \approx 1.43.$$

□

B.3.3 Phân phối đồng thời

Cho các biến ngẫu nhiên X, Y (cùng liên quan đến thí nghiệm T) với tập giá trị tương ứng là S_X, S_Y , **hàm xác suất đồng thời** (joint probability function) của X, Y là hàm $f_{XY} : S_X \times S_Y \rightarrow \mathbb{R}$, được xác định bởi

$$f_{XY}(x, y) = P(X = x, Y = y), \quad x \in S_X, y \in S_Y.$$

Kí hiệu $P(X = x, Y = y)$ là viết gọn của $P((X = x) \cap (Y = y))$.

Ta có: $f_{XY}(x, y) \geq 0, \forall x \in S_X, y \in S_Y$ và $\sum_{(x,y) \in S_X \times S_Y} f(x, y) = 1$.

Hàm xác suất đồng thời xác định **phân phối đồng thời** (joint distribution) của X, Y

$$P((X, Y) \in C) = \sum_{(x,y) \in C} f_{XY}(x, y), \quad C \subset S_X \times S_Y.$$

Từ hàm xác suất đồng thời f_{XY} , theo công thức xác suất toàn phần, ta có được hàm xác suất của riêng X, Y , còn gọi là **hàm xác suất lề** (marginal probability function)

$$f_X(x) = P(X = x) = \sum_{y \in S_Y} f_{XY}(x, y), \quad x \in S_X,$$

$$f_Y(y) = P(Y = y) = \sum_{x \in S_X} f_{XY}(x, y), \quad y \in S_Y.$$

Hơn nữa, ta có thể định nghĩa **hàm xác suất có điều kiện** (conditional probability function) của X khi biết Y là

$$f_{X|Y}(x|y) = P(X = x|Y = y) = \frac{P(X = x, Y = y)}{P(Y = y)} = \frac{f_{XY}(x, y)}{f_Y(y)}.$$

B.3.4 Các biến ngẫu nhiên độc lập

Hai biến ngẫu nhiên X, Y có tập giá trị lần lượt là S_X, S_Y được gọi là **độc lập** (independent) nếu với mọi $A \subset S_A, B \subset S_B$ ta có

$$P(X \in A, Y \in B) = P(X \in A)P(Y \in B).$$

Mệnh đề B.3.3. Hai biến ngẫu nhiên X, Y độc lập khi và chỉ khi

$$f_{XY}(x, y) = P(X = x, Y = y) = P(X = x)P(Y = y) = f_X(x)f_Y(y)$$

với mọi $x \in S_X, y \in S_Y$. Khi đó, ta cũng có

$$f_{X|Y}(x, y) = f_X(x) \text{ và } f_{Y|X}(y, x) = f_Y(y).$$

△

B.3.5 Hiệp phương sai và hệ số tương quan

Cho các biến ngẫu nhiên X, Y và hàm $r : S_X \times S_Y \rightarrow S_Z$, ta nói $Z : \Omega \rightarrow S_Z$ là biến ngẫu nhiên biến đổi từ X, Y qua hàm r , kí hiệu $Z = r(X, Y)$, nếu Z được xác định bởi

$$Z(\omega) = r(X(\omega), Y(\omega)), \quad \omega \in \Omega.$$

Nếu $S_Z \subset \mathbb{R}$, ta có

$$E(Z) = E(r(X, Y)) = \sum_{(x,y)} r(x, y) f_{XY}(x, y).$$

Cho hai biến ngẫu nhiên X, Y có các tập giá trị là tập con của \mathbb{R} , **hiệp phương sai** (covariance) của X, Y là

$$\text{Cov}(X, Y) = E((X - E(X))(Y - E(Y))) = E(XY) - E(X)E(Y).$$

Hiệp phương sai phản ánh sự “biến thiên cùng nhau” (“cùng tăng hoặc cùng giảm”) của hai biến ngẫu nhiên. Giá trị

$$\rho_{XY} = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y}$$

được gọi là **hệ số tương quan** (correlation coefficient) của X, Y . Nhận xét

1. $-1 \leq \rho_{XY} \leq 1$,
2. $\rho_{XY} = 1$ khi và chỉ khi $Y = aX + b$ với $a > 0$,
3. $\rho_{XY} = -1$ khi và chỉ khi $Y = aX + b$ với $a < 0$,
4. Nếu $U = aX + b$, $V = cY + d$, $a, c > 0$ thì $\rho_{UV} = \rho_{XY}$.

Như vậy, hệ số tương quan phản ánh “tương quan tuyến tính” giữa hai biến ngẫu nhiên. Ta cũng nói X, Y **không tương quan** (uncorrelated) nếu $\rho_{XY} = 0$ ($\Leftrightarrow \text{Cov}(X, Y) = 0$), ngược lại là **có tương quan** (correlated).

Mệnh đề B.3.4. Cho X, Y là hai biến ngẫu nhiên độc lập, ta có

1. $E(XY) = E(X)E(Y)$,
2. $\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y)$.
3. $\text{Cov}(X, Y) = 0$.

△

B.3.6 Hàm đặc trưng của biến cố

Cho biến cố A liên quan đến thí nghiệm T với không gian mẫu Ω , ta gọi **hàm đặc trưng** (characteristic function, indicator function) của A là hàm $\mathbb{I}_A : \Omega \rightarrow \mathbb{R}$ được xác định bởi

$$\mathbb{I}_A(\omega) = \begin{cases} 1 & \text{nếu } \omega \in A, \\ 0 & \text{nếu } \omega \notin A. \end{cases}$$

Ta có

- \mathbb{I}_A là biến ngẫu nhiên có tập giá trị $S = \{0, 1\}$,
- $P(\mathbb{I}_A = 1) = P(A)$, $P(\mathbb{I}_A = 0) = 1 - P(A)$,
- $E(\mathbb{I}_A) = P(A)$, $\text{Var}(\mathbb{I}_A) = P(A)(1 - P(A))$,
- $\text{Cov}(\mathbb{I}_A, \mathbb{I}_B) = P(A \cap B) - P(A)P(B)$ nên $\mathbb{I}_A, \mathbb{I}_B$ không tương quan khi và chỉ khi A, B độc lập.

Hàm đặc trưng giúp khảo sát biến cổ như là một biến ngẫu nhiên.

B.3.7 Các phân phối rời rạc thông dụng

Trong thực tế, ta hay gặp một số dạng phân phối sau cho biến ngẫu nhiên X

1. X được gọi là có **phân phối đều** (uniform distribution) trên tập hữu hạn $S = \{x_1, x_2, \dots, x_n\}$, kí hiệu $X \sim \text{Uniform}(S)$, nếu X có tập giá trị là S và

$$P(X = x_i) = \frac{1}{n}, \quad i = 1, \dots, n.$$

Nếu X là phần tử được “chọn ngẫu nhiên” từ tập S hữu hạn thì $X \sim \text{Uniform}(S)$. Các ví dụ điển hình là

- tung đồng xu “đồng chất”, $S = \{H, T\} = \{0, 1\} = \mathbb{B}$.
 - gieo xúc xắc “đồng chất”, $S = \{1, 2, 3, 4, 5, 6\}$.
 - chọn ngẫu nhiên một số nguyên trong khoảng từ 1 đến n , $S = \{1, 2, \dots, n\}$.
 - chọn ngẫu nhiên một chuỗi nhị phân chiều dài n , $S = \mathbb{B}^n = \{0, 1\}^n$.
2. X được gọi là có **phân phối Bernoulli** (Bernoulli distribution) với tham số p ($0 \leq p \leq 1$), kí hiệu $X \sim \text{Bernoulli}(p)$, nếu X có tập giá trị là $S = \mathbb{B}$ và

$$f(x) = P(X = x) = \begin{cases} p & \text{nếu } x = 1, \\ 1 - p & \text{nếu } x = 0. \end{cases}$$

Khi đó, X có kì vọng $E(X) = p$ và phương sai $\text{Var}(X) = p(1 - p)$.

Trong thí nghiệm tung một đồng xu với xác suất ra ngửa p , gọi X là “số lần được ngửa” thì $X \sim \text{Bernoulli}(p)$. Trường hợp đồng xu đồng chất thì $X \sim \text{Bernoulli}(0.5)$.

Trong thí nghiệm T , biến cổ A có $P(A) = p$, ta có $\mathbb{I}_A \sim \text{Bernoulli}(p)$.

3. X được gọi là có **phân phối nhị thức** (binomial distribution) với tham số n ($n \in \mathbb{N}$), p ($0 \leq p \leq 1$), kí hiệu $X \sim \mathcal{B}(n, p)$, nếu X có tập giá trị là $S = \{0, 1, \dots, n\}$ và

$$f(x) = P(X = x) = C_n^x p^x (1 - p)^{n-x}, \quad x \in \{0, 1, \dots, n\}.$$

Khi đó, X có kì vọng $E(X) = np$ và phương sai $\text{Var}(X) = np(1 - p)$.

Cho thí nghiệm T với biến cố A có $P(A) = p$. Xét thí nghiệm R “thực hiện T lặp lại n lần độc lập”, gọi X là “số lần A xảy ra” thì $X \sim \mathcal{B}(n, p)$.

Mệnh đề B.3.5. Nếu X_1, \dots, X_n **độc lập và cùng phân phối** (independent and identically distributed - iid) Bernoulli với tham số p , thường kí hiệu $X_1, \dots, X_n \stackrel{\text{iid}}{\sim} \text{Bernoulli}(p)$, và $X = \sum_{i=1}^n X_i$ thì $X \sim \mathcal{B}(n, p)$. \triangle

4. X được gọi là có **phân phối hình học** (geometric distribution) với tham số p ($0 < p \leq 1$), kí hiệu $X \sim \text{Geometric}(p)$, nếu X có tập giá trị là $S = \{1, 2, \dots\} = \mathbb{N}_{>0}$ và

$$f(x) = P(X = x) = (1 - p)^{x-1} p, \quad x \in \{1, 2, \dots\}.$$

Khi đó, X có kì vọng $E(X) = \frac{1}{p}$ và phương sai $\text{Var}(X) = \frac{1-p}{p^2}$.

Cho thí nghiệm T với biến cố A có $P(A) = p$. Xét thí nghiệm R “thực hiện T lặp lại nhiều lần độc lập cho đến khi A xảy ra thì dừng”, gọi X là “số lần thực hiện” thì $X \sim \text{Geometric}(p)$.

Ví dụ B.3.2. Trạng thái của một qubit được xác định bởi một vector phức đơn vị

$$|\psi\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \in \mathbb{C}^2 \text{ với } |\alpha_0|^2 + |\alpha_1|^2 = 1$$

mà khi đo sẽ được 0 hoặc 1 với xác suất tương ứng là $|\alpha_0|^2, |\alpha_1|^2$.

Như vậy, “đo qubit” là một thí nghiệm ngẫu nhiên mà nếu gọi X là kết quả đo thì X là biến ngẫu nhiên Bernoulli với tham số

$$p = P(X = 1) = |\alpha_1|^2.$$

Đặc biệt, khi qubit ở trạng thái “tổ hợp đều” như $|\psi\rangle = |+\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$ thì $p = \frac{1}{2}$ nên “đo qubit” chính là tung một đồng xu đồng chất.

Trạng thái của 2 qubit được xác định bởi một vector phức đơn vị

$$|\psi\rangle = \begin{bmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{bmatrix} \in \mathbb{C}^4 \text{ với } \sum_{xy \in \{0,1\}^2} |\alpha_{xy}|^2 = 1$$

mà khi đo qubit trái, phải sẽ được giá trị X, Y là các biến ngẫu nhiên Bernoulli với hàm xác suất đồng thời

$$f_{XY}(x, y) = P(X = x, Y = y) = |\alpha_{xy}|^2, x, y \in \{0, 1\}.$$

Xét trạng thái “tổ hợp đều” $|\psi_1\rangle = \left[\frac{1}{2} \quad \frac{1}{2} \quad \frac{1}{2} \quad \frac{1}{2}\right]^T \in \mathbb{C}^4$, kết quả đo XY có phân phối đều trên tập $\mathbb{B}^2 = \{00, 01, 10, 11\}$ vì

$$f_{XY}(x, y) = \frac{1}{4}, \forall xy \in \{0, 1\}^2.$$

X (tương tự Y) cũng có phân phối đều trên tập $\{0, 1\}$ vì

$$f_X(x) = P(X = x) = \sum_{y \in \{0,1\}} f_{XY}(x, y) = \frac{1}{2}, \forall x \in \{0, 1\}.$$

Hơn nữa, X, Y độc lập vì

$$\begin{aligned} \text{Cov}(X, Y) &= P(X = 1, Y = 1) - P(X = 1)P(Y = 1) \\ &= \frac{1}{4} - \frac{1}{2} \frac{1}{2} = 0. \end{aligned}$$

Như vậy, đây là trường hợp tung 2 đồng xu đồng chất độc lập.

Xét trạng thái “rời” $|\psi_2\rangle = |\Phi^+\rangle = \left[\frac{1}{\sqrt{2}} \quad 0 \quad 0 \quad \frac{1}{\sqrt{2}}\right]^T \in \mathbb{C}^4$, kết quả đo XY không có phân phối đều trên tập \mathbb{B}^2 , cụ thể

$$P(XY = 00) = P(XY = 11) = \frac{1}{2}, P(XY = 01) = P(XY = 10) = 0.$$

X (tương tự Y) lại có phân phối đều trên tập $\{0, 1\}$ vì $P(X = 0) = P(X = 1) = \frac{1}{2}$.

Lưu ý, X, Y không độc lập vì $\text{Cov}(X, Y) = \frac{1}{2} - \frac{1}{2} \frac{1}{2} = \frac{1}{4}$. Hơn nữa với mọi $b \in \mathbb{B}$

$$P(Y = b|X = b) = 1, \quad P(X = b|Y = b) = 1.$$

Đây là trường hợp tung 2 đồng xu đồng chất “liên kết hoàn hảo”. □

Phụ lục C

Các trạng thái và công lượng tử thông dụng

C.1 Các trạng thái lượng tử thông dụng

C.1.1 1 qubit

$$|0\rangle = 1|0\rangle + 0|1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

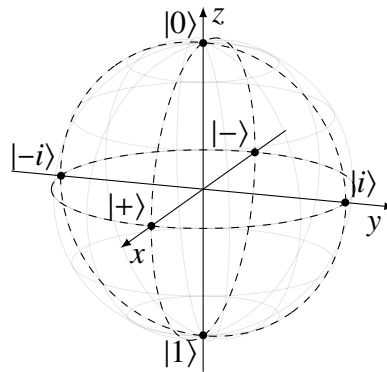
$$|1\rangle = 0|0\rangle + 1|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

$$|-\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$$

$$|i\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{i}{\sqrt{2}}|1\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} \end{bmatrix}$$

$$|-i\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{i}{\sqrt{2}}|1\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{i}{\sqrt{2}} \end{bmatrix}$$



$B_Z = \{|0\rangle, |1\rangle\}$, $B_X = \{|+\rangle, |-\rangle\}$, $B_Y = \{|i\rangle, |-i\rangle\}$ là các cơ sở trực chuẩn của \mathbb{C}^2 , trong đó B_Z là cơ sở chuẩn tắc và thường được gọi là **cơ sở tính toán** (computational basis).

C.1.2 2 qubit

Với hệ 2 qubit, ngoài các trạng thái cơ bản ứng với các vector cơ sở chuẩn tắc là $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ thì các trạng thái sau cũng hay được dùng, gọi là các **trạng thái Bell** (Bell state)

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}, \quad |\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ -\frac{1}{\sqrt{2}} \end{bmatrix},$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) = \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}, \quad |\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) = \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}.$$

Các trạng thái này tạo thành một cơ sở trực chuẩn của \mathbb{C}^4 được gọi là **cơ sở Bell** (Bell basis).

C.1.3 3 qubit

Các trạng thái vướng 3 qubit sau đây hay được dùng.

- **Trạng thái GHZ** (Greenberger–Horne–Zeilinger state)

$$|\text{GHZ}\rangle = \frac{1}{\sqrt{2}}|000\rangle + \frac{1}{\sqrt{2}}|111\rangle.$$

- **Trạng thái W** (Wolfgang Dür state)

$$|\text{W}\rangle = \frac{1}{\sqrt{3}}|001\rangle + \frac{1}{\sqrt{3}}|010\rangle + \frac{1}{\sqrt{3}}|100\rangle.$$

C.1.4 n qubit

$$|+\rangle^{\otimes n} = \sum_{k=0}^{2^n-1} \frac{1}{\sqrt{2^n}}|k\rangle$$

C.2 Các cổng lượng tử thông dụng

C.2.1 1 qubit

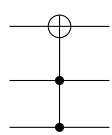
Cổng	Tác động trên cơ sở tính toán	Ma trận
Đơn vị I	$I 0\rangle = 0\rangle, \quad I 1\rangle = 1\rangle$	$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
Pauli X	$X 0\rangle = 1\rangle, \quad X 1\rangle = 0\rangle$	$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli Y	$Y 0\rangle = i 1\rangle, \quad Y 1\rangle = -i 0\rangle$	$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli Z	$Z 0\rangle = 0\rangle, \quad Z 1\rangle = - 1\rangle$	$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard H	$H 0\rangle = +\rangle, \quad H 1\rangle = -\rangle$	$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Pha S	$S 0\rangle = 0\rangle, \quad S 1\rangle = i 1\rangle$	$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
T	$T 0\rangle = 0\rangle, \quad T 1\rangle = e^{i\pi/4} 1\rangle$	$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Dịch pha $R(\theta)$	$R(\theta) 0\rangle = 0\rangle, \quad R(\theta) 1\rangle = e^{i\theta} 1\rangle$	$R(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}$

C.2.2 3 qubit

Cổng Toffoli (Toffoli gate) là cổng 3 qubit được xác định bởi

$$\text{Toffoli}|a\rangle|b\rangle|c\rangle = |a\rangle|b\rangle|ab \oplus c\rangle, \quad a, b, c \in \{0, 1\}.$$

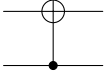
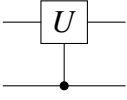

Kí hiệu và ma trận biểu diễn của cổng Toffoli là



Toffoli =

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

C.2.3 2 qubit

Cổng	Kí hiệu	Tác động trên cơ sở tính toán	Ma trận
CNOT		$\text{CNOT} 00\rangle = 00\rangle$ $\text{CNOT} 01\rangle = 01\rangle$ $\text{CNOT} 10\rangle = 11\rangle$ $\text{CNOT} 11\rangle = 10\rangle$	$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
C-U		$CU 00\rangle = 00\rangle$ $CU 01\rangle = 01\rangle$ $C 10\rangle = 1\rangle U 0\rangle$ $CU 11\rangle = 1\rangle U 1\rangle$	$CU = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{11} & u_{12} \\ 0 & 0 & u_{21} & u_{22} \end{bmatrix}$
SWAP		$\text{SWAP} 00\rangle = 00\rangle$ $\text{SWAP} 01\rangle = 10\rangle$ $\text{SWAP} 10\rangle = 01\rangle$ $\text{SWAP} 11\rangle = 11\rangle$	$\text{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Tài liệu tham khảo

- [1] Chris Bernhardt, *Quantum Computing for Everyone*, The MIT Press (2019).
- [2] Michael A. Nielsen and Isaac L. Chuang, *Quantum Computation and Quantum Information*, 10th Anniversary edition, Cambridge University Press (2010).
- [3] Noson S. Yanofsky and Mirco A. Mannucci, *Quantum Computing for Computer Scientists*, Cambridge University Press (2008).
- [4] Phillip Kaye, Raymond Laflamme and Michele Mosca, *An Introduction to Quantum Computing*, Oxford University Press (2007).
- [5] Thomas G. Wong, *Introduction to Classical and Quantum Computing*, Rooted Grove (2022).
- [6] IBM Quantum Learning (<https://learning.quantum.ibm.com/>).