



UNIVERSIDAD
DE MÁLAGA



INTRODUCCIÓN A LA COMPUTACIÓN CUÁNTICA

Autor:

David Castaño Bandín
SCBI (Universidad de Málaga)

25/11/2023



VICEPRESIDENCIA
PRIMERA DEL GOBIERNO
MINISTERIO
DE ASUNTOS ECONÓMICOS
Y TRANSFORMACIÓN DIGITAL

SECRETARÍA DE ESTADO
DE DIGITALIZACIÓN E
INTELIGENCIA ARTIFICIAL



Plan de Recuperación,
Transformación
y Resiliencia



Financiado por
la Unión Europea
NextGenerationEU



Índice general

Introducción	9
Código de colores	10
I Conceptos básicos	11
1. Formalismo matemático	13
1.1. Números complejos	13
1.1.1. Introducción	13
1.1.2. Forma cartesiana, polar y conjugación compleja	14
1.1.3. Operaciones básicas	17
1.1.4. Casos particulares	18
1.2. Vectores	19
1.2.1. Espacio Vectorial Complejo	19
1.2.2. Bases	20
1.2.3. Espacios de Hilbert	24
1.2.4. Bases ortogonales	26
1.3. Operadores	27
1.3.1. Operadores y matrices	27
1.3.2. Producto externo	29
1.3.3. Base canónica de operadores	30
1.3.4. El espacio vectorial $L(\mathcal{H})$	32
1.3.5. Clases de operadores	33
1.3.6. Comutador y Traza	38
1.3.7. Autovalores y autovectores	39
1.3.8. Factorización de unitarios	43
1.3.9. Funciones de Operadores	43
1.3.10. Matrices de Pauli	45
1.4. Tensores	48
1.4.1. Producto tensorial	48
1.4.2. Factorización y entrelazamiento	50
1.4.3. Producto tensorial múltiple	52
1.4.4. Espacio $L(\mathcal{H}^{\otimes})$	54
1.5. Probabilidades	56
1.5.1. Variables aleatorias	57
1.5.2. La conexión estadística	57
1.5.3. Probabilidades combinadas	60
1.5.4. Entropía de una variable aleatoria	61
2. Fundamentos de Mecánica Cuántica	63

2.1.	Axiomas	63
2.1.1.	¿Qué es un axioma?	63
2.1.2.	Axiomas de la mecánica cuántica	63
2.2.	Bases ortogonales	65
2.3.	Medidas Estadísticas	66
2.3.1.	Valor esperado	66
2.3.2.	Varianza y Desviación estándar	66
2.4.	Evolución temporal	67
2.5.	Estados Mezcla y Matriz Densidad	70
2.5.1.	Estado puro y mezcla	70
2.5.2.	Operador densidad	71
2.5.3.	Medidas en estados mezcla	75
2.5.4.	Estado mezcla bipartito	75
II	Fundamentos de Computación Cuántica	77
3.	Qubits	79
3.1.	Definición y bases	79
3.1.1.	Definición de Qubit	79
3.1.2.	Bases computacionales y bases X e Y	80
3.2.	El estado de un qúbit y la esfera de Bloch	81
3.2.1.	Parametrización del estado de un qúbit.	81
3.2.2.	La esfera de Bloch	82
4.	Puertas simples	85
4.1.	Rotaciones en la esfera de Bloch	85
4.1.1.	Rotaciones en X , Y , Z	86
4.1.2.	Parametrización de Euler	86
4.2.	Puertas simples	87
4.2.1.	Dos formas de escribir las matrices 2×2	87
4.2.2.	Puertas de fase	87
4.2.3.	Puertas discretas	88
4.2.4.	Descomposición	89
4.3.	Circuitos Cuánticos (1 qubit)	90
4.3.1.	Matriz de un circuito.	90
4.3.2.	Simulador de un estado con qiskit.	90
5.	Medidas, Parte I: Medida de 1 qúbit	91
5.1.	En la base computacional	91
5.1.1.	Superposiciones, medidas y colapso.	91
5.1.2.	Sobre medir en la base computaciones.	92
5.1.3.	Código de Qiskit: simulación de un estado y medida.	92
5.1.4.	Código de Qiskit: ejecución en un ordenador real.	92
5.2.	La moneda cuántica	92
5.2.1.	Código de Qiskit.	93
5.3.	Medidas en una base general	93
5.3.1.	Bases X , Y , Z	93
5.3.2.	Base arbitraria	94
5.4.	Valores esperados.	95
5.4.1.	Valor esperado de un observable arbitrario (operador hermítico).	95
5.4.2.	Valor esperado de un operador unitario (no necesariamente hermítico)	97

6. Multi-qubit: definiciones y puertas.	99
6.1. Algunas definiciones	99
6.1.1. Estados multi-qubit.	99
6.2. Entrelazamiento	101
6.2.1. Base de Bell	101
6.2.2. Medidas parciales	102
6.3. Circuitos Multiqubit	103
6.4. Puertas (multi-qubit) no controladas	104
6.4.1. Walsh-Hadamard	104
6.4.2. SWAP	105
6.5. Puertas controladas	105
6.5.1. CNOT	106
6.5.2. Control-SWAP	107
6.6. Puertas multicontroladas	108
6.6.1. CCNOT o Toffoli	108
6.6.2. X multi-controlada.	108
7. Medidas, Parte II: Medida de estados multi-Qubit	111
7.1. Medidas en la base computacional	111
7.2. Medidas en bases generales	111
7.2.1. Medidas de Pauli	111
7.2.2. Medidas de Bell	112
7.3. Valores esperados	113
7.4. Medida de Hadamard	113
7.4.1. Valor esperado de un operador a partir de $\langle X \rangle$ y $\langle Y \rangle$	113
7.4.2. Proyección de Hadamard	115
8. Entrelazamiento en acción	117
8.1. Desigualdades de Bell	117
8.1.1. Perfecta anticorrelación	118
8.1.2. Desigualdad CSCH	119
8.2. Experimento GHZ	121
8.3. Teleportación	122
8.4. Intercambio de Entrelazamiento	124
8.5. Teorema de no-clonación	125
9. Hardware: Técnicas de control y computación en RMN.	127
9.1. Introducción.	127
9.2. El espín nuclear	127
9.3. Qúubits de RMN	129
9.3.1. Hamiltoniano del sistema	129
9.3.2. Hamiltoniano de control	135
9.4. Técnicas de pulsos elementales.	138
9.4.1. Control cuántico, circuitos y pulsos	138
10. Decoherencia y desfase	145
10.1. Introducción	145
10.2. Decoherencia y las ecuaciones de Bloch.	146
10.3. Resumen	150

III Algoritmos Cuánticos	151
11.Elementos básicos de los algoritmos cuánticos	153
11.1. Circuitos	153
11.1.1. Qué es un circuito cuántico?	153
11.1.2. Ejemplo: Circuito de teleportación.	154
11.2. Retroceso de fase (Phase kickback)	155
11.3. Circuitos equivalentes	156
11.3.1. Conjugación	156
11.3.2. Conjugación de una exponencial	157
11.3.3. Varios qubits	158
11.4. Operadores de Clifford	159
11.5. Universalidad de la computación cuántica con puertas.	160
11.5.1. Teorema	160
11.5.2. Conjuntos de puertas universales	162
11.6. Medidas de calidad de un Circuitos.	162
12.Estado inicial y oráculos	165
12.1. Preparación de un estado inicial genérico	165
12.2. Oráculos (funciones digitales)	166
12.2.1. Construcción de funciones binarias. Los min-términos	167
12.2.2. Oráculos booleanos y de fase	168
13.Primeros algoritmos: algoritmos del oráculo.	171
13.1. Algoritmo de Deutsch-Jozsa	173
13.2. Algoritmo de Bernstein-Vazirani	174
13.3. Algoritmo de Simon	176
14.QFT: Quantum Fourier Transform	179
14.1. La QFT en computación cuántica	179
14.2. Intuición	183
14.2.1. Contando en la base de Fourier.	183
14.3. Circuito que implementa la QFT.	185
14.3.1. Algunos comentarios sobre la implementación.	186
14.4. Complejidad y QFT aproximada	186
14.4.1. Complejidad y ventaja exponencial	186
14.4.2. QFT aproximada	186
15.QPE: Estimación de Fase Cuántica	189
15.1. Introducción	189
15.2. Circuito	189
15.3. Formulación matemática	190
15.4. ¿Y si no conocemos el autoestado?	192
16.Algoritmo de Shor (Periodicity Finding)	195
16.1. Introducción	195
16.1.1. Criptografía y factorización.	195
16.1.2. Algoritmo de factorización.	196
16.1.3. Explicación cualitativa	197
16.1.4. Formalismo matemático	198
16.2. Hallar del periodo de una función (Period Finding)	199
16.2.1. La función	199

16.2.2. Solución: Estimación de fase de un operador U	199
16.3. Implementación (ad hoc) en Qiskit para $N = 15$	202
16.3.1. Caso con muchos shots (ineficiente pero didáctico)	202
16.3.2. Caso shot a shot (óptimo)	202
17. Algoritmo de Shor: Implementación con $2n+3$ qubit	203
17.1. La idea	203
17.2. Explicación desgranada	204
17.2.1. Algoritmo cuántico de suma	204
17.2.2. Valor clásico + registro cuántico (puerta $\phi ADD(a)$)	206
17.2.3. Suma modulada (puerta $\phi ADD(a)MOD(N)$)	207
17.2.4. Multiplicación modulada (puerta $CMULT(a)MOD(N)$)	209
17.2.5. Puerta controlada $C-U_a$	210
17.2.6. Exponencial modulada (puerta $C-U_{a^s}$)	211
17.2.7. Circuito final con $4n + 2$ qúbits (sin la simplificación del registro de conteo)	212
17.2.8. Circuito final con $2n + 3$. Algoritmo de estimación iterativa de fase (IPE)	212
17.3. Implementación aproximada de la QFT	215
17.4. Implementación de las SWAP controladas	215
18. Algoritmo de Grover (Amplificación de amplitud)	217
18.1. Introducción	217
18.2. Explicación geométrica del algoritmo	218
18.2.1. Estado inicial: superposición	218
18.2.2. Amplificación de amplitud mediante iteraciones del algoritmo	219
18.3. Número conocido de soluciones	223
18.3.1. Generalización de las expresiones de la sección 18.2 para M soluciones	224
18.3.2. Número de iteraciones	225
18.3.3. Extra: Formulación recursiva de $k(t)$ y $l(t)$.	225
18.4. Número desconocido de soluciones	226
18.4.1. Conocimientos previos	227
18.4.2. Algoritmo para el caso de M desconocido	228
18.5. Conteo de soluciones (Quantum counting)	230
18.5.1. Breve resumen de la estimación de fase cuántica (QPE)	230
18.5.2. Estimación de fase con el operador de Grover	231
18.6. Consideraciones sobre la implementacion	233
18.6.1. Creación de un difusor U_{Ψ_0}	233
18.7. Distribución de probabilidad inicial aleatoria	235
18.7.1. Algoritmo	235
18.7.2. Evolución de las amplitudes (M soluciones)	235
18.7.3. Propiedades de las soluciones: Probabilidad de acierto	237
18.7.4. Resumen de la sección	240
18.8. Implementaciones con qiskit	240
18.8.1. Puerta multicontrolada Z (MCZ)	240
18.8.2. Difusor genérico	241
18.8.3. Oráculo “trivial”	241
18.8.4. Oráculos que verifican condiciones: sudoku 2×2	241
19. Criptografía y Quantum Key Distribution (QKD)	243
19.1. Introducción	243
19.1.1. Criptografía	244
19.1.2. Criptografía de clave privada	244
19.1.3. Criptografía de clave pública	245

19.2. Fundamentos de QKD	246
19.2.1. Un poco de historia	247
19.2.2. Conceptos generales	247
19.2.3. El origen de la seguridad	248
19.2.4. La elección de la luz	248
19.2.5. Tratamiento cuántico de la información: marcos P&M y EB	249
19.3. Protocolos de QKD: Tres familias	250
19.3.1. Protocolos de variable discreta	250
19.3.2. Protocolos de variable continua	253
19.3.3. Protocolos Distributed-phase-reference	254
Bibliografía	257

Introducción

En estas notas vamos a ver una introducción a la Computación Cuántica y los algoritmos cuánticos más famosos. En las mismas, se presentará todo lo necesario para entender estos algoritmos suponiendo que el lector no tiene conocimiento alguno sobre Mecánica Cuántica ni Computación Cuántica.

La parte I de estas notas se dedica a introducir los Conceptos Básicos necesarios para entender el resto de las mismas. Esto incluye una introducción al formalismo matemático (capítulo 1), donde se verán los conceptos matemáticos necesarios para abordar el segundo capítulo de esta parte, una introducción a la Mecánica Cuántica (capítulo 2).

Después de estos dos capítulos introductorios, pasamos a la parte II, donde damos una introducción a las Fundamentos de la Computación Cuántica: qúbits, puertas, medidas, circuitos, hardware, decoherencia,

Finalmente, cerramos las notas con la parte III, donde presentando de forma detallada los algoritmos más famosos. Esto incluye el algoritmo de Quantum Fourier Transform (capítulo 14), el de Estimación Cuántica de Fase (capítulo 15) y los archiconocidos algoritmos de Grover (capítulo 18) y Shor (capítulo 16). Además hablaremos también de Criptografía y Quantum Key Distribution (capítulo 19).

Como veremos a lo largo de estas notas, los algoritmos en computación cuántica se basan en construir **circuitos cuánticos** a los que añadimos **puertas cuánticas**. A estos circuitos se les pasa como entrada un **estado cuántico**. Las puertas de circuito operan sobre el estado y nos devuelven otro estado cuántico. Este último representará la solución de nuestro problema, mientras que el circuito representa el **algoritmo cuántico** que se usa para hallar la solución. La idea es bastante similar a la de la computación clásica, donde se parte de una cadena bit, se aplican una serie de puertas lógicas (AND, OR, NOT,...) y se obtiene otra cadena de bits que codifica la solución.

Como veremos más adelante, la computación cuántica (al igual que la clásica) es un paradigma de **computación universal**. Es decir, al menos sobre el papel, todo lo que se puede calcular con un ordenador clásico (todo lo que es **computable**), se puede calcular con un ordenador cuántico. La gran diferencia entre estos dos modelos es que el primero aprovecha las propiedades exóticas de la mecánica cuántica (superposición y entrelazamiento) para plantear algoritmos más rápidos que los clásicos. De esta forma, la computación cuántica tiene el potencial de acelerar ciertos cálculos.

Nota: Paradigmas de Computación Cuántica

Habitualmente cuando se habla de **computación cuántica** se suele hablar de la **Computación Cuántica basada en Puertas**. Los otros dos paradigmas de computación cuántica son la **Computación Cuántica Adiabática** y el **Quantum Annealing**. Los dos primeros son paradigmas de computación universal mientras que el Quantum Annealing no lo es.

En estas notas hablaremos solo de Computación Cuántica basada en Puertas, a partir de ahora simplemente computación cuántica o QC (del inglés, Quantum Computing).

Código de colores

Teoremas, Lemmas, Definiciones, Demostraciones y conceptos importantes

Ejercicios

Nota

Notas o aclaraciones

Ejemplo

Ejemplos

Cuadros naranjas

Referencias a cuadernos de Jupyter Notebook

Parte I

Conceptos básicos

Capítulo 1

Formalismo matemático

La Mecánica Cuántica (así como la Computación Cuántica) hace uso de unas matemáticas con las que es necesario familiarizarse. En esta capítulo vamos a repasar el formalismo matemático básico para el resto del curso.

A pesar de que la expresión “formalismo matemático” puede llegar a imponer respeto, a lo largo de este capítulo veremos que las cosas son más simples de lo que pudieran parecer en un principio. La Mecánica Cuántica alcanza unos niveles de complejidad matemática insospechados, pero al mismo tiempo es sorprendente como se pueden entender los fundamentos de la misma con una serie de pinceladas de ciertos conceptos matemáticos: *números complejos, vectores, operadores (matrices), tensores* y algo de *probabilidad*.

Los conceptos que se presentan a continuación están un pasito por encima del nivel de matemáticas de bachillerato, pero no mucho más. Nuestro objetivo es entender las bases de la Mecánica Cuántica, sin llegar a plantearnos trabajar de forma seria con sus ecuaciones. Es decir, ser capaces de entender los conceptos y trabajar con las soluciones, no con las ecuaciones. Por este motivo, los conceptos matemáticos más complejos como integrales o resolución de ecuaciones diferenciales están más allá del objetivo de este curso.

En resumen, no hay que tener miedo a las matemáticas de este curso, son sencillas.

Este capítulo se basa en [1], que a su vez toma como referencias los capítulo 2 de [2], [3] y [4]

1.1. Números complejos

1.1.1. Introducción

Para entender de donde surge la idea de los número complejos, tenemos que recordar primero que el cuadrado de cualquier número real, $a \in \mathbb{R}$, es *siempre* positivo: $a^2 > 0$. Por ejemplo

$$2^2 = (-2)^2 = +4$$

Por eso, la raíz cuadrada de un número real *sólo* existe (como número real) si el número es positivo y tiene dos soluciones, la positiva y la negativa:

$$\sqrt{4} = \pm 2$$

En este punto, surge la siguiente pregunta: cual es la solución de la siguiente ecuación?

$$x^2 + 1 = 0 \quad \Rightarrow \quad x^2 = -1 \quad \Rightarrow \quad x = \sqrt{-1} \tag{1.1}$$

Como comentamos, no hay ningún número real que cumpla esa ecuación. Podemos pues, definir un nuevo número. Este número no pertenece a los números reales:

Definición 1 Se postula la existencia de un número, i , que es solución de la ecuación

$$i^2 = -1 \quad (1.2)$$

Equivalentemente $i = \sqrt{-1}$. No hay nada misterioso en i , se puede sumar, restar, multiplicar y dividir normalmente:

$$\begin{aligned} i + i &= 2i \\ i - i &= 0 \\ i + 2i &= 3i \\ i^3 &= i^2 \cdot i = -i \\ i^4 &= i^2 \cdot i^2 = 1 \\ \frac{i}{i} &= 1 \end{aligned}$$

Una vez definido este número, podemos resolver cualquier raíz con un número negativo:

$$x^2 + 4 = 0 \Rightarrow x = \sqrt{-4} = \sqrt{4}\sqrt{-1} = 2i \quad (1.3)$$

En resumen: la solución pasa por ampliar el conjunto de los números reales definiendo un nuevo conjunto, el de los **números imaginarios o complejos**, \mathbb{C}

1.1.2. Forma cartesiana, polar y conjugación compleja

Los números complejos pueden escribirse de dos formas diferentes: la forma **cartesiana** y la forma **polar**.

1.1.2.1. Forma cartesiana

Un número complejo, $z \in \mathbb{C}$, se representa en la **forma cartesiana** mediante dos números reales $x, y \in \mathbb{R}$,

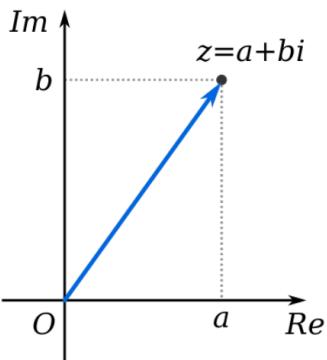
$$z = x + iy \quad \text{donde} \quad \begin{cases} x & \text{es la parte real} \\ y & \text{es la parte imaginaria} \end{cases} \quad (1.4)$$

Como podemos ver, tenemos dos casos:

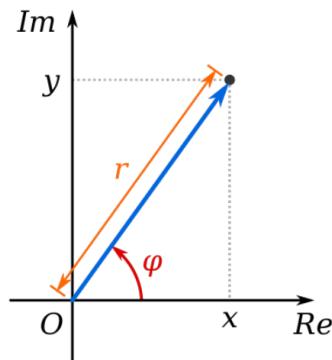
- Si $y = 0 \rightarrow z = x$ es un **número real puro**.
- Si $x = 0 \rightarrow z = iy$ es un **número imaginario puro**.

Aquí podemos ver perfectamente que los números complejos (o imaginarios) son una extensión de los números reales, en el sentido de que los engloban: como acabamos de ver, un número real puede verse como un número complejo con la parte imaginaria igual a cero.

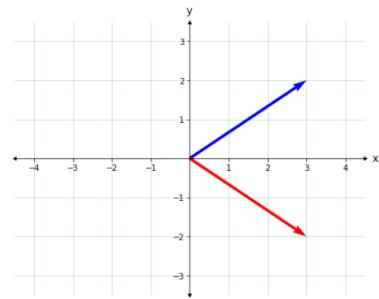
Es bien conocido que los números reales se representan sobre una recta, la **recta real**. Los números complejos por su parte se representan en un plano, el **plano complejo**, donde el eje horizontal es la recta real y el eje vertical es el eje imaginario. Esto se ve precisamente muy bien en la forma cartesiana, pues x sería el valor del eje real e y sería el valor del eje imaginario. Podemos ver esto más claramente en la Fig. 1.1b).



(a) Representación de un número $z = a + bi$ en el plano complejo.



(b) Representación polar de un número complejo $z = x + iy$. En el texto se usan ρ y θ en lugar de r y ϕ , pero es lo mismo.



(c) Un número complejo y su conjugado

Figura 1.1: Representaciones de números en el plano complejo.

1.1.2.2. Forma Polar

Teorema 1 (Fórmula de Euler) Dado un ángulo $\theta \in [0, 2\pi)$ las dos expresiones siguientes son equivalentes

$$\cos \theta + i \sin \theta = e^{i\theta} \quad (1.5)$$

A esta identidad se la denomina **Fórmula de Euler**.

Demostración: La demostración de la Fórmula de Euler viene de expandir ambos miembros en serie de Taylor en torno a $\theta = 0$ y comprobar que ambas series son iguales

$$\begin{aligned} e^{i\theta} &= 1 + i\theta + \frac{1}{2}(i\theta)^2 + \frac{1}{3!}(i\theta)^3 + \dots \\ &= 1 - \frac{1}{2}\theta^2 + \dots + i\left(\theta - \frac{1}{3!}\theta^3 + \dots\right) \\ &= \cos \theta + i \sin \theta \end{aligned} \quad (1.6)$$

■

Con esto, veamos lo que es la forma polar:

El número $z = x + iy$ se puede representar en **forma polar**

$$z = \rho e^{i\theta} = \rho(\cos \theta + i \sin \theta) \quad (1.7)$$

de donde obtenemos las componentes cartesianas

$$x = \rho \cos \theta, \quad y = \rho \sin \theta. \quad (1.8)$$

Tanto ρ como θ son números reales y se denominan **módulo** y **fase**.

En principio, esta forma de representar los números complejos puede parecer rara, pero es muy fácil de entender. Para ello, solo hay que recordar que los números complejos los podemos representar en un plano. Los puntos en un plano pueden verse como **vectores** que parten del origen y cuya punta acaba en el punto que queremos describir. La forma habitual de referirse a estos vectores (puntos) es usar las coordenadas sobre los dos ejes, es decir, usar lo que en matemáticas se llama **sistema de coordenadas cartesiano** (ver la Fig. 1.1b)). Sin embargo, otra forma de describir estos puntos es

usando el módulo del vector (su longitud) y el ángulo que este forma con el eje horizontal (ver la Fig. 1.1b)). A esta forma de describir el plano se la llama **sistema de coordenadas polar**.

La forma polar que acabamos de ver de los números complejos (Ec. 1.7) no es más que describir el número complejo usando el sistema de coordenadas polar. En concreto, el número real ρ es el módulo del vector y el número real θ es su fase, es decir, el ángulo con el eje horizontal. Usando el **teorema de Pitágoras** para los triángulos rectángulos es muy fácil comprobar que ρ es el módulo si tenemos en cuenta que x e y son los catetos:

$$x^2 + y^2 = \rho^2 \cos^2 \theta + \rho^2 \sin^2 \theta = \rho^2 (\cos^2 \theta + \sin^2 \theta) = \rho^2. \quad (1.9)$$

Nota: Números con módulo 1

Como ya comentamos, a θ se le denomina **fase**. Habitualmente también se denominan fases a los números complejos con $\rho = 1$ pues en estos casos tenemos $z = e^{i\theta}$. Esto es un abuso del lenguaje pero es algo muy extendido.

1.1.2.3. Conversión entre forma cartesiana y polar

La conversión de la representación *polar a cartesiana* es muy sencilla gracias a las fórmulas de Euler

$$z = re^{i\theta} = x + iy \quad \text{con} \quad \begin{cases} x = r \cos \theta \\ y = r \sin \theta. \end{cases} \quad (1.10)$$

La conversión inversa, *de cartesiana a polar* es un poco más delicada. Formalmente sería

$$z = x + iy = re^{i\theta} \quad \text{con} \quad \begin{cases} r = \sqrt{x^2 + y^2} \\ \theta = \arctan(y/x). \end{cases} \quad (1.11)$$

Nota: Signo de la arcotangente

En la fórmula de la arcotangente de la Ec. (1.11) hay que elegir el signo con cuidado. La arcotangente devuelve valores en $(-\pi/2, \pi/2)$, pero sabemos que los ángulos en la circunferencia están en $[0, 2\pi)$. Por ello, hay que aplicar el siguiente criterio:

- Si $x = 0$ e $y > 0$ (sobre el eje y) $\Rightarrow \theta = \pi$
- Si $x = 0$ e $y < 0$ (sobre el eje y) $\Rightarrow \theta = 3\pi/2$
- Si $x > 0$ e $y \geq 0$ (primer cuadrante) $\Rightarrow \theta = \arctan(y/x)$
- Si $x < 0$ e $y \geq 0$ (segundo cuadrante) $\Rightarrow \theta = \arctan(-y/x) + \pi/2$
- Si $x < 0$ e $y < 0$ (tercer cuadrante) $\Rightarrow \theta = \arctan(y/x) + \pi$
- Si $x < 0$ e $y < 0$ (cuarto cuadrante) $\Rightarrow \theta = \arctan(-y/x) + 3\pi/2$

Ejercicio 1 Calcula a mano la forma polar de los números:

$$4 + 3i, \quad 7 - 2i, \quad -5 + 5i, \quad -3 + 4i \quad (1.12)$$

1.1.2.4. Conjugación compleja

Todo número complejo, z , lleva *asociado* otro, z^* , denominado el **complejo conjugado** que se obtiene cambiando $i \rightarrow -i$, tanto en forma cartesiana

$$z = x + iy \quad \leftrightarrow \quad z^* = x - iy \quad (1.13)$$

como en forma polar

$$\boxed{z = \rho e^{i\theta} \quad \leftrightarrow \quad z^* = \rho e^{-i\theta}} \quad (1.14)$$

Hablando de representaciones, la conjugación compleja no es más que una reflexión sobre el eje horizontal como podemos ver en la Fig. 1.1.

Matemáticamente, el complejo conjugado z^* de un número complejo z es el número por el cual hay que multiplicar z para obtener el módulo cuadrado, es decir:

$$\boxed{|z|^2 = z \cdot z^* = \rho^2} \quad (1.15)$$

1.1.3. Operaciones básicas

Los números complejos \mathbb{C} forman una estructura matemática denominada *cuerpo*. Esto quiere decir que admiten dos operaciones *internas*: la **suma** y la **multiplicación**. Vamos a estudiarlas por separado

1.1.3.1. Suma

- En *forma cartesiana* se suman las partes real e imaginaria por separado

$$(a + ib) + (c + id) = (a + c) + i(b + d) \quad (1.16)$$

La resta es obvia, ya que a, b, c, d pueden ser números negativos.

- En *forma polar*, la suma de dos números complejos no admite ninguna simplificación, y deben transformarse primeramente a forma cartesiana, para sumarse.

$$z + w = \rho e^{i\theta} + \sigma e^{i\phi} = (\rho \cos \theta + \sigma \cos \phi) + i(\rho \sin \theta + \sigma \sin \phi) \quad (1.17)$$

1.1.3.2. Multiplicación

- En *forma cartesiana* debemos multiplicar todos los factores entre sí, y teniendo en cuenta que $i^2 = -1$

$$(a + ib)(c + id) = ac + aid + ibc + i^2bd = (ab - bd) + i(ac + bd) \quad (1.18)$$

- En la *forma polar*, la cosa es más simple, pues solo hay que multiplicar los módulos y sumar las fases:

$$zw = re^{i\theta}se^{i\phi} = rs e^{i(\theta+\phi)} \quad (1.19)$$

1.1.3.3. Valor absoluto

El cuadrado de un número real $a \in \mathbb{R}$ es otro número real positivo $a^2 > 0$. Ello nos permite definir el valor absoluto $|a| = \sqrt{a^2}$ que es el mismo para a y para $-a$.

Esto no sucede con un número complejo z . En efecto, $z^2 = x^2 - y^2 + 2ixy$ es complejo. Sin embargo, el producto de un número por su conjugado es un número *real y positivo*

$$zz^* = (x + iy)(x - iy) = x^2 + y^2 > 0 \quad (1.20)$$

lo nos permite definir el **valor absoluto** de un número complejo

$$\boxed{|z| = \sqrt{zz^*} = \sqrt{x^2 + y^2}} \quad (1.21)$$

El valor absoluto de una fase es 1

$$|e^{i\theta}| = \sqrt{e^{i\theta}e^{-i\theta}} = \sqrt{e^{i(\theta-\theta)}} = \sqrt{e^0} = 1 \quad (1.22)$$

El valor absoluto de un número complejo coincide con el **módulo** escrito en forma polar

$$|z| = \sqrt{zz^*} = \sqrt{\rho e^{i\theta} \rho e^{-i\theta}} = \sqrt{\rho^2} \Rightarrow |z| = \rho \quad (1.23)$$

1.1.3.4. División

Al igual que la multiplicación, en forma cartesiana, la división no es simple. Sea $z = a + ib$ y $w = c + id$

$$\frac{z}{w} = \frac{z}{w} \frac{w^*}{w^*} = \frac{(a + ib)(c - id)}{|w|^2} = \frac{ac + bd + i(bc - ad)}{c^2 + d^2} = \frac{ac + bd}{c^2 + d^2} + i \frac{bc - ad}{c^2 + d^2} \quad (1.24)$$

En forma polar la división es tan sencilla como la multiplicación. Se toma el cociente de los módulos y la resta de las fases

$$\frac{z}{w} = \frac{\rho e^{i\theta}}{\sigma e^{i\phi}} = \frac{\rho}{\sigma} e^{i(\theta - \phi)} \quad (1.25)$$

1.1.4. Casos particulares

1.1.4.1. Sumas nulas

En muchas ocasiones nos encontraremos la siguiente representación del numero cero (complejo) $0 = 0 + i0$

$$\sum_{k=0}^{N-1} e^{2\pi i k/N} = e^{2\pi i 0/N} + e^{2\pi i 1/N} + \dots + e^{2\pi i (N-2)/N} + e^{2\pi i (N-1)/N} = 0. \quad (1.26)$$

No es trivial ver que esta igualdad es cierta, así que no vamos a demostrarla.

Si multiplicamos todas la fases por un número entero j tal que $1 \leq j \leq N - 1$, el resultado es el mismo

$$\sum_{k=0}^{N-1} e^{2\pi i j k/N} = e^{2\pi i 0/N} + e^{2\pi i j/N} + \dots + e^{2\pi i j(N-2)/N} + e^{2\pi i j(N-1)/N} = 0. \quad (1.27)$$

Sin embargo si $j = 0, N, 2N, \dots = 0 \bmod N$, entonces **la suma no se anula** y su valor es igual a N . Tomemos por ejemplo $j = N$

$$\sum_{k=0}^{N-1} e^{2\pi i (N) k/N} = \sum_{k=0}^{N-1} e^{2\pi i k} = \sum_{k=0}^{N-1} 1 = N. \quad (1.28)$$

Una manera de resumir todos los casos anteriores en una sola expresión involucra la función **δ de Kronecker**

$$\delta_{ij} = \begin{cases} 0 & \text{si } i \neq j \\ 1 & \text{si } i = j \end{cases} \quad (1.29)$$

Con ella podemos enunciar el siguiente resultado

$$\boxed{\frac{1}{N} \sum_{k=0}^{N-1} e^{2\pi i j k/N} = \delta_{j0 \bmod N}}, \quad (1.30)$$

que usaremos al estudiar la transformada de Fourier cuántica.

1.1.4.2. Desigualdad triangular

Desigualdad triangular:

El módulo de la suma de dos números complejos verifica que

$$|z + w| \leq |z| + |w| \quad (1.31)$$

donde la igualdad sólo se verifica cuando ambos números complejos son paralelos en el plano complejo.

1.2. Vectores

1.2.1. Espacio Vectorial Complejo

1.2.1.1. Definición

Definición 2 De forma poco rigurosa, definiremos un **vector de dimensión N** como una columna de N números complejos

$$|u\rangle = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} \quad (1.32)$$

-
- El símbolo $|u\rangle$ representa al vector y se denomina **ket** en la **notación de Dirac**. Otra notación sería \vec{u} , pero esta nunca se usa en Mecánica Cuántica.
 - Los números complejos $u_i \in \mathbb{C}$ con $i = 1, \dots, N$ se denominan **componentes** del vector $|u\rangle$. Estas componentes toman un valor concreto cuando especificamos una **base**. En otra base, estos elementos pueden tener otro valores. Es decir, el vector es un objeto abstracto.

Definición 3 La colección de todos los posibles vectores de N componentes, con las propiedades de suma y multiplicación forman un **espacio vectorial**, V de dimension compleja N .

Es decir, en un espacio vectorial tenemos las siguientes propiedades:

- Dos operaciones posibles: **suma** de dos vectores y **multiplicación de un vector por un número complejo** $\lambda \in \mathbb{C}$

$$|u\rangle + |v\rangle = \begin{bmatrix} u_1 + v_1 \\ u_2 + v_2 \\ \vdots \\ u_N + v_n \end{bmatrix} = |w\rangle, \quad \lambda|u\rangle = \begin{bmatrix} \lambda u_1 \\ \lambda u_2 \\ \vdots \\ \lambda u_N \end{bmatrix} \equiv |\lambda u\rangle \quad (1.33)$$

- Existencia de un **elementos neutro** (respecto a la suma). Todo vector de V se denota mediante el símbolo $|v\rangle$ menos el elemento neutro, que se escribe como 0.

$$|v\rangle + 0 = |v\rangle \quad (1.34)$$

- La existencia de un **elemento opuesto** (respecto a la suma):

$$|v\rangle + |-v\rangle = |v\rangle - |v\rangle = 0 \quad (1.35)$$

Nota: Dimensión de un espacio vectorial complejo

La **dimensión** es igual al número de cantidades (*grados de libertad*) que debemos fijar para especificar un vector. Como estamos lidiando con un espacio vectorial complejo donde nuestro vector está compuesto por N números complejos, la cantidad de números reales que debemos fijar es $2N$.

Entonces, podemos decir que la **dimensión compleja** de un espacio vectorial complejo V es N , o que su **dimensión real** es $2N$

$$\dim_{\mathbb{C}} V = N \iff \dim_{\mathbb{R}} V = 2N \quad (1.36)$$

Habitualmente, cuando se habla simplemente de *dimensión* se habla de la real.

1.2.1.2. Conjugación adjunta

La operación **conjugación adjunta**, \dagger , es una *extensión* de la **conjugación compleja** a los vectores.

Definición 4 Asociado a cada ket $|u\rangle$, definimos un vector **adjunto**, o **bra** $\langle u| \equiv (\langle u|)^\dagger$, que representamos mediante un vector fila con las componentes conjugadas complejas:

$$\dagger : |u\rangle = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} \rightarrow (\langle u|)^\dagger \equiv \langle u| = [u_1^* \ u_2^* \ \cdots \ u_N^*] \quad (1.37)$$

Consistentemente encontramos para el producto de un vector por un número complejo λ

$$\dagger : \lambda |u\rangle = |\lambda u\rangle \rightarrow (\lambda |u\rangle)^\dagger = \lambda^* \langle u| = \langle u| \lambda^* = \langle \lambda u| \quad (1.38)$$

ya que el producto de un vector por un número es comutativo.

Nota: Sobre la notación

En la notación con *kets* y *bras* vemos que es lo mismo escribir $\lambda |u\rangle$ que $|\lambda u\rangle$. Esto podemos pensarlo como que los números complejos entran y salen de los kets sin modificaciones.

Sin embargo, en los bras tenemos $\langle u| \lambda^* = \langle \lambda u|$. Es decir, sacar o meter un número complejo dentro de los bras implica una conjugación compleja del número.

Esto, como comentamos, es solo **notación**, es decir, un convenio para escribir las cosas.

Al igual que la conjugación compleja, la conjugación adjunta es una **involución**: su aplicación sucesiva devuelve el vector original

$$(|u\rangle^\dagger)^\dagger = \langle u|^\dagger = |u\rangle \quad (1.39)$$

es decir, $\dagger^2 = I$, el operador identidad.

1.2.2. Bases

Definición 5 En un espacio vectorial V de dimensión N una **base** es una colección de N vectores $\{|e_1\rangle, \dots, |e_N\rangle\}$ tales que, cualquier vector $|v\rangle \in V$ se puede expresar como una **combinación**

lineal de ellos

$$|v\rangle = \sum_{i=1}^N v_i |e_i\rangle \quad (1.40)$$

Los coeficientes v_i son las **componentes** de $|v\rangle$ en la base dada.

Existen *infinitas bases* en un espacio vectorial. Podemos escoger una de ellas y asociarle el siguiente conjunto de columnas

$$|e_1\rangle \sim \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \quad |e_2\rangle \sim \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \quad \dots \quad |e_{N-1}\rangle \sim \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 0 \end{bmatrix} \quad |e_N\rangle \sim \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (1.41)$$

De esta forma, cualquier vector, escrito como una combinación lineal de sus elementos adquiere la representación usual

$$\begin{aligned} |u\rangle &= u_1 |e_1\rangle + u_2 |e_2\rangle + \dots + u_N |e_N\rangle = \sum_{i=1}^N u_i |e_i\rangle \sim \\ &\sim u_1 \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} + u_2 \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} + \dots + u_{N-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 0 \end{bmatrix} + u_N \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix} \end{aligned} \quad (1.42)$$

1.2.2.1. Cambio de base

Existen *infinitas bases* en un espacio vectorial de dimensión finita. Todas ellas sirven para representar un vector arbitrario.

Consideremos dos bases $\{|e_i\rangle\}$ y $\{|f_j\rangle\}$ donde $i, j = 1, \dots, N$. Cualquier vector $|v\rangle$ se puede expandir de forma diferente en cada una de ellas

$$|v\rangle = \sum_{i=1}^N v_i |e_i\rangle = \sum_{i=1}^N \tilde{v}_i |f_i\rangle \quad (1.43)$$

donde las componentes v_i y \tilde{v}_i deben ser distintas pero estar relacionadas. Es decir, las componentes de los vectores dependen de la base elegida y tiene una *regla de cambio* cuando decidimos escribir nuestro vector en otra base.

Para encontrar esta relación entre los elementos del vector en distintas bases tenemos que primero encontrar la relación entre las bases. Como ya dijimos, cualquier vector se puede escribir como combinación lineal de elementos de una base, sin ninguna excepción. Es decir, como caso particular tenemos que cualquier elemento (vector) de una base se puede expresar como una combinación lineal de elementos de la otra. Tomemos por ejemplo, $|v\rangle = |f_1\rangle$

$$|f_1\rangle = \sum_{i=1}^N A_{i1} |e_i\rangle \quad (1.44)$$

donde $A_{i1} \in \mathbb{C}$ son las componentes del vector $|f_1\rangle$ en la base $\{|e_i\rangle\}$. Haciendo esto para todos los elementos de la base $\{|f_j\rangle\}$, los coeficientes constituyen una **matriz de cambio de base** A_{ij} :

$$|f_j\rangle = \sum_{i=1}^N A_{ij} |e_i\rangle , \quad j = 1, \dots, N \quad (1.45)$$

Nota

La forma en que están sumados los índices $|f_j\rangle = \sum_{i=1}^N A_{ij} |e_i\rangle$.

Dado un vector $|v\rangle$ con coordenadas (v_1, \dots, v_N) en la base $\{|e_i\rangle\}$ y coordenadas $(\tilde{v}_1, \dots, \tilde{v}_N)$ en la base $\{|f_i\rangle\}$, es decir,

$$|v\rangle = \sum_{i=1}^N v_i |e_i\rangle = \sum_{i=1}^N \tilde{v}_i |f_i\rangle \quad (1.46)$$

la fórmula de cambio de base expresa las coordenadas sobre la base antigua, $\{|e_i\rangle\}$, en términos de las coordenadas sobre la base nueva, $\{|f_i\rangle\}$:

$$v_i = \sum_{j=1}^N A_{ij} \tilde{v}_j \quad (1.47)$$

Nota

La forma en que están sumados los índices $v_i = \sum_{j=1}^N A_{ij} \tilde{v}_j$.

En términos de matrices, la formula de cambio de base es

$$|v\rangle_{\{|e_i\rangle\}} = A |v\rangle_{\{|f_i\rangle\}} \quad (1.48)$$

donde $|v\rangle_{\{|e_i\rangle\}}$ y $|v\rangle_{\{|f_i\rangle\}}$ son los vectores columna con las coordenadas de $|v\rangle$ en las bases $\{|e_i\rangle\}$ y $\{|f_i\rangle\}$ respectivamente.

Demostración: Usando definición (1.45) de la matriz de cambio de base, tenemos

$$\begin{aligned} |v\rangle &= \sum_{j=1}^N \tilde{v}_j |f_j\rangle \\ &= \sum_{j=1}^N \tilde{v}_j \left(\sum_{i=1}^N A_{ij} |e_i\rangle \right) \\ &= \sum_{i=1}^N \left(\sum_{j=1}^N A_{ij} \tilde{v}_j \right) |e_i\rangle \end{aligned}$$

Como $|v\rangle = \sum_{i=1}^N v_i |e_i\rangle$, la Ec. (1.47) se demuestra debido a la unicidad de la descomposición de un vector sobre una base. ■

Por supuesto, el cambio de base de la Ec. (1.47) se puede **invertir** para obtener las coordenadas sobre la base nueva, $\{|f_i\rangle\}$, en términos de las coordenadas sobre la base vieja, $\{|e_i\rangle\}$:

$$\tilde{v}_i = \sum_{j=1}^N A_{ij}^{-1} v_j , \quad |v\rangle_{\{|f_i\rangle\}} = A^{-1} |v\rangle_{\{|e_i\rangle\}} \quad (1.49)$$

Es más, que el cambio de base sea invertible es un **requisito**. Si no lo es, no es un cambio de base.

Nota: Los elementos de la base expresados en la propia base

Quizás la siguiente afirmación parezca contradictoria, pero veremos que no: dadas dos bases $\{|e_i\rangle\}$ y $\{|f_i\rangle\}$, los elementos de la base $\{|e_i\rangle\}$ expresados en la base $\{|e_i\rangle\}$ toman la forma

$$|e_1\rangle \sim \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad |e_2\rangle \sim \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \quad \dots \quad |e_N\rangle \sim \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

mientras que los elementos de la base $\{|f_i\rangle\}$ expresados en la base $\{|f_i\rangle\}$ toman la forma

$$|f_1\rangle \sim \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad |f_2\rangle \sim \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \quad \dots \quad |f_N\rangle \sim \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

La pregunta ahora es, ¿dónde está el truco? Lo engañoso aquí es la notación de los vectores como columnas de números. Como ya comentamos, un vector es un elemento abstracto que toma la forma de una columna de números complejos **cuando especificamos una base**. Es decir, cuando escribimos un vector como una columna de números, esa columna está expresada en una base de la forma (1.41). Es decir, si tenemos un vector $|u\rangle$ y trabajamos, por ejemplo, en la base $\{|e_i\rangle\}$ tenemos

$$|u\rangle = u_1 |e_1\rangle + u_2 |e_2\rangle + \dots + u_N |e_N\rangle = \sum_{i=1}^N u_i |e_i\rangle = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix}_{\{|e_i\rangle\}}$$

mientras que si trabajamos en la base $\{|f_i\rangle\}$ tenemos

$$|u\rangle = \tilde{u}_1 |f_1\rangle + \tilde{u}_2 |f_2\rangle + \dots + \tilde{u}_N |f_N\rangle = \sum_{i=1}^N \tilde{u}_i |f_i\rangle = \begin{bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ \vdots \\ \tilde{u}_N \end{bmatrix}_{\{|f_i\rangle\}}$$

Es decir, cuando escribimos los elementos de una base como vectores columna **en su propia base**, siempre toman la forma de vectores con todo 0's menos un 1.

Nota

Complementando a la nota anterior, cuando elegimos una base y decimos, por ejemplo, que nuestra base es

$$|f_1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad |f_2\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

lo que estamos haciendo al escribirlos como vectores columna es expresar la nueva base $\{|f_i\rangle\}$, en la base

$$|e_1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |e_2\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Es decir, siempre que escribimos un vector como una columna de números, esa columna está expresada en una base de la forma (1.41). Si queremos pasar a trabajar en la base $\{|f_i\rangle\}$ debemos transformar todos los vectores para expresarlos en la base donde $|f_1\rangle$ y $|f_2\rangle$ toman la forma

$$|f_1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |f_2\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Después de aplicar esta transformación (el cambio de base), los vectores $|e_1\rangle$ y $|e_2\rangle$ dejarán de tener la forma anterior.

Ejemplo

Cuando escribimos la base nueva en componentes, automáticamente estamos dando el cambio de base: Por ejemplo, sea una nueva base $\{|f_1\rangle, |f_2\rangle\}$ definida en términos de la antigua mediante

$$|f_1\rangle = \frac{1}{\sqrt{2}} (|e_1\rangle + i|e_2\rangle), \quad |f_2\rangle = \frac{1}{\sqrt{2}} (|e_1\rangle - i|e_2\rangle). \quad (1.50)$$

En componentes, esto quiere decir que

$$|f_1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix}, \quad |f_2\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix}. \quad (1.51)$$

Poniendo las dos columnas en una sola matriz, obtenemos

$$A_{ij} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix}, \quad (1.52)$$

que es, efectivamente, la matriz que efectúa el cambio de los vectores de la base

$$|f_j\rangle = \sum_{i=1}^N A_{ij} |e_i\rangle, \quad (1.53)$$

así como de las componentes de los nuevos vectores en la antigua base

$$|f_1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot |f_2\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (1.54)$$

Nota

Aquí con “en componentes” nos referimos a las componentes respecto a la base antigua, $\{|e_i\rangle\}$. Es decir, todos los vectores de este ejemplo están escritos en la base $\{|e_i\rangle\}$:

$$\begin{bmatrix} a \\ b \end{bmatrix} = a|e_1\rangle + b|e_2\rangle \quad (1.55)$$

1.2.3. Espacios de Hilbert

Definición 6 Un espacio de Hilbert, \mathcal{H} , es un espacio vectorial (real o complejo) dotado de una operación interna denominada **producto escalar**.

En matemáticas, los espacios de Hilbert (llamados así por David Hilbert) permiten generalizar los métodos del álgebra lineal y el cálculo de *espacios vectoriales euclidianos* (de dimensiones finitas) a

espacios que pueden ser de dimensiones infinitas. Los espacios de Hilbert surgen de forma natural y frecuente en matemáticas y física, normalmente como espacios de funciones. Formalmente, un espacio de Hilbert es un espacio vectorial equipado con un producto interior que induce una función de **distancia** según la cual el espacio es un espacio métrico completo.

1.2.3.1. Producto escalar

Definición 7 El **producto escalar** de dos vectores $|u\rangle$ y $|v\rangle$ es un número complejo $a \in \mathbb{C}$ que denotamos con un **braket**, $a \equiv \langle u|v\rangle$. El producto escalar verifica las tres propiedades siguientes

- **Linealidad:**

$$\langle u| (a|v\rangle + b|w\rangle) = a\langle u|v\rangle + b\langle u|w\rangle \quad (1.56)$$

para cualquier $a, b \in \mathbb{C}$ y cualquier vector $|u\rangle$, $|v\rangle$ y $|w\rangle$.

- **Hermiticidad:**

$$\langle v|u\rangle = \langle u|v\rangle^* \quad (1.57)$$

- **Positividad:**

$$\langle u|u\rangle > 0 \text{ para todo ket } |u\rangle \neq 0 \quad (1.58)$$

Combinando las dos primeras propiedades, el producto escalar también es lineal en el primer argumento

$$(a\langle u| + b\langle w|)|v\rangle = a\langle u|v\rangle + b\langle w|v\rangle, \quad (1.59)$$

es decir, el producto escalar es **bilineal**.

1.2.3.2. Norma

La positividad del producto escalar de un vector por sí mismo permite definir su **norma**:

$$\| |v\rangle \| = \sqrt{\langle v|v\rangle} \quad (1.60)$$

Nota

Véase que:

- En contraste con la definición de producto escalar en espacios vectoriales reales, en el caso complejo se hace necesario conjugar (usar un *bra*), para que la **norma** de un vector sea siempre real y positiva. Esta es la idea detrás de la definición de la *conjugación adjunta*.
- El único vector que tiene norma nula en un espacio de Hilbert es el elemento neutro

$$\langle v|v\rangle = 0 \Leftrightarrow |v\rangle = 0 \quad (1.61)$$

Ejemplo: espacio Euclídeo

Uno de los ejemplos más conocidos de espacio de Hilbert es el **espacio vectorial euclídeo** (es el espacio real de tres dimensiones, \mathbb{R}^3 , habitual, el clásico de geometría) formado por vectores tridimensionales, denotado por \mathbb{R}^3 , y dotado del **producto punto**. El producto punto toma dos vectores \vec{x} e \vec{y} y produce un número real $\vec{x} \cdot \vec{y}$. Si \vec{x} e \vec{y} se representan en coordenadas cartesianas, el producto punto se define como

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = x_1y_1 + x_2y_2 + x_3y_3$$

El producto punto satisface las siguientes propiedades:

- Es lineal en su primer argumento: $(a\vec{x}_1 + b\vec{x}_2) \cdot \vec{y} = a(\vec{x}_1 \cdot \vec{y}) + b(\vec{x}_2 \cdot \vec{y})$, para cualquier

- escalares a, b y vectores \vec{x}_1, \vec{x}_2 y \vec{y} .
- Es simétrico: $\vec{x} \cdot \vec{y} = \vec{y} \cdot \vec{x}$
 - Es definido positivo: para todo vector \vec{x} , $\vec{x} \cdot \vec{x} \geq 0$, cuya igualdad solo se satisface si $\vec{x} = 0$.

1.2.3.3. Desigualdades

En muchas ocasiones será necesario acotar cantidades.

Desigualdad de Cauchy-Schwarz:

$$|\langle u|v \rangle| \leq \| |u\rangle\| \| |v\rangle\| \quad (1.62)$$

Una consecuencia inmediata de la desigualdad de Cauchy-Schwarz es la desigualdad triangular

Desigualdad triangular:

$$\| |u\rangle + |v\rangle \| \leq \| |u\rangle\| + \| |v\rangle\| \quad (1.63)$$

1.2.4. Bases ortogonales

Hasta ahora, a los vectores de una base $\{|e_i\rangle\}$ sólo se les ha pedido que sean N vectores *linealmente independientes*, donde N es la dimensión del espacio vectorial V . En un espacio de Hilbert \mathcal{H} tiene sentido calcular el producto escalar de dos elementos de una base.

1.2.4.1. Base ortonormal

Una base **ortonormal** se caracteriza por la siguiente lista de productos escalares

$$\langle e_i | e_j \rangle = \delta_{ij} \quad (1.64)$$

Es decir:

- Por un lado, dos elementos distintos de la base son ortogonales $\langle e_1 | e_2 \rangle = 0$.
- Por otro, todos están normalizados $\|e_i\| = \sqrt{\langle e_i | e_i \rangle} = \sqrt{1} = 1$.

En este curso siempre supondremos que las bases con las que trabajamos son ortonormales. Ello se justifica en base al siguiente teorema:

Teorema 2 *Dada una base general $\{\langle f_i | f_j \rangle \neq \delta_{ij}\}$ de vectores no ortonormales, existe una procedimiento iterativo (de Gramm-Schmidt [5]) para construir, a partir de ella, una nueva base ortonormal $\{\langle e_i | e_j \rangle\} = \delta_{ij}$.*

- Dado un vector $|v\rangle = \sum_{i=1}^N v_i |e_i\rangle$ donde $|e_i\rangle$ es una base ortonormal, la componente v_k se extrae mediante la **proyección** ortogonal

$$v_k = \langle e_k | v \rangle \quad (1.65)$$

Demostración:

$$\begin{aligned}
 \langle e_k | v \rangle &= \langle e_k | \left(\sum_{j=1}^N v_j | e_j \rangle \right) \\
 &= \sum_{j=1}^N v_j \langle e_k | e_j \rangle \\
 &= \sum_{j=1}^N v_j \delta_{kj} = v_k
 \end{aligned}$$

■

- Calcular el valor de un *producto escalar* $a = \langle u | v \rangle$ es muy simple si conocemos las componentes de $|u\rangle$ y $|v\rangle$ en una base ortonormal:

$$\begin{aligned}
 a = \langle u | v \rangle &= \left(\sum_i u_i^* \langle e_i | \right) \left(\sum_j v_j | e_j \rangle \right) = \sum_{ij} u_i^* v_j \langle e_i | e_j \rangle = \sum_{ij} u_i^* v_j \delta_{ij} = \sum_i u_i^* v_i \Rightarrow \\
 \Rightarrow a = \langle u | v \rangle &= [u_1^* \quad u_2^* \quad \cdots \quad u_N^*] \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix} \tag{1.66}
 \end{aligned}$$

Nota: Importante

La expresión de la izquierda $a = \langle u | v \rangle$ **no hace referencia a ninguna base**. Por tanto, el resultado $\sum_{i=1}^n u_i^* v_i$ debe ser independiente de la base que utilizamos para representar estos vectores mediante sus componentes u_i y v_i .

Este resultado es *no trivial* y subrayamos su importancia: $\langle u | v \rangle$ puede ser calculado en la base más conveniente.

1.3. Operadores

1.3.1. Operadores y matrices

En un espacio vectorial, además de los vectores, será esencial entender la manera en que estos se pueden transformar entre sí. Un tipo de transformaciones es aquella dada por un *operador lineal*:

Definición 8 *Un operador lineal es una aplicación que transforma un vector en otro*

$$A : |u\rangle \rightarrow |v\rangle \tag{1.67}$$

de forma lineal, esto es que sobre una combinación lineal de vectores actúa de forma lineal. Es decir, para todo $\alpha, \beta \in \mathbb{C}$:

$$A : (\alpha |u\rangle + \beta |w\rangle) \rightarrow |v\rangle = \alpha A |u\rangle + \beta A |w\rangle \tag{1.68}$$

Podemos escribir también $|v\rangle = A |u\rangle \equiv |Au\rangle$ (donde Au debe entenderse como una etiqueta).

Nota

Con números (reales o complejos), un ejemplo de una operación que no es lineal es *elevar a una potencia*, por ejemplo, al cuadrado:

$$f(x) = x^2 \Rightarrow f(a+b) = (a+b)^2 \neq a^2 + b^2 \quad (1.69)$$

Nota: El cambio en física

Podemos ver la física como *el arte de estudiar el cambio*, es decir, estudiar como varía un sistema con el tiempo. Como acabamos de comentar, los *operadores lineales* nos introducen una noción de cambio (de evolución): un vector se transforma (evoluciona) a otro. Ya adelantamos aquí que en Mecánica Cuántica los vectores representan **estados** del sistema, con lo cual, el hecho de aplicar un operador sobre un vector implica pasar de un estado del sistema a otro.

Existen transformaciones inducidas por operadores *no lineales*. Sin embargo, **la Mecánica Cuántica es intrínsecamente lineal**: todas las evoluciones se dan por medio de operadores lineales.

Ejemplo

Un *operador* fácil de visualizar es el operador de **rotación en un plano**. Dado un ángulo $\theta \in (0, 2\pi)$ el operador $A = R(\theta)$ gira cualquier vector un ángulo θ en el sentido antihorario.

Un vector en el plano $\vec{u} = (u_1, u_2)$ es equivalente al número complejo $u = u_1 + iu_2$ en el plano complejo $V = \mathbb{C}$.

Escrito en polares, $u = |u|e^{i\phi}$, y sabemos que una rotación de ángulo θ es equivalente a añadirle dicho ángulo a la fase

$$v = R(\theta)u = |u|e^{i(\phi+\theta)} = |u|e^{i\phi}e^{i\theta} = u \cdot e^{i\theta} \quad (1.70)$$

Por tanto, para rotar un número complejo un ángulo θ basta con multiplicarlo por la fase $e^{i\theta}$, que se corresponde con el operador $R(\theta)$ en el espacio vectorial $V = \mathbb{C}$.

La propiedad fundamental de una rotación es la de mantener invariante el módulo $|v| = |u|$.

1.3.1.1. Matriz de un operador

Dada una base, un vector queda especificado por una colección de números, sus *componentes*. Igualmente, un operador queda definido por una *matriz numérica*.

Efectivamente, en una base, la relación $|v\rangle = A|u\rangle$ equivale a una ecuación que relacione las componentes de ambos vectores

$$v_i = \sum_{j=1}^N A_{ij} u_j. \quad (1.71)$$

Esta operación se corresponde con la siguiente composición de matrices

$$\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NN} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} \quad (1.72)$$

Ejemplo

Continuando con el ejemplo del operador de rotación en un plano, hemos visto que las componentes de $u = u_1 + iu_2$ y las de $R(\theta)u = v = v_1 + iv_2$ se obtienen mediante la multiplicación por una fase pura

$$v = ue^{i\theta}$$

Vamos a desarrollar cada miembro en cartesianas, separando las partes real e imaginaria

$$\begin{aligned} v &= v_1 + iv_2 = ue^{i\theta} = (u_1 + iu_2)(\cos \theta + i \sin \theta) \\ &= (\cos \theta u_1 - \sin \theta u_2) + i(\sin \theta u_1 + \cos \theta u_2) \end{aligned}$$

es decir las coordenadas del vector origen y el vector rotado imagen se relacionan en la forma

$$v_1 = \cos \theta u_1 - \sin \theta u_2 \quad , \quad v_2 = \sin \theta u_1 + \cos \theta u_2$$

que podemos expresar en forma matricial

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

1.3.2. Producto externo

1.3.2.1. Producto externo de vectores

Consideremos dos vectores $|u\rangle$ y $|v\rangle$. Dependiendo del orden en que los compongamos, $\langle u|v\rangle$ o $|v\rangle\langle u|$, el resultado produce dos cantidades muy distintas.

- El **producto interno**, o *producto escalar* es un **número complejo**

$$a = \langle u|v\rangle = \langle v|u\rangle^* \tag{1.73}$$

- El **producto externo** es un **operador**

$$A = |v\rangle\langle u| \tag{1.74}$$

Para comprender por qué es un operador, observamos que dicha expresión aplicada a un vector $|w\rangle$ da otro, paralelo a $|v\rangle$

$$A : |w\rangle \rightarrow A|w\rangle = |v\rangle\langle u|w\rangle = |v\rangle b = b|v\rangle \tag{1.75}$$

El número complejo $b = \langle u|w\rangle$ es la *proyección* de $|w\rangle$ en la dirección de $|u\rangle$.

Nota

Véase que:

1. El *orden* en que escribimos las cosas es *muy* relevante.
 - $\langle u|v\rangle$ y $|v\rangle\langle u|$ son objetos *radicalmente distintos*: el primero es un número y el segundo es un operador. Decimos que los vectores no commutan.
 - En cambio $|v\rangle b = b|v\rangle$, así como $\langle u|b = b\langle u|$, es decir, los números complejos y los *kets* o *bras* pueden escribirse en cualquier orden (decimos que commutan).
2. La acción del operador $A = |v\rangle\langle u|$ es muy fácil de expresar con palabras: el operador A toma *cualquier vector* $|w\rangle$ y lo convierte en un vector *paralelo* a $|v\rangle$ proporcionalmente a su proyección $b = \langle u|w\rangle$.

Si la proyección es nula $b = 0$, el operador aniquila, es decir, da el elemento neutro.

1.3.2.2. Producto externo de vectores (en componentes)

La diferencia entre el *producto interno* $a = \langle u|v \rangle$ y el *externo* $A = |u\rangle\langle v|$ tiene su reflejo. Expresando ambos vectores en la misma base ortonormal, $|u\rangle = \sum_i u_i |e_i\rangle$ y $|v\rangle = \sum_j v_j |e_j\rangle$:

- El *número complejo* a es el *producto escalar*

$$a = \langle u|v \rangle = [u_1^*, \dots, u_N^*] \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix} = \sum_i u_i^* v_i \quad (1.76)$$

- La matriz A_{ij} representa el operador A en la base $\{|e_i\rangle\}$

$$A = |v\rangle\langle u| \sim \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix} [u_1^*, \dots, u_N^*] = \begin{bmatrix} v_1 u_1^* & v_1 u_2^* & \dots & v_1 u_N^* \\ v_2 u_1^* & v_2 u_2^* & \dots & v_2 u_N^* \\ \vdots & \vdots & \ddots & \vdots \\ v_N u_1^* & \dots & v_N u_N^* \end{bmatrix} = A_{ij} \quad (1.77)$$

1.3.3. Base canónica de operadores

1.3.3.1. Producto externo de operadores de la base

Un caso importante es cuando A es el producto externo de *dos elementos de la base ortonormal* $|i\rangle$ y $|j\rangle$

$$A = |i\rangle\langle j| \quad (1.78)$$

La acción de A sobre un vector $|k\rangle$ arbitrario de la base es sencilla: cambia el vector $|j\rangle \rightarrow |i\rangle$ y aniquila a todos los demás

$$A|k\rangle = |i\rangle\langle j|k\rangle = |i\rangle \delta_{jk} = \begin{cases} 0 & \text{if } k \neq j \\ |i\rangle & \text{if } k = j \end{cases} \quad (1.79)$$

La matriz asociada al operador tiene sólo un 1 en el elemento (ij) y cero en todos los demás. Por ejemplo, supongamos que $N = 4$

$$|2\rangle\langle 3| \rightarrow \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow A_{ij} = \delta_{i2}\delta_{j3} \quad (1.80)$$

Nota: Cambio de notación en la base

Véase que en esta sección se han denominado a los elementos de la base simplemente como $|i\rangle$ en vez de como $|e_i\rangle$. Esto es simplemente un cambio de notación. Las dos notaciones son habituales en matemáticas y física, pero la usada en computación cuántica es la primera, $|i\rangle$.

1.3.3.2. Expansión de un operador general

Ahora podemos obtener una nueva perspectiva sobre la matriz A_{ij} asociada a un operador A .

De la misma manera que las componentes u_i expresan la *expansión de un vector* $|u\rangle$ en una base ortonormal de vectores, $\rightarrow |u\rangle = \sum_{i=1}^N u_i |i\rangle$, los *elementos de matriz* A_{ij} expresan la *expansión*

de un operador en una **base de operadores** $|i\rangle\langle j|$

$$A = \sum_{i,j=1}^N A_{ij} |i\rangle\langle j| \quad (1.81)$$

Esta base es la **base canónica**.

Escribiendo las matrices asociadas a $|i\rangle\langle j|$, es evidente que $\sum_{i,j=1}^N A_{ij} |i\rangle\langle j|$ reconstruye la matriz A_{ij} asociada a A

$$A = \sum_{i,j=1}^N A_{ij} |i\rangle\langle j| \iff \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NN} \end{bmatrix}. \quad (1.82)$$

1.3.3.3. Elementos de matriz

De la misma manera que obteníamos las componentes de un vector proyectando sobre un elemento de la base

$$v_i = \langle i | v \rangle \quad (1.83)$$

ahora podemos obtener los *elementos de matriz* de un operador A en la forma

$$A_{ij} = \langle i | A | j \rangle \quad (1.84)$$

Ejercicio 2 Demuestra la expresión $A_{ij} = \langle i | A | j \rangle$. Para ello usa la Ec. (1.81)

Resumen

Dada una base $\{|e_i\rangle\}$ podemos expresar un operador mediante una matriz A_{ij} . La relación concreta es

- Como operador $\rightarrow A = \sum_{ij} A_{ij} |i\rangle\langle j|$
- Como elemento de matriz $\rightarrow A_{ij} = \langle i | A | j \rangle$

1.3.3.4. Cambios de base

Sean dos bases ortonormales relacionadas mediante el cambio unitario $|j\rangle \rightarrow |\tilde{j}\rangle = \sum_{ij} U_{ij} |i\rangle$. En cada una de ellas, un operador adquiere una forma matricial concreta

$$A_{ij} = \langle i | A | j \rangle, \quad \tilde{A}_{ij} = \langle \tilde{i} | A | \tilde{j} \rangle. \quad (1.85)$$

Ambas se relacionan muy sencillamente

$$\tilde{A}_{ij} = \langle \tilde{i} | A | \tilde{j} \rangle = \sum_{k,l} \langle e_k | U_{ki}^* A U_{lj} | e_l \rangle = \sum_{k,l} U_{lj} \langle e_k | A | e_l \rangle U_{ik}^\dagger = \sum_{k,l} U_{ik}^\dagger A_{kl} U_{lj}. \quad (1.86)$$

Lema 1 Bajo un cambio de base $|e_j\rangle \rightarrow |\tilde{e}_j\rangle = \sum_{ij} U_{ij} |e_i\rangle$ las matrices asociadas a un operador A cambian según la regla

$$\tilde{A}_{ij} = (U^\dagger A U)_{ij} \quad (1.87)$$

1.3.3.5. Relación de completitud

La acción del operador identidad es

$$I |v\rangle = |v\rangle \quad (1.88)$$

En particular sobre todo elemento de la base $I |i\rangle = |i\rangle$. En otras palabras, el operador identidad I tiene por matriz $I_{ij} = \delta_{ij}$ = diagonal $(1, 1, \dots, 1)$ con lo que

$$\boxed{I = \sum_i |i\rangle\langle i|} = \sum_{ij} \delta_{ij} |i\rangle\langle j| = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \quad (1.89)$$

Esta expresión se conoce también como *relación de completitud* o, también **relación de cierre** y se utiliza muy frecuentemente.

La relación de completitud es, en realidad, una propiedad de **cualquier base**. Dicho de otro modo, si $\{|e_i\rangle\}$ y $\{|f_i\rangle\}$ son, ambas, bases entonces $I |e_i\rangle = |e_i\rangle$ y $I |f_j\rangle = |f_j\rangle$, entonces

$$I = \sum_i |e_i\rangle\langle e_i| = \sum_j |f_j\rangle\langle f_j|. \quad (1.90)$$

La relación de cierre, o completitud, siempre se puede insertar en cualquier momento del cálculo. Se utiliza con frecuencia para efectuar cambios de base.

Por ejemplo, vamos a ver que el producto escalar $\langle u|v\rangle$ puede calcularse en cualquier. Sea $|u\rangle = \sum_i u_i |e_i\rangle = \sum_i \tilde{u}_i |f_i\rangle$ y $|v\rangle = \sum_i v_i |e_i\rangle = \sum_i \tilde{v}_i |f_i\rangle$. Entonces

$$\langle v|u\rangle = \langle v| I |u\rangle = \langle v| \left(\sum_i |e_i\rangle\langle e_i| \right) |u\rangle = \sum_i \langle v|e_i\rangle\langle e_i|u\rangle = \sum_i v_i^* u_i \quad (1.91)$$

$$\langle v|u\rangle = \langle v| I |u\rangle = \langle v| \left(\sum_i |f_i\rangle\langle f_i| \right) |u\rangle = \sum_i \langle v|f_i\rangle\langle f_i|u\rangle = \sum_i \tilde{v}_i^* \tilde{u}_i \quad (1.92)$$

1.3.4. El espacio vectorial $L(\mathcal{H})$

El *conjunto de todos* los *operadores lineales* sobre un espacio vectorial \mathcal{H} tiene, de forma natural, una estructura de espacio vectorial que denominamos $L(\mathcal{H})$.

En efecto, sean A y B dos operadores, y $\lambda \in \mathbb{C}$ un número complejo. Tanto la suma $C = A + B$ como la multiplicación externa $D = \lambda A$ son nuevos operadores definidos por su acción sobre un vector cualquiera $|v\rangle \in \mathcal{H}$

$$C |v\rangle = (A + B) |v\rangle = A |v\rangle + B |v\rangle \quad (1.93)$$

$$D |v\rangle = (\lambda A) |v\rangle = \lambda(A |v\rangle) \quad (1.94)$$

1.3.4.1. Operador Adjunto

La conjugación *adjunta* se puede extender a $L(\mathcal{H})$

$$\dagger \rightarrow \begin{cases} z & \leftrightarrow z^* \\ |u\rangle & \leftrightarrow \langle u| \\ A & \leftrightarrow A^\dagger \end{cases} \quad (1.95)$$

pero añadiendo dos reglas que permiten aplicar \dagger a sumas y productos de *objetos* $a, b \in \{z, |u\rangle, A\}$

- **linealidad** $(a + b)^\dagger = a^\dagger + b^\dagger$
- **trasposición** $(ab)^\dagger = b^\dagger a^\dagger$ (sólo relevante cuando a y b no comuten)

Ejemplos

- $|v\rangle = A|u\rangle \Leftrightarrow \langle v| = \langle u|A^\dagger$ donde el operador en la derecha actúa sobre el *bra* a su izquierda. Notar que, como $|v\rangle^\dagger = |Au\rangle^\dagger = \langle Au|$, la ecuación anterior implica

$$\langle Au| = \langle u|A^\dagger \quad (1.96)$$

- $\langle w|A|u\rangle^* = (\langle w|A|u\rangle)^\dagger = \langle u|A^\dagger|w\rangle$

1.3.4.2. Matriz Adjunta

Estas reglas nos permiten obtener el adjunto de un operador

$$A^\dagger = \sum_{ij} (A_{ij}|i\rangle\langle j|)^\dagger = \sum_{ij} |j\rangle\langle i|A_{ij}^* = \sum_{ji} A_{ji}^*|i\rangle\langle j| \quad (1.97)$$

donde en la última ecuación hemos intercambiado los nombres $i \leftrightarrow j$.

Vemos que la matriz que representa A^\dagger es la *matriz adjunta* de A_{ij} , es decir, la traspuesta y conjugada

$$(A^\dagger)_{ij} = A_{ji}^* = (A_{ij}^*)^t \equiv (A_{ij})^\dagger \quad (1.98)$$

donde † significa el adjunto de un operador a la izquierda, y de una matriz a la derecha.

1.3.4.2.1. Dimensión de $L(\mathcal{H})$

Si \mathcal{H} tiene dimensión N , un *operador general* $A \in L(\mathcal{H})$ se especifica mediante una matriz de N^2 números complejos $\Rightarrow A = A_{ij}|e_i\rangle\langle e_j|$. Además, N^2 números complejos equivalen a $2N^2$ números reales.

En otras palabras: A tiene N^2 grados de libertad complejos y, por tanto, ésta es la dimensión del espacio $L(\mathcal{H})$

$$\dim_{\mathbf{C}}(L(\mathcal{H})) = N^2 \iff \dim_{\mathbf{R}}(L(\mathcal{H})) = 2N^2 \quad (1.99)$$

1.3.5. Clases de operadores

Vamos a considerar clases de operadores que satisfagan algún tipo de *condición* o *restricción*.

1.3.5.1. Operador Unitario

Definición 9 Un *operador unitario* U es tal que su adjunto es igual a su inverso. Es decir verifica la siguiente ecuación

$$U^\dagger = U^{-1} \quad (1.100)$$

Naturalmente, esta ecuación se traduce en la misma ecuación para las matrices asociadas

$$(U_{ij})^\dagger = U_{ji}^* = U_{ij}^{-1} \quad (1.101)$$

Veamos ahora por qué hemos definido esta clase de operadores.

Teorema 3 La acción de un operador unitario conserva **intacto el producto escalar** de dos vectores cualesquiera. En particular, conserva también la norma de cualquier vector.

Demostración: Sea U un operador unitario, y $|\varphi'\rangle = U|\varphi\rangle$ y $|\psi'\rangle = U|\psi\rangle$ dos vectores transformados por U , entonces

$$\langle\varphi'|\psi'\rangle = (\langle\varphi|U^\dagger)U|\psi\rangle = \langle\varphi|U^\dagger U|\psi\rangle = \langle\varphi|\psi\rangle \quad (1.102)$$

particularizando para $|\varphi\rangle = |\psi\rangle$ tenemos que un operador unitario conserva la norma.

$$\|U|\varphi\rangle\| = \||\varphi\rangle\| \quad (1.103)$$

■

Lemma 2 La composición de dos operadores U y V unitarios es unitaria. Sin embargo, la combinación lineal de operadores unitarios no es unitaria.

Ejercicio 3 Demuestra este resultado, es decir, demuestra que:

$$\begin{aligned} (UV)^\dagger &= (UV)^{-1} \\ (U + V)^\dagger &\neq (U + V)^{-1} \end{aligned}$$

Por tanto, los operadores unitarios *no forman* un subespacio vectorial dentro de $L(\mathcal{H})$. La estructura matemática que forman se denomina **grupo**: el grupo unitario $U(m)$, donde m es la dimensión del espacio de Hilbert \mathcal{H} .

Aun así, forman una *variedad*: un conjunto continuo que se puede parametrizar mediante una colección de parámetros, la *dimensión de la variedad*. Como hay una relación 1 a 1 entre un operador una matriz (en una base), esa dimensión será igual a la *dimensión del conjunto de matrices unitarias*.

Ejercicio 4 Resta de $\dim_{\mathbf{R}}(L(\mathcal{H})) = 2N^2$ el número de ecuaciones que restringen la matriz de un operador unitario y halla así la dimensión (real) de la variedad de operadores unitarios.

1.3.5.2. Operador Unitario sobre bases ortonormales

Como caso particular, aplicando un operador unitario U a una base ortonormal $\{|e_i\rangle\}$ obtenemos otra base ortonormal $\{|f_i\rangle\}$

$$\left. \begin{array}{l} |f_i\rangle = U|e_i\rangle \\ U^{-1} = U^\dagger \end{array} \right\} \iff \langle f_i|f_j\rangle = \langle e_i|e_j\rangle = \delta_{ij} \quad (1.104)$$

Inversamente, dadas dos bases ortonormales, $\{|e_i\rangle\}$ y $\{|f_i\rangle\}$, el operador que las relaciona es un operador unitario

$$\begin{aligned} U &= \sum_i |f_i\rangle\langle e_i| \Rightarrow U|e_j\rangle = |f_j\rangle \\ U^\dagger &= \sum_i |e_i\rangle\langle f_i| \Rightarrow U^\dagger|f_j\rangle = |e_j\rangle \end{aligned} \quad (1.105)$$

1.3.5.2.1. Operador ortogonal

Un **operador ortogonal** es un caso particular de operador unitario con *elementos de matriz reales*. El operador de rotación $R(\theta)$ que hemos estudiado al comienzo de este tema es un operador ortogonal. Es inmediato comprobar que

$$R(\theta)^t = R(-\theta) = R(\theta)^{-1} \quad (1.106)$$

1.3.5.3. Operador Normal

Definición 10 Un operador N es **normal** si conmuta con su adjunto

$$NN^\dagger = N^\dagger N \quad (1.107)$$

1.3.5.4. Operador Hermítico

Definición 11 Un operador H es **Hermítico** (o **autoadjunto**) si verifica la ecuación siguiente

$$H = H^\dagger \quad (1.108)$$

- Es evidente que un operador hermítico es un operador normal, pero a la inversa no tiene por qué.
- La **combinación lineal** de dos operadores *hermíticos* con coeficientes *reales* es un operador *hermítico*

$$C^\dagger = (aA + bB)^\dagger = a^* A^\dagger + b^* B^\dagger = aA + bB = C \quad (1.109)$$

En otras palabras, los operadores autoadjuntos forman un subespacio vectorial $\subset L(\mathcal{H})$.

- En cambio la composición (producto) de dos operadores hermíticos, en general no es hermítico

$$(AB)^\dagger = B^\dagger A^\dagger = BA \neq AB \quad (1.110)$$

salvo que A y B conmuten.

- La matriz asociada a un operador hermítico también se llama hermítica, y coincide con su traspuesta y conjugada

$$A_{ij} = A_{ij}^\dagger \equiv A_{ij}^{*t} = A_{ji}^* \quad (1.111)$$

Naturalmente, las matrices hermíticas también forman un subespacio vectorial dentro del espacio vectorial de matrices.

- A partir de cualquier operador $C \neq C^\dagger$ siempre podemos construir un operador hermítico $A = A^\dagger$ mediante la combinación lineal

$$A = C + C^\dagger \quad (1.112)$$

donde a es un número real. Esto se extiende trivialmente a las matrices que los representan en cualquier base

$$A_{ij} = C_{ij} + C_{ji}^* \quad (1.113)$$

Ejercicio 5 Resta de $2N^2$ el número de ecuaciones que restringen la matriz de un operador hermítico y halla así la dimensión del subespacio vectorial hermítico

1.3.5.5. Operador anti-Hermítico

Definición 12 Un operador A es **anti-hermítico** si verifica la ecuación siguiente

$$A = -A^\dagger \quad (1.114)$$

- Un operador anti-hermitico siempre se puede obtener de un operador hermítico mediante el número i

$$A = iH \iff A^\dagger = (iH)^\dagger = -iH \quad (1.115)$$

- Se deduce de lo anterior que un operador anti-hermítico es también un operador normal

$$AA^\dagger = A(-A) = -AA = A^\dagger A \quad (1.116)$$

1.3.5.6. Proyectores

El operador $P = |u\rangle\langle u|$ proyecta cualquier vector en la dirección de $|u\rangle$

$$P|w\rangle = |u\rangle\langle u|w\rangle = a|u\rangle \quad (1.117)$$

donde $a = \langle u|w\rangle$.

Aplicado al vector $|u\rangle$ lo deja invariante

$$P|u\rangle = |u\rangle\langle u|u\rangle = |u\rangle \quad (1.118)$$

Es por esto que, una segunda aplicación de P , no modifica el resultado

$$P(P|w\rangle) = \langle u|w\rangle P(|u\rangle) = \langle u|w\rangle|u\rangle \quad (1.119)$$

Como el vector $|w\rangle$ es arbitrario, que $P^2|w\rangle = P|w\rangle$ sean siempre iguales es una propiedad del operador P . De hecho **esta propiedad define una clase de operadores**.

Definición 13 *Un proyector es un operador hermítico que verifica la ecuación*

$$P^2 = P \quad (1.120)$$

Nota: Los proyectores son no-unitarios

El proyector es un operador *no-unitario*: la proyección reduce la norma. Supongamos que $|u\rangle$ y $|w\rangle$ son vectores unitarios y distintos

$$\|P|w\rangle\|^2 = \langle w|P^\dagger P|w\rangle = \langle w|P|w\rangle = \langle w|u\rangle\langle u|w\rangle = |\langle u|w\rangle|^2 < \| |u\rangle \| \| |w\rangle \| = 1 \quad (1.121)$$

donde hemos aplicado la desigualdad de Cauchy Schwarz estricta, al suponer que $|u\rangle \neq |w\rangle$.

■ Matriz asociada a un proyector

- Si $|u\rangle = |e_1\rangle$ el operador $P_1 = |e_1\rangle\langle e_1|$ proyecta cualquier vector sobre su componente a lo largo de $|e_1\rangle$. En forma matricial

$$|e_1\rangle\langle e_1| = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} \quad (1.122)$$

de modo que

$$|e_1\rangle\langle e_1|u\rangle = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} u^1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = u^1|e_1\rangle \quad (1.123)$$

- Si $|u\rangle = \sum_i u^i |e_i\rangle$ es un vector unitario $\| |u\rangle \| = 1$, entonces el proyector a lo largo de $|u\rangle$ viene dado por

$$P(u) = |u\rangle\langle u| = \sum_{i,j} u_i u_j^* |e_i\rangle\langle e_j| \quad (1.124)$$

Es decir, le está asociada una matriz dada por $P_{ij} = u_i u_j^*$. Es trivial verificar que

$$\sum_j P_{ij} P_{jk} = \sum_j u_i u_j^* u_j u_k^* = u_i \left(\sum_j u_j^* u_j \right) u_k = u_i u_k^* = P_{ik} \quad (1.125)$$

como corresponde a un proyector.

■ Proyectores ortogonales

Si $\{|\mu_i\rangle\}$ es un conjunto de vectores ortonormales, forman la base de un subespacio $S \subset \mathcal{H}$. El operador

$$P_S = \sum_i P(\mu_i) = \sum_i |\mu_i\rangle\langle\mu_i| \quad (1.126)$$

es un proyector sobre el subespacio S , es decir $P_S |u\rangle \in S$ para todo $|u\rangle \in \mathcal{H}$.

■ Proyector perpendicular

Cualquier vector se puede descomponer como combinación de un vector y otro perpendicular

$$|\psi\rangle = a|u\rangle + b|u_\perp\rangle \quad (1.127)$$

Si tomamos $|u\rangle$ y $|u_\perp\rangle$ unitarios, los proyectores asociados

$$P_{\parallel} = P(u) = |u\rangle\langle u| \quad , \quad P_{\perp} = P(u_\perp) = |u_\perp\rangle\langle u_\perp| \quad (1.128)$$

son perpendiculares

$$P_{\parallel} P_{\perp} = 0 \quad , \quad P_{\parallel} + P_{\perp} = I \quad (1.129)$$

1.3.5.7. Reflectores

Cualquier vector se puede descomponer como combinación de un vector y otro perpendicular

$$|\psi\rangle = a|u\rangle + b|u_\perp\rangle \quad (1.130)$$

■ Reflector paralelo a $|u\rangle$: $|u\rangle : R_u = R_{\parallel}$

El operador que *refleja paralelamente* a $|u\rangle$ será el que invierte la componente de $|\psi\rangle$ en esa dirección

$$R_{\parallel} = I - 2P_{\parallel} = I - 2|u\rangle\langle u| \quad (1.131)$$

Efectivamente

$$R_{\parallel} |\psi\rangle = -a|u\rangle + b|u_\perp\rangle . \quad (1.132)$$

■ Reflector perpendicular a $|u\rangle$: $|u\rangle : R_{u_\perp} = R_{\perp}$

El operador que *refleja perpendicularmente* a $|u\rangle$ debe invertir la componente de $|\psi\rangle$ a lo largo de $|u_\perp\rangle$

$$\begin{aligned} R_{\perp} |\psi\rangle &= a|u\rangle - b|u_\perp\rangle \\ &= -(-a|u\rangle + b|u_\perp\rangle) \\ &= -R_{\parallel} |\psi\rangle \end{aligned}$$

De modo que

$$R_{\perp} = -R_{\parallel} = 2P_{\parallel} - I , \quad (1.133)$$

Si tenemos en cuenta la relación $P_{\parallel} + P_{\perp} = I \Rightarrow P_{\parallel} = I - P_{\perp}$, podemos concluir que

$$R_{\perp} = I - 2|u_\perp\rangle\langle u_\perp| = I - 2P_{\perp} \quad (1.134)$$

1.3.6. Comutador y Traza

1.3.6.1. Comutador

A diferencia de los números, el orden en el que se componen dos operadores es relevante.

Definición 14 *Dados dos operadores, A y B , definimos el **comutador***

$$[A, B] = AB - BA. \quad (1.135)$$

El comutador tiene las dos siguientes propiedades algebraicas, elementales de probar

$$\text{Derivacion} \rightarrow [A, BC] = B[A, C] + [A, B]C$$

$$\text{Identidad de Jacobi} \rightarrow [A, [B, C]] + [B, [C, A]] + [C, [A, B]] = 0$$

La conmutatividad de dos operadores $[A, B] = 0$ es una propiedad algebraica muy deseable que, cuando se da, implica propiedades muy ventajosas.

1.3.6.2. Traza de un operador

Definición 15 *La traza de un operador se define como la suma de los elementos diagonales de la matriz que lo representa en una base*

$$\text{tr}A \equiv \sum_i A_{ii} = \sum_i \langle i | A | i \rangle \quad (1.136)$$

- La traza de A es *independiente de la base*. Por eso decimos que es la *traza del operador*.

Demostración:

$$\begin{aligned} \text{tr}A &= \sum_i \langle i | A | i \rangle = \sum_i \langle i | A \left(\sum_j |\tilde{j}\rangle \langle \tilde{j}| \right) | i \rangle \\ &= \sum_{ij} \langle i | A | \tilde{j} \rangle \langle \tilde{j} | i \rangle = \sum_{ij} \langle \tilde{j} | i \rangle \langle i | A | \tilde{j} \rangle \\ &= \sum_j \langle \tilde{j} | \left(\sum_i |i\rangle \langle i| \right) A | \tilde{j} \rangle = \sum_j \langle \tilde{j} | A | \tilde{j} \rangle \\ &= \sum_j \tilde{A}_{jj} \end{aligned}$$

■

- La traza es una operación **lineal**

$$\text{tr}(A + B) = \text{tr}A + \text{tr}B \quad (1.137)$$

- La traza de un producto de operadores es **cíclica**: es invariante bajo permutaciones cíclicas de los operadores en su argumento. Por ejemplo, para tres operadores A, B y C

$$\text{tr}(ABC) = \text{tr}(BCA) \quad (1.138)$$

Ejercicio 6 Demuestra este resultado

Para un producto de dos operadores, el anterior resultado implica que la *traza de un commutador es cero*. Dicho de otra forma

$$\text{tr}(AB) = \text{tr}(BA) \Rightarrow \text{tr}([A, B]) = 0. \quad (1.139)$$

1.3.7. Autovalores y autovectores

Definición 16 Existen vectores, $|\lambda\rangle$, para los cuales la acción de un operador A devuelve un vector **paralelo**

$$A|\lambda\rangle = \lambda|\lambda\rangle. \quad (1.140)$$

Decimos que $|\lambda\rangle$ es un vector propio (o **autovector**) de A con valor propio (o **autovalor**) asociado $\lambda \in \mathbb{C}$.

Si $|\lambda\rangle$ es un autovector, también lo es $|\mu\rangle = \alpha|\lambda\rangle$, para cualquier número complejo $\alpha \in \mathbb{C}$. En efecto

$$A|\mu\rangle = A(\alpha|\lambda\rangle) = \alpha A|\lambda\rangle = \alpha\lambda|\lambda\rangle = \lambda\alpha|\lambda\rangle = \lambda|\mu\rangle \quad (1.141)$$

Es decir, asociado a un autovalor tenemos infinitos autovectores paralelos.

1.3.7.1. Autovalores degenerados

Decimos que un autovalor λ_k es d_k veces **degenerado** si existen d_k autovectores **linealmente independientes**, $|\lambda_k^a\rangle$ con $a = 1, \dots, d_k$ asociados al **mismo** autovalor

$$A|\lambda_k^a\rangle = \lambda_k|\lambda_k^a\rangle, \quad \text{con } a = 1, \dots, d_k \quad (1.142)$$

1.3.7.2. Subespacio propio

Bajo la acción del operador, los autovalores generan un **subespacio propio** $S(\lambda_k) \in \mathcal{H}$. Esto quiere decir que aquellos vectores que pertenecen a un subespacio, siguen perteneciendo al mismo subespacio tras la aplicación del operador.

Por ejemplo, sea un operador A tal que

$$A|\lambda_k^a\rangle = \lambda_k|\lambda_k^a\rangle, \quad \text{con } a = 1, \dots, d_k \quad (1.143)$$

Podemos construir

$$|u\rangle = \sum_{a=1}^{d_k} c_a |\lambda_k^a\rangle \quad (1.144)$$

como una combinación de los autovectores asociados a un autovalor concreto λ_k . Entonces, bajo la acción del operador A tenemos

$$A|u\rangle = \sum_{a=1}^{d_k} c_a A|\lambda_k^a\rangle = \sum_{a=1}^{d_k} c_a \lambda_k |\lambda_k^a\rangle = \lambda_k \sum_{a=1}^{d_k} c_a |\lambda_k^a\rangle = \lambda_k |u\rangle$$

Por tanto $|u\rangle \in S(\lambda_k)$.

El Teorema de Gramm-Schmidt (Teorema 2) garantiza que podemos elegir (mediante un cambio adecuado) el conjunto $\{|\lambda_k^a\rangle\} \in (\lambda_k), a = 1, \dots, d_k$ de forma que sea una *base ortonormal*

$$\langle \lambda_k^a | \lambda_k^b \rangle = \delta_{ab} \quad (1.145)$$

El **proyector ortogonal** sobre el subespacio propio $S(\lambda_k)$ será

$$P_k = \sum_{a=1}^{d_k} |\lambda_k^a\rangle\langle\lambda_k^a| \quad (1.146)$$

Ejemplo

Llamemos $R_z(\theta)$ el operador que efectúa una rotación de ángulo θ entorno al eje z . Cuando $\theta = \pi$ encontramos la siguiente acción sobre los tres elementos $\{\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}\}$ de la base cartesiana

$$\begin{aligned} R_z(\pi)\hat{\mathbf{x}} &= -\hat{\mathbf{x}} \\ R_z(\pi)\hat{\mathbf{y}} &= -\hat{\mathbf{y}} \\ R_z(\pi)\hat{\mathbf{z}} &= +\hat{\mathbf{z}} \end{aligned}$$

Vemos que hay un autovector $\hat{\mathbf{z}}$ con autovalor $+1$ y dos autovectores $\hat{\mathbf{x}}$ y $\hat{\mathbf{y}}$ con autovalor -1 .

El espacio \mathbb{R}^3 se divide en dos subespacios propios de $R_z(\pi)$, uno de dimensión 1 (a lo largo del eje $\hat{\mathbf{z}}$) y otro de dimensión 2 (en el plano $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$).

Los proyectores asociados serán

$$P_{\hat{\mathbf{z}}} = |\hat{\mathbf{z}}\rangle\langle\hat{\mathbf{z}}| = \begin{bmatrix} 0 & & \\ & 0 & \\ & & 1 \end{bmatrix}, \quad P_{\hat{\mathbf{x}}\hat{\mathbf{y}}} = |\hat{\mathbf{x}}\rangle\langle\hat{\mathbf{x}}| + |\hat{\mathbf{y}}\rangle\langle\hat{\mathbf{y}}| = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 0 \end{bmatrix}, \quad (1.147)$$

1.3.7.3. Espectro de Operadores Normales

Recordemos la definición de un operador normal. N será un operador normal si commuta con su adjunto

$$NN^\dagger = N^\dagger N \quad (1.148)$$

La importancia de los operadores normales radica en el siguiente lema

Lemma 3 *Dos autovectores de un operador normal asociados a dos autovalores **distintos** son ortogonales*

$$\lambda_i \neq \lambda_j \iff \langle\lambda_i|\lambda_j\rangle = 0 \quad (1.149)$$

Demostración: De la ecuación de autovalores $N|\lambda_j\rangle = \lambda_j|\lambda_j\rangle$, y de $NN^\dagger = N^\dagger N$, se sigue que

$$\langle\lambda_j|(N^\dagger - \lambda_j^*)(N - \lambda_j)|\lambda_j\rangle = \langle\lambda_j|(N - \lambda_j)(N^\dagger - \lambda_j^*)|\lambda_j\rangle = 0 \quad (1.150)$$

de donde obtenemos $(N^\dagger - \lambda_j^*)|\lambda_j\rangle = 0 \Rightarrow \langle\lambda_j|N = \langle\lambda_j|\lambda_j$. Entonces

$$\langle\lambda_j|N|\lambda_i\rangle = \lambda_j\langle\lambda_j|\lambda_i\rangle = \lambda_i\langle\lambda_j|\lambda_i\rangle, \quad (1.151)$$

de donde se sigue que, para $\lambda_i \neq \lambda_j \Rightarrow \langle\lambda_i|\lambda_j\rangle = 0$. ■

En general, cada autovalor λ_k será $d_k \geq 1$ veces degenerado. En ese caso hay $\{|\lambda_k^a\rangle\}$, $a = 1, \dots, d_k$ autovectores que generan el subespacio propio, $S(\lambda_k) \subset \mathcal{H}$, de dimensión d_k .

Según este lemma, los subespacios $S(\lambda_k) \perp S(\lambda_j)$ son ortogonales para $k \neq j$.

En resumen: siempre podemos encontrar una base ortonormal de \mathcal{H} , formada por autovectores de un operador normal N

$$I = \sum_k \sum_{a=1}^{d_k} |\lambda_k^a\rangle\langle\lambda_k^a| \quad ; \quad \langle\lambda_j^a|\lambda_k^b\rangle = \delta_{ab}\delta_{jk} \quad (1.152)$$

El proyector sobre el subespacio propio $S(\lambda_k)$ será

$$P_k = \sum_{a=1}^{d_k} |\lambda_k^a\rangle\langle\lambda_k^a| \quad (1.153)$$

1.3.7.4. Descomposición Espectral

Teorema 4 (Teorema Espectral) Para todo operador normal N existe una base de autovectores ortonormales, $\{|\lambda_k^a\rangle\}$, tales que N admite la siguiente **descomposición espectral**

$$N = \sum_{k=1}^d \lambda_k P_k \quad (1.154)$$

donde $d = \dim(\mathcal{H})$ y $P_k = \sum_{a=1}^{d_k} |\lambda_k^a\rangle\langle\lambda_k^a|$ es el proyector sobre el subespacio propio $S(\lambda_k)$

La matriz N_{ij} que expresa N en la base $|\lambda_i\rangle$ es diagonal

$$N_{ij} = \langle\lambda_i^a|N|\lambda_j^b\rangle = \lambda_k \delta_{kj} \delta_{ab} = \begin{bmatrix} \lambda_1 & & & \\ & \ddots & & \\ & & \lambda_2 & \\ & & & \ddots \\ & & & & \lambda_N \end{bmatrix} \quad (1.155)$$

donde λ_k aparecerá d_k veces repetido.

Nota: El operador identidad

El operador identidad tiene a cualquier vector por autovector $I|v\rangle = |v\rangle$, con autovalores $\lambda_i = 1$. Por tanto, en **cualquier base**, la matriz asociada a I tiene la forma diagonal

$$I_{ij} = \delta_{ij} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \quad (1.156)$$

La descomposición espectral de I no es otra que la **relación de completitud**, que es cierta para cualquier base, ya que todas las bases son bases de autoestados de I

$$I = \sum_{i=1}^N |\lambda_i\rangle\langle\lambda_i| = \sum_{i=1}^N |e_i\rangle\langle e_i| \quad (1.157)$$

1.3.7.5. Espectro de Operadores Hermíticos

El **espectro de un operador hermítico** $A = A^\dagger$, tiene dos propiedades importantes:

1. Los **autovalores** de un operador hermítico son reales $\lambda_i \in \mathbb{R}$.
2. Los **autovectores** $|\lambda_i\rangle$ de un operador hermítico asociados a autovalores distintos son ortogonales. Es decir

$$\lambda_i \neq \lambda_j \iff \langle \lambda_i | \lambda_j \rangle = 0. \quad (1.158)$$

Demostración:

1. Tomemos un autovector normalizado de A , $|\lambda\rangle$ de autovalor λ .

$$\lambda = \langle \lambda | A | \lambda \rangle = (\langle \lambda | A^\dagger | \lambda \rangle)^* = (\langle \lambda | A | \lambda \rangle)^* = \lambda^*. \quad (1.159)$$

2. Los operadores hermíticos son también normales, así que la demostración ya la vimos para esos operadores.

■

Base ortonormal: el conjunto de autovectores $|\lambda_i\rangle$ de un operador hermítico forma una base ortonormal. Puede normalizarse para formar una base ortonormal

$$\langle \lambda_i | \lambda_j \rangle = \delta_{ij}. \quad (1.160)$$

1.3.7.6. Espectro de Operadores Unitarios

Los autovalores de un **operador unitario** son **fases puras**

$$U^\dagger = U^{-1} \iff \lambda_i = e^{i\phi_i} \quad (1.161)$$

Demostración: Es evidente puesto que sólo estos números complejos verifican que el complejo conjugado y el inverso coinciden $\lambda^* = \lambda^{-1} \Leftrightarrow \lambda = e^{i\phi}$. ■

1.3.7.7. Operadores que conmutan

Cuando dos operadores conmutan se dan ciertas propiedades algebraicas que son muy ventajosas. En cierto modo se parecen más a c-números. Veamos la primera.

Teorema 5 *Dados dos operadores A y B que conmutan, existe una base $\{|\lambda_i\rangle\}$ de autovalores simultáneos de ambos operadores, es decir*

$$A = \lambda_i^A |\lambda_i\rangle\langle\lambda_i|, \quad B = \lambda_i^B |\lambda_i\rangle\langle\lambda_i| \quad (1.162)$$

Demostración: Supongamos que A y B conmutan. Entonces la acción de A estabiliza los subespacios propios de B .

Es decir, si $|\lambda\rangle$ es autoestado de A , entonces $|\mu\rangle = B|\lambda\rangle$ también es autoestado con idéntico autovalor. Se comprueba fácilmente

$$A|\mu\rangle = A(B|\lambda\rangle) = B(A|\lambda\rangle) = B(\lambda|\lambda\rangle) = \lambda(B|\lambda\rangle) \quad (1.163)$$

Por tanto $|\lambda\rangle$ y $B|\lambda\rangle$ pertenecen al *mismo subespacio propio*. Esto es lo que se entiende por *estabilizar el subespacio*.

Si λ es degenerado esto sólo asegura que $B|\lambda\rangle = |\lambda'\rangle$ pertenece al subespacio propio del mismo autovalor λ . Esto quiere decir que, dentro de cada subespacio propio de B , podemos escoger la

base que queramos. En particular podemos escoger una base que diagonalice A dentro de dicho subespacio. ■

En otras palabras, dos operadores que comutan, son diagonalizables simultáneamente. Su matriz en la base $\{|\lambda_i\rangle\}$ es

$$A = \begin{bmatrix} \lambda_1^A & & & \\ & \lambda_2^A & & \\ & & \ddots & \\ & & & \lambda_n^A \end{bmatrix}, \quad B = \begin{bmatrix} \lambda_1^B & & & \\ & \lambda_2^B & & \\ & & \ddots & \\ & & & \lambda_n^B \end{bmatrix}. \quad (1.164)$$

1.3.8. Factorización de unitarios

Como ya vimos, fabricar un operador hermítico A a partir de otro operador B no hermítico, es fácil: $A = B + B^\dagger$. Sin embargo, fabricar operadores unitarios no es tan fácil. Veamos dos métodos para ello.

1.3.8.1. Descomposición Polar (PD)

Teorema 6 *Todo operador $A \in L(\mathcal{H})$ admite la descomposición polar $A = UR$ donde U es un operador unitario, y R es un operador semi-definido positivo (sólo tiene autovalores positivos o cero)*

- La descomposición polar es *única* y generaliza la representación polar de números complejos $z = re^{i\phi}$ a operadores.
- El hecho de que $r \geq 0$ es la contrapartida a que R sea semi-definida positiva.
- El factor $e^{i\phi}$ es análogo al hecho de que un operador unitario, como veremos, sólo tiene autovalores que son fases puras.

1.3.8.2. Descomposición en valores singulares (SVD)

Vamos a enunciar este teorema para matrices. Concretamente el teorema habla de una matriz $m \times n$. Este tipo de matrices se corresponden con operadores $O \in L(\mathcal{H}_A, \mathcal{H})$ entre espacios de dimensiones m y n .

Teorema 7 *Sea A una matriz compleja $m \times n$. Entonces admite la siguiente forma (**descomposición en valores singulares**)*

$$A = U\Sigma V^\dagger, \quad (1.165)$$

donde $U \in U(m)$, $V \in U(n)$ son matrices unitarias y Σ es una matriz $m \times n$ con $\lambda_1, \dots, \lambda_r$ valores singulares reales y positivos en la diagonal, con $r \leq \min(m, n)$.

1.3.9. Funciones de Operadores

1.3.9.1. Funciones analíticas

Estamos acostumbrados a escribir funciones *de una variable real o compleja*. Por ejemplo $f(x) = x^2$, ó, $f(z) = e^z$. Querríamos dar sentido a una función *de un operador* $A \rightarrow f(A)$

Como en operadores la operación de composición la tenemos bien definida (multiplicación de matrices), en el caso de que $f(z)$ sea una función analítica expresable como una **serie de Taylor** en torno a

$$x = 0$$

$$f(z) = \sum_{n=0}^{\infty} \frac{1}{n!} f^{(n)}(0) z^n \quad (1.166)$$

tomaremos como **definición** la *misma serie* cambiando el argumento $x \rightarrow A$

$$f(A) = \sum_{n=0}^{\infty} \frac{1}{n!} f^{(n)}(0) A^n \quad (1.167)$$

Vemos que esta definición es una definición formal, que nos puede ser útil para algunas demostraciones, pero que no es práctica, en el sentido de que no sabemos como lidar con infinitos términos. Veremos otra forma de lidar con esto en el Teorema 10.

Nota

De la misma forma que, para funciones analíticas $f(z)^* = f(z^*)$, también la definición anterior asegura que $f(A)^\dagger = f(A^\dagger)$

1.3.9.2. Exponencial de un operador

La exponencial de un operador será

$$\exp(A) = e^A = I + A + \frac{1}{2}A^2 + \frac{1}{3!}A^3 + \dots \quad (1.168)$$

Una propiedad importante de la función exponencial es $e^x e^y = e^{x+y}$. La propiedad análoga para operadores *sólo es cierta cuando comutan entre sí*. Para el caso genérico tenemos dos opciones

■ Teorema de Baker-Campbel-Haussdorf

Teorema 8 Sean $A, B \subset L(\mathcal{H})$ dos operadores lineales genéricos. Entonces

$$e^A e^B = e^{(A+B+\frac{1}{2}[A,B]+\frac{1}{12}[A,[A,B]]+\frac{1}{12}[B,[B,A]]+\dots)} \quad (1.169)$$

Vemos que:

1. Si A y B comutan,

$$[A, B] = 0 \Leftrightarrow e^A e^B = e^{A+B} \quad (1.170)$$

2. Si el comutador de A y B es un c-número

$$[A, B] = cI \Leftrightarrow e^A e^B = e^{A+B+\frac{c}{2}} \quad (1.171)$$

3. El inverso de e^A es e^{-A} . Efectivamente, como

$$[A, A] = 0 \Rightarrow e^A e^{-A} = e^{A-A} = e^0 = I \quad (1.172)$$

■ Teorema de Lie-Suzuki-Trotter

Teorema 9 Sean $A, B \subset L(\mathcal{H})$ dos operadores lineales genéricos. Entonces

$$e^{A+B} = \lim_{n \rightarrow \infty} \left(e^{A/n} e^{B/n} \right)^n \quad (1.173)$$

Esta segunda opción es de uso muy frecuente en el contexto de la *simulación cuántica*.

1.3.9.3. Operadores unitarios a partir de hermíticos

Todo operador unitario U se puede expresar como la exponencial imaginaria de un operador hermítico H

$$U = e^{iH} \quad (1.174)$$

Efectivamente,

$$U^\dagger = (e^{iH})^\dagger = e^{-iH^\dagger} = e^{-iH} = U^{-1} \quad (1.175)$$

por tanto, U es unitario si y sólo si H es hermítico.

1.3.9.4. Funciones generales

No siempre $f(z)$ admite una expansión en serie de Taylor. Por ejemplo $f(z) = \exp(1/z)$ en torno a $z = 0$ no es analítica. En estos casos, el operador $f(A)$ existe, pero para construirlo es necesario recurrir a la *forma diagonalizada*

Teorema 10 *Sea A un operador diagonalizable, y sea $A = \sum_i \lambda_i |\lambda_i\rangle\langle\lambda_i|$ su representación espectral. Entonces el operador $f(A)$ tiene la representación espectral siguiente*

$$f(A) = \sum_i f(\lambda_i) |\lambda_i\rangle\langle\lambda_i| \iff f(A_{ij}^{(D)}) = \begin{bmatrix} f(\lambda_1) & & & \\ & f(\lambda_2) & & \\ & & \ddots & \\ & & & f(\lambda_n) \end{bmatrix} \quad (1.176)$$

Ejemplos

$$e^{1/A} = \sum_i e^{1/\lambda_i} |\lambda_i\rangle\langle\lambda_i| \quad (1.177)$$

$$\begin{aligned} \text{tr}(A \log A) &= \text{tr} \left[\left(\sum_j \lambda_j |\lambda_j\rangle\langle\lambda_j| \right) \left(\sum_k \log \lambda_k |\lambda_k\rangle\langle\lambda_k| \right) \right] \\ &= \text{tr} \left[\sum_k \lambda_k \log \lambda_k |\lambda_k\rangle\langle\lambda_k| \right] \\ &= \sum_k \lambda_k \log \lambda_k \end{aligned}$$

1.3.10. Matrices de Pauli

1.3.10.1. Definición

Definición 17 *Se denominan matrices de Pauli a las tres matrices siguientes:*

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \quad (1.178)$$

- También se usan los subíndices enteros $\sigma_1 = \sigma_x$, $\sigma_2 = \sigma_y$ y $\sigma_3 = \sigma_z$.
- Las matrices de Pauli son hermíticas y unitarias.

Ejercicio 7 Escribe la descomposición espectral de las tres matrices de Pauli, σ_x, σ_y y σ_z .

1.3.10.2. Base ortogonal del espacio de Hilbert de matrices hermiticas 2×2

Estas tres matrices, junto con la matriz identidad, forman una **base ortogonal del espacio de Hilbert de matrices hermiticas** 2×2 . En otras palabras, cualquier matriz hermítica 2×2 puede escribirse como una combinación lineal de estas matrices

$$A = cI + \sum_k^3 a_k \sigma^k. \quad (1.179)$$

Es decir

$$\begin{aligned} A &= a_0 I + a_1 \sigma_1 + a_2 \sigma_2 + a_3 \sigma_3 \\ &= \begin{bmatrix} a_0 + a_3 & a_1 - ia_2 \\ a_1 + ia_2 & a_0 - a_3 \end{bmatrix} = A^\dagger. \end{aligned}$$

1.3.10.3. Composición

La **composición** de dos matrices de Pauli es otra matriz de Pauli (o la identidad I) que cumple la siguiente propiedad

$$\sigma_i \sigma_j = \delta_{ij} I + i \epsilon_{ijk} \sigma_k, \quad (1.180)$$

donde

$$\epsilon_{123} = \epsilon_{231} = \epsilon_{312} = 1, \quad \epsilon_{213} = \epsilon_{132} = \epsilon_{312} = 1 \quad (1.181)$$

y el resto son cero. La propiedad (1.180) se refiere a que las matrices de Pauli forman un grupo cerrado, es decir, al multiplicar dos matrices nos da otra matriz de Pauli (o la identidad).

De la Ec. (1.180) se deduce que

$$\sigma_x^2 = \sigma_y^2 = \sigma_z^2 = -i\sigma_x \sigma_y \sigma_z = I. \quad (1.182)$$

1.3.10.4. Traza y ortogonalidad

Las matrices de Pauli tienen **traza nula**

$$\text{tr } \sigma_j = 0 \quad (1.183)$$

Tomando la traza de la relación de composición obtenemos que las matrices de Pauli son **ortogonales** en el sentido siguiente

$$\text{tr}(\sigma_i \sigma_j) = \text{tr}(\delta_{ij} I + i \epsilon_{ijk} \sigma_k) = 2\delta_{ij} \quad (1.184)$$

Ejercicio 8 Demuestra las siguientes relaciones de (anti)comutación

$$\{\sigma_i, \sigma_j\} = 2\delta_{ij}, \quad [\sigma_i, \sigma_j] = 2i\epsilon_{ijk} \sigma_k \quad (1.185)$$

Nota: Conmutador y anti-conmutador

- Conmutador:

$$[A, B] = AB - BA \quad (1.186)$$

- Anti-conmutador

$$\{A, B\} = AB + BA \quad (1.187)$$

1.3.10.5. Exponencial de matrices de Pauli

Recordar que la fórmula de Euler permite escribir, para una fase compleja $e^{i\alpha} = \cos \alpha + i \sin \alpha$. Ahora estamos en disposición de probar la generalización a matrices de Pauli. Sea

$$\mathbf{a} = (a_1, a_2, a_3) = a \left(\frac{a_1}{a}, \frac{a_2}{a}, \frac{a_3}{a} \right) = a \hat{\mathbf{n}} \quad (1.188)$$

donde $a = |\mathbf{a}| = \sqrt{a_1^2 + a_2^2 + a_3^2}$ es el módulo

Entonces, escribimos

$$\mathbf{a} \cdot \boldsymbol{\sigma} = a \hat{\mathbf{n}} \cdot \boldsymbol{\sigma} \quad (1.189)$$

Teorema 11

$$\exp(i \mathbf{a} \cdot \boldsymbol{\sigma}) = \exp(i a \hat{\mathbf{n}} \cdot \boldsymbol{\sigma}) = \cos a I + i \sin a (\hat{\mathbf{n}} \cdot \boldsymbol{\sigma}). \quad (1.190)$$

1.3.10.6. Matrices de Pauli en otras bases

Un punto importante es que la representación de las matrices de Pauli que acabamos de ver en la Ec. (1.178) no es única (aunque sí la más extendida). Como podemos ver, esta representación es aquella en la que se diagonaliza la matriz σ_z . Es decir, están representadas en la base de **autoestados** (autovectores) de la matriz σ_z

$$|z_+\rangle = |\uparrow\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

$$|z_-\rangle = |\downarrow\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

donde

$$\sigma_z |z_+\rangle = \sigma_z, \quad \sigma_z |z_-\rangle = -\sigma_z.$$

Estas matrices podrían escribirse en otras bases, como por ejemplo la base de autoestados de σ_x , i.e. $|x_+\rangle, |x_-\rangle$, que diagonalizaría la matriz σ_x o en la base de autoestados de σ_y , i.e. $|y_+\rangle, |y_-\rangle$, que diagonalizaría la matriz σ_y . Cabe destacar que solo podemos tener a la vez una de las tres matrices en forma diagonal (puesto que no comutan). Por completitud, podemos ver la expresión de estos autoestados en función de los de σ_z :

$$\begin{aligned} |x_+\rangle &= \frac{1}{\sqrt{2}} (|z_+\rangle + |z_-\rangle) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, & |x_-\rangle &= \frac{1}{\sqrt{2}} (|z_+\rangle - |z_-\rangle) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \\ |y_+\rangle &= \frac{1}{\sqrt{2}} (|z_+\rangle + i |z_-\rangle) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix}, & |y_-\rangle &= \frac{1}{\sqrt{2}} (|z_+\rangle - i |z_-\rangle) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix}. \end{aligned} \quad (1.191)$$

donde

$$\begin{aligned} \sigma_x |x_+\rangle &= +|x_+\rangle, & \sigma_x |x_-\rangle &= -|x_-\rangle \\ \sigma_y |y_+\rangle &= +|y_+\rangle, & \sigma_y |y_-\rangle &= -|y_-\rangle. \end{aligned} \quad (1.192)$$

Nota: Autoestados y autovectores. El espín

En esta sección hemos estado hablando de **autoestados** en vez de **autovectores**. Esto es porque el significado físico que tienen las matrices de Pauli. Estas se usan, entre otras cosas, para describir el espín de una partícula, así que sus autovectores representan autoestados de espín de la partícula. En concreto, representan las proyecciones del espín sobre los tres ejes x , y y z .

Cuando decimos que una partícula tiene espín $+1/2$ o $-1/2$ habitualmente nos referimos a que la proyección del espín sobre el eje z es $+1/2$ o $-1/2$ (obviando constantes multiplicativas). Es decir, está en el estado $|+\rangle$ o $|-\rangle$.

En física cuántica, un autoestado de un operador es un estado que no evoluciona con el tiempo (si no hay influencias externas). Por ejemplo, cuando tenemos un electrón aislado con proyección del espín $+1/2$, este estado no va a cambiar con el tiempo, pues el electrón está aislado.

1.4. Tensores

1.4.1. Producto tensorial

Llegados a este punto, un lector con ciertos conocimientos de Computación Cuántica podría plantearse la siguiente pregunta: ¿por qué durante toda la explicación anterior estamos tratando con espacios de Hilbert de dimensión arbitraria, si los qubits tienen dimensión dos?

Esto es muy sencillo de explicar. Con lo que vamos a trabajar no es con un solo qubit, sino con un conjunto de qubits. Cada qubit vivirá en un espacio de Hilbert de dimensión dos, pero el estado total del sistema de muchos qubits vivirá en un espacio de dimensión mayor. Este espacio estará formado por el **producto tensorial** de los espacios de Hilbert de cada qubit.

Es decir, el producto tensorial es una herramienta matemática que describe los **sistemas compuestos** (en nuestro caso, sistemas compuestos por varios qubits).

1.4.1.1. Definiciones

Definición 18 Dados dos vectores $|u\rangle_1 \in \mathcal{H}_1$ y $|v\rangle_2 \in \mathcal{H}_2$, denominamos **producto tensorial al par ordenado**

$$|uv\rangle \equiv |u\rangle_1 \otimes |v\rangle_2 \quad (1.193)$$

Este producto tensorial es **bilineal**:

$$\begin{aligned} (|u\rangle_1 + |v\rangle_1) \otimes (|y\rangle_2 + |z\rangle_2) &\equiv |u\rangle_1 \otimes |y\rangle_2 + |u\rangle_1 \otimes |z\rangle_2 + |v\rangle_1 \otimes |y\rangle_2 + |v\rangle_1 \otimes |z\rangle_2 \\ &= |uy\rangle + |uz\rangle + |vy\rangle + |vz\rangle \end{aligned}$$

Ahora viene la segunda definición clave:

Definición 19 El **espacio producto tensorial** $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2$ está formado por **todas las combinaciones lineales posibles de productos tensoriales**

$$|s\rangle = a|u\rangle_1 \otimes |u\rangle_2 + b|v\rangle_1 \otimes |v\rangle_2 + \dots \quad (1.194)$$

donde $|u\rangle_1, |v\rangle_1, \dots \in \mathcal{H}_1$ y $|u\rangle_2, |v\rangle_2, \dots \in \mathcal{H}_2$, y $a, b, \dots \in \mathbb{C}$ son coeficientes complejos.

Nota: Comentarios para el resto de la sección

- En el resto de esta lección nos restringiremos al caso de $\mathcal{H}_1 = \mathcal{H}_2$ y, por tanto, $d_1 = d_2 \equiv d$, y $\mathcal{H} \otimes \mathcal{H} \equiv \mathcal{H}^{\otimes 2}$
- Prescindiremos del subíndice $|u\rangle_1 \otimes |y\rangle_2 = |u\rangle \otimes |y\rangle \equiv |uy\rangle$ que estará implícito en el orden.
- Para computación cuántica con *qubits* (*qudits*), el valor relevante es $d = 2$ ($d \geq 3$).

1.4.1.2. Base y dimension

Sea $|e_i\rangle$ una base de \mathcal{H} . Entonces, una base de $\mathcal{H} \otimes \mathcal{H} = \mathcal{H}^{\otimes 2}$ se obtiene a partir de *todos* los emparejamientos

$$|e_{ij}\rangle = |e_i\rangle \otimes |e_j\rangle \quad i, j = 1 \dots d \quad (1.195)$$

- El número de parejas posibles es d^2 , que coincide con la **dimensión** de $\mathcal{H}^{\otimes 2}$.
- Vemos que las etiquetas de los vectores de la base forman un *bi-índice* $ij = 11, 12, 21, 22, \dots$
- Un vector general se escribirá igualmente usando un bi-índice en lugar de un índice

$$|\omega\rangle = \sum_{i,j=1}^d w_{ij} |e_{ij}\rangle = w_{11} |e_{11}\rangle + w_{12} |e_{12}\rangle + \dots \quad (1.196)$$

donde w_{ij} son d^2 componentes complejas.

1.4.1.3. Producto de Kronecker

Como sabemos, cualquier vector admite, en una base, una representación como un vector columna con sus coeficientes como entradas. La matriz columna asociada $|uv\rangle = |u\rangle \otimes |v\rangle$ se forma a partir de las matrices columna de $|u\rangle$ y $|v\rangle$ mediante el denominado **producto de Kronecker** o, también *producto tensorial*

$$|uv\rangle = |u\rangle \otimes |v\rangle \sim \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \otimes \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \equiv \begin{bmatrix} u_1 & \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \\ u_2 & \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} u_1 v_1 \\ u_1 v_2 \\ u_2 v_1 \\ u_2 v_2 \end{bmatrix} \quad (1.197)$$

Ejemplo

Con $d = 2$ tendríamos $d^2 = 4$ elementos de la base de $\mathcal{H}^{\otimes 2}$

$$|e_{11}\rangle = |e_1\rangle \otimes |e_1\rangle \sim \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad |e_{12}\rangle = |e_1\rangle \otimes |e_2\rangle \sim \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad (1.198)$$

$$|e_{21}\rangle = |e_2\rangle \otimes |e_1\rangle \sim \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad |e_{22}\rangle = |e_2\rangle \otimes |e_2\rangle \sim \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (1.199)$$

El *vector más general* $|w\rangle \in \mathcal{H} \otimes \mathcal{H}$ admitirá una representación de la forma

$$|w\rangle = \sum_{i,j=1}^2 w_{ij} |e_{ij}\rangle = \begin{bmatrix} w_{11} \\ w_{12} \\ w_{21} \\ w_{22} \end{bmatrix} \quad (1.200)$$

1.4.1.4. Indexación equivalente

Podemos etiquetar las componentes (o los elementos de la base) con índices. Para ello debemos definir

un mapa entre bi-índices $i, j = 1, 2$ e índices $a = 1, \dots, 4$

$$w_{ij} = \begin{bmatrix} w_{11} \\ w_{12} \\ w_{21} \\ w_{22} \end{bmatrix} \longleftrightarrow w_a = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}, \quad (1.201)$$

donde hemos querido resaltar que los vectores son los mismos, etiquetados de forma diferente. Haciendo lo mismo con los elementos de la base tenemos que

$$|w\rangle = \sum_{i,j=1}^2 w_{ij} |e_{ij}\rangle = \sum_{a=1}^4 w_a |e_a\rangle \quad (1.202)$$

En el caso general, $i, j = 1, \dots, d$ el mapa es $w_{ij} \rightarrow w_a$ con

$$a = d \cdot (i - 1) + j \quad (1.203)$$

1.4.2. Factorización y entrelazamiento

1.4.2.1. Estado factorizable o entrelazado

Llegamos a uno de los conceptos clave en Teoría Cuántica de la Información.

Definición 20 Decimos que, un vector $|w\rangle \in \mathcal{H} \otimes \mathcal{H}$ es **factorizable** cuando es posible encontrar vectores $|u\rangle, |v\rangle \in \mathcal{H}$ tales que $|w\rangle = |u\rangle \otimes |v\rangle$.

Cuando esto no sea posible, decimos que $|w\rangle$ es un vector entrelazado.

Ya hemos visto que, dada una base $|e_i\rangle$ de \mathcal{H} , el vector más general que pertenece al espacio producto admite una descomposición

$$|w\rangle = \sum_{i,j=1}^d w_{ij} |e_i\rangle \otimes |e_j\rangle = w_{11} |e_1\rangle \otimes |e_1\rangle + w_{12} |e_1\rangle \otimes |e_2\rangle + \dots \quad (1.204)$$

Podría ocurrir que en otra base $|w\rangle = \tilde{w}_{11} |f_1\rangle \otimes |f_1\rangle$ sólo tuviese un término y fuese factorizable. Discernir si un vector es factorizable o entrelazado no es algo que se pueda hacer a primera vista.

1.4.2.2. Criterio de factorización

El estado es $|w\rangle$ es factorizable si y sólo si las componentes w_{ij} son factorizables en la forma $w_{ij} = u_i v_j$ con $i, j = 1, \dots, d$.

Demostración:

$$|w\rangle = \sum_{i,j=1}^d w_{ij} |e_{ij}\rangle = \sum_{i,j} u_i v_j |e_i\rangle \otimes |e_j\rangle = \sum_{i,j} u_i |e_i\rangle \otimes v_j |e_j\rangle = \sum_i u_i |e_i\rangle \otimes \sum_j v_j |e_j\rangle = |u\rangle \otimes |v\rangle \quad (1.205)$$

Identidad que se puede leer en ambos sentidos ■

El carácter entrelazado de un vector es *genérico*, mientras que el carácter factorizable es *accidental*. Esto se sigue de un sencillo conteo: como función de d , $\{w_{ij}\}$ forma un conjunto de d^2 parámetros complejos (grados de libertad). Sin embargo en $\{u_i v_j\}$ sólo hay $2d$ números independientes. Es evidente que $d^2 > 2d$.

Nota: caso con $d = 2$

En el caso $d = 2$ la condición $w_{ij} = u_i v_j$ es *equivalente* a verificar la anulación del determinante de la matriz 2×2 formada por las componentes

$$\det w_{ij} = w_{11}w_{22} - w_{12}w_{21} = u_1v_1u_2v_2 - u_1v_2u_2v_1 = 0 \quad (1.206)$$

1.4.2.3. Descomposición de Schmidt

Vamos a estudiar el caso general en el que los *espacios factor* no son iguales $|w\rangle \in \mathcal{H} \otimes \mathcal{H}_2$. Supongamos que, en sendas bases arbitrarias $\{|e_{1,i}\rangle, i = 1, \dots, d_1\}$ de \mathcal{H}_1 y $\{|e_{2,a}\rangle, a = 1, \dots, d_2\}$ de \mathcal{H}_2 nuestro vector se escribe

$$|w\rangle = \sum_{i=1}^{d_1} \sum_{a=1}^{d_2} w_{ia} |e_{1,i}\rangle \otimes |e_{2,a}\rangle \quad (1.207)$$

Los valores de las *componentes* w_{ia} **dependen de las bases escogida**. En *otras* base $|\tilde{e}_{1,i}\rangle \otimes |\tilde{e}_{2,a}\rangle$ encontraremos *otras* componentes \tilde{w}_{ia} para el *mismo* vector.

Si existe una base en la que $\tilde{w}_{ia} = 0$ para todos los i, a menos para uno (por ejemplo $\tilde{w}_{11} \neq 0$), entonces

$$|w\rangle = \tilde{w}_{11} |\tilde{e}_{1,1}\rangle \otimes |e_{2,1}\rangle \quad (1.208)$$

y, secretamente, el vector $|w\rangle$ sería factorizable.

El siguiente teorema nos dice **cuánto nos podemos acercar a esta situación**.

Teorema 12 (de Schmidt) *Para cada vector $|w\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$, existen bases $\{|f_{1,i}\rangle\}$ de \mathcal{H}_1 y $\{|f_{2,i}\rangle\}$ de \mathcal{H}_2 , tales que, podemos expresar*

$$|w\rangle = \sum_{i=1}^r s_i |f_{1,i}\rangle \otimes |f_{2,i}\rangle , \quad (1.209)$$

con $s_i > 0$, que involucra el **mínimo número**, r , de términos.

Es importante darse cuenta de que en la descomposición de la Ec. (1.209) se está sumando solo sobre un índice. Este es igual en los dos elementos de la base en cada término. Es decir, si representamos $|w\rangle$ como una **matriz** (en vez de como un vector), esta sería diagonal y rectangular, con los lados iguales a las dimensiones de los dos espacios \mathcal{H}_1 y \mathcal{H}_2 .

El número $1 \leq r \leq \min(d_1, d_2)$ se denomina **Número de Schmidt** y es la información relevante porque:

- Cuando $r = 1$, el estado $|w\rangle$ será *factorizable*.
- Cuando $r \geq 2$, el estado será *entrelazado*.

La demostración del Teorema de Schmidt es interesante porque nos da un **método constructivo** para encontrar la descomposición.

Demostración: Supongamos que nuestro vector se escribe

$$|w\rangle = \sum_{i=1}^{d_2} \sum_{a=1}^{d_2} w_{ia} |e_{1,i}\rangle \otimes |e_{2,a}\rangle \quad (1.210)$$

La matriz de coeficientes w_{ia} tiene dimensión $d_1 \times d_2$. El **teorema de descomposición en**

valores singulares nos garantiza que podemos expresar dicha matriz en la forma siguiente

$$w = U\Sigma V^\dagger \Rightarrow w_{ia} = \sum_{j=1}^{d_1} \sum_{b=1}^{d_2} U_{ij}\Sigma_{jb}V_{ab}^* \quad (1.211)$$

donde U y V son unitarias ($d_1 \times d_1$) y ($d_2 \times d_2$) respectivamente, mientras que Σ es diagonal

$$\Sigma = \underbrace{\begin{bmatrix} s_1 & \cdots & & 0 \\ \vdots & \ddots & & \vdots \\ & & s_r & 0 \\ & & & \ddots \\ 0 & \cdots & & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & & 0 \end{bmatrix}}_{d_1} \quad \Rightarrow \quad \Sigma_{jb} = s_j \delta_{jb} \quad (1.212)$$

Esto quiere decir que podemos escribir

$$\begin{aligned} |w\rangle &= \sum_{i=1}^{d_1} \sum_{a=1}^{d_2} \left(\sum_{j=1}^{d_1} \sum_{b=1}^{d_2} U_{ij}\Sigma_{jb}V_{ab}^* \right) |e_{1,i}\rangle \otimes |e_{2,a}\rangle \\ &= \sum_{j=1}^{d_1} \sum_{b=1}^{d_2} \Sigma_{jb} \left(\sum_{i=1}^{d_1} U_{ij} |e_{1,i}\rangle \right) \otimes \left(\sum_{a=1}^{d_2} V_{ab}^* |e_{2,a}\rangle \right) \\ &= \sum_{j=1}^{d_1} \sum_{b=1}^{d_2} \Sigma_{jb} |f_{1,j}\rangle \otimes |f_{2,b}\rangle \\ &= \sum_{j=1}^r s_j |f_{1,j}\rangle \otimes |f_{2,j}\rangle \end{aligned}$$

■

Por tanto, podemos saber si un **estado bipartito** es entrelazado calculando las descomposición en valores singulares de su matriz de coeficientes en cualquier base.

1.4.3. Producto tensorial múltiple

El producto tensorial se puede generalizar a más de un factor:

El espacio $\mathcal{H}_1 \otimes \mathcal{H}_2 \dots \otimes \mathcal{H}_n$ formado por **todas** las **n-tuplas** ordenadas de vectores

$$|u\rangle = |u_1 u_2 \dots u_n\rangle \equiv |u_1\rangle \otimes |u_2\rangle \otimes \dots \otimes |u_n\rangle \quad (1.213)$$

donde $|u_i\rangle \in \mathcal{H}_i$ y sus combinaciones lineales $\{a|u\rangle + b|v\rangle + \dots\}$.

Salvo mención, en adelante asumiremos que todos los $\mathcal{H}_j = \mathcal{H}$ son iguales y de dimensión d .

1.4.3.1. Base

Una base de $\mathcal{H}^{\otimes n}$ se obtiene a partir de cadenas

$$|i_1 i_2 \dots i_n\rangle = |i_1\rangle |i_2\rangle \dots |i_n\rangle \quad (1.214)$$

donde $i_1, \dots, i_n = 0, \dots, d - 1$.

Nota: los multi-índices

Con la notación $i_1, \dots, i_n = 0, \dots, d - 1$ queremos decir que **cada uno de los índices** i_1, \dots, i_n toma valores de 0 a $d - 1$.

- El número de posibles cadenas es $d d \dots d = d^n$ que es la dimensión de $\mathcal{H}^{\otimes n}$.

$$\dim_{\mathbb{C}} \mathcal{H}^{\otimes n} = d^n \quad (1.215)$$

- Podemos cambiar de etiqueta

$$|i_1 \dots i_n\rangle \rightarrow |a\rangle \quad (1.216)$$

con

$$a = i_n + d i_{n-1} + d^2 i_{n-2} + \dots + d^{n-1} i_1 \in (0, d^n - 1) \quad (1.217)$$

- Si cada base $\{|i\rangle\}$ es ortonormal, tendremos que la base producto también lo será

$$\langle i_1 i_2 \dots i_n | j_1 j_2 \dots j_n \rangle = \delta_{i_1 j_1} \delta_{i_2 j_2} \dots \delta_{i_n j_n} \quad \leftrightarrow \quad \langle a | b \rangle = \delta_{ab} \quad (1.218)$$

1.4.3.2. Estado entrelazado

Un **vector general** admitirá una expansión en esta base mediante d^n **componentes complejas** $u_{i_1 i_2 \dots i_n}$ en la forma

$$|u\rangle = \sum_{i_1, \dots, i_n=0}^{d-1} u_{i_1 i_2 \dots i_n} |i_1 i_2 \dots i_n\rangle = \sum_{a=0}^{d^n-1} u_a |a\rangle . \quad (1.219)$$

Nota: notación de multi-índices en el sumatorio

Véase que en el sumatorio anterior **cada uno de los índice** se suma de 0 a $d - 1$. Es decir

$$\sum_{i_1, \dots, i_n=0}^{d-1} = \sum_{i_1=0}^{d-1} \sum_{i_2=0}^{d-1} \dots \sum_{i_n=0}^{d-1} \quad (1.220)$$

Podemos obtener cualquier componente compleja proyectando sobre el elemento correspondiente de la base

$$u_{i_1 i_2 \dots i_n} = \langle i_1 i_2 \dots i_n | u \rangle \quad \leftrightarrow \quad u_a = \langle a | u \rangle \quad (1.221)$$

1.4.3.3. Estado factorizable

Como ya comentamos, de lo más común es que un vector sea entrelazado. Lo que son casos particulares son los **estados factorizables**. Un vector de $\mathcal{H}^{\otimes n}$ se podrá escribir en forma factorizada

$$|w\rangle = |v_1\rangle |v_2\rangle \dots |v_n\rangle \equiv |v_1 v_2 \dots v_n\rangle \quad (1.222)$$

Escribiendo $|v_k\rangle = \sum_{i_k=1}^d v_{i_k} |i_k\rangle$ vemos que un *vector factorizable* admite una expansión general en la que los coeficientes son factorizables

$$u_{i_1 i_2 \dots i_n} = v_{i_1} v_{i_2} \dots v_{i_n} \quad (1.223)$$

están parametrizados por nd cantidades v_{i_k} , $i_k = 1, \dots, d$, $k = 1, \dots, n$.

Nota:

- $nd \ll d^n$. El crecimiento exponencial del número de estados entrelazados es el ingrediente crucial para la computación cuántica. Observar que d^n es el *número de enteros* alcanzables por n bits. Pero en computación cuántica es el *número de dimensiones* en la que podemos poner d^n amplitudes complejas.
- No existe un criterio general para saber si un estado es, a priori, factorizable o entrelazado.
- Además, hay formas de caracterizar matemáticamente el nivel de entrelazamiento (*entanglement witnesses*, *entanglement monotones* etc.) desde nulo (estado factorizable) hasta maximal (contiene todos los estados de la base con igual amplitud).

1.4.4. Espacio $L(\mathcal{H}^{\otimes})$

El espacio $\mathcal{H}^{\otimes n}$ admite, como cualquier espacio vectorial, la acción de *operadores lineales* $A : \mathcal{H}^{\otimes n} \rightarrow \mathcal{H}^{\otimes n}$ donde

$$A : |u\rangle \rightarrow |v\rangle \equiv A|u\rangle \quad (1.224)$$

El conjunto de todos los operadores lineales forman el espacio vectorial $L(\mathcal{H}^{\otimes n})$.

1.4.4.1. Matrices

- A cada operador, A , le podemos asociar una *matriz*, una vez elegimos nuestra base $\{|i_1 i_2 \dots i_n\rangle\}$ donde, $i_a = 1, \dots, d$.
- Los *elementos de matriz* ahora vendrán etiquetados por dos *multi-índices*.

$$A_{i_1 \dots i_n, j_1 \dots j_n} = \langle i_1 \dots i_n | A | j_1 \dots j_n \rangle \quad \leftrightarrow \quad A_{ab} = \langle a | A | b \rangle \quad (1.225)$$

donde hemos re-etiquetado los multi-índices como un solo índice

- Con la matriz, el operador se reconstruye de la forma usual

$$A = \sum_{i_1, \dots, i_n, j_1, \dots, j_n=0}^{d-1} A_{i_1 \dots i_n, j_1 \dots j_n} |i_1 \dots i_n\rangle \langle j_1 \dots j_n| = \sum_{a,b=0}^{d^N-1} A_{ab} |a\rangle \langle b| \quad (1.226)$$

- En $A_{i_1 \dots i_n, j_1 \dots j_n} = A_{ab}$ hay $d^n \times d^n = d^{2n}$ grados de libertad. Esta sería la dimensión del espacio $L(\mathcal{H}^{\otimes n})$.

1.4.4.2. Producto tensorial de operadores

En $L(\mathcal{H}^{\otimes n})$ hay un análogo de los vectores factorizables de $\mathcal{H}^{\otimes n}$ que ahora serán *operadores factorizables*. Supongamos que existen n operadores lineales $A^{(a)}$, $a = 1, \dots, n$ definidos sobre cada espacio factor \mathcal{H} .

Definición 21 La acción del producto tensorial de operadores $A = A^{(1)} \otimes A^{(2)} \otimes \dots \otimes A^{(n)}$ sobre un vector $|v\rangle = |v\rangle_1 \otimes \dots \otimes |v_n\rangle \in \mathcal{H}$ viene dada por

$$A|v\rangle = A^{(1)}|v_1\rangle \otimes \dots \otimes A^{(n)}|v_n\rangle. \quad (1.227)$$

La acción sobre vectores generales se sigue imponiendo linealidad

$$A(|v\rangle + |w\rangle) = A|v\rangle + A|w\rangle . \quad (1.228)$$

Se sigue automáticamente de la definición que:

- El adjunto de un producto tensorial de operadores es el producto de los adjuntos (no se permuta el orden)

$$A^\dagger = A^{(1)\dagger} \otimes \dots \otimes A^{(n)\dagger} \quad (1.229)$$

- El producto tensorial de operadores hermíticos es hermítico

$$A^{(a)\dagger} = A^{(a)} \implies A^\dagger = A \quad (1.230)$$

- El producto tensorial de operadores unitarios, es unitario

$$A^{(a)\dagger} = A^{(a)-1} \implies A^\dagger = A^{-1} \quad (1.231)$$

1.4.4.3. Producto de Kronecker de matrices. Operadores factorizables.

¿Cómo será la matriz $A_{i_1\dots i_n, j_1\dots j_n}$ de un operador factorizable, $A = A^{(1)} \otimes A^{(2)} \otimes \dots \otimes A^{(n)}$, en términos de las matrices $A_{ij}^{(a)}$ de sus factores? Vamos a tomar $n = 2$ por simplicidad

$$A = A^{(1)} \otimes A^{(2)} = \left(\sum_{i_1 i_2} A_{i_1 j_1}^{(1)} |i_1\rangle \langle j_1| \right) \left(\sum_{i_2 j_2} A_{i_2 j_2}^{(2)} |i_2\rangle \langle j_2| \right) \quad (1.232)$$

$$= \sum_{i_1 i_2, j_1 j_2} A_{i_1 j_1}^{(1)} A_{i_2 j_2}^{(2)} |i_1 i_2\rangle \langle j_1 j_2| \quad (1.233)$$

$$= \sum_{i_1 i_2, j_1 j_2} A_{i_1 i_2, j_1 j_2} |i_1 i_2\rangle \langle j_1 j_2| \quad (1.234)$$

Vemos que la matriz asociada a A se obtiene a partir de las matrices de $A^{(a)}$ mediante el *producto exterior de las matrices*, o **producto de Kronecker**.

$$A_{i_1 i_2, j_1 j_2} = A_{i_1 j_1}^{(1)} A_{i_2 j_2}^{(2)} \quad (1.235)$$

El método para de **representar** matricialmente el producto de Kronecker de dos matrices $A \otimes B$ es sencillo. Supongamos que $d = 2$ y tenemos un operador producto $A \otimes B$. Entonces su matriz

$$(A \otimes B)_{ab} = \begin{pmatrix} A_{00}B & A_{01}B \\ A_{10}B & A_{11}B \end{pmatrix} = \begin{pmatrix} A_{00}B_{00} & A_{00}B_{01} & A_{01}B_{00} & A_{01}B_{01} \\ A_{00}B_{10} & A_{00}B_{11} & A_{01}B_{10} & A_{01}B_{11} \\ A_{10}B_{00} & A_{10}B_{01} & A_{11}B_{00} & A_{11}B_{01} \\ A_{10}B_{10} & A_{10}B_{11} & A_{11}B_{10} & A_{11}B_{11} \end{pmatrix}. \quad (1.236)$$

El producto de Kronecker verifica las siguientes propiedades para dos matrices A y B de dimensiones

d_A y d_B .

$$\begin{aligned}
 (A \otimes B)(C \otimes D) &= (AC) \otimes (BD) \\
 \text{tr}(A \otimes B) &= (\text{tr}A)(\text{tr}B) \\
 A \otimes (B + D) &= A \otimes B + A \otimes D \\
 (A \otimes B)^\dagger &= A^\dagger \otimes B^\dagger \\
 (A \otimes B)^{-1} &= A^{-1} \otimes B^{-1} \\
 \det(A \otimes B) &= (\det A)^{d_B}(\det B)^{d_A}
 \end{aligned} \tag{1.237}$$

donde AC significa el producto de matrices A y C .

Ejercicio 9 Demuestra estos resultados.

La generalización a todo n es obvia. El producto de Kronecker de n matrices $A_{i_a j_a}^{(a)}$ asociadas a operadores $A^{(a)}$ es

$$A_{i_1 \dots i_n, j_1 \dots j_n} = A_{i_1 j_1}^{(1)} \dots A_{i_n j_n}^{(n)} \tag{1.238}$$

Ejercicio 10 Calcula $\sigma_1 \otimes \sigma_2 \otimes \sigma_3$

Nota: operadores factorizables y no factorizables

Observar que en un operador general, la matriz $A_{i_1 \dots i_n, j_1 \dots j_n}$ tiene $d^n \times d^n = d^{2n}$ entradas independientes. Sin embargo en un producto de Kronecker $A_{i_1 j_1}^{(1)} \dots A_{i_n j_n}^{(n)}$ sólo hay nd^2 . Por tanto, los *operadores factorizables* forman un subconjunto muy pequeño dentro del conjunto de los operadores generales.

1.4.4.4. Generación de entrelazamiento

Supongamos que $|u\rangle = |u_1\rangle \otimes |u_2\rangle$ es factorizable.

- Si $A = A_1 \otimes A_2$ es un operador factorizable, el resultado $|v\rangle = A|u\rangle = A_1|u_1\rangle \otimes A_2|u_2\rangle$ también es factorizable.
- Inversamente, si buscamos un operador que genere entrelazamiento, entonces no puede ser factorizable $A \neq A_1 \otimes A_2$

Ejercicio 11 Considera la base $\{|0\rangle, |1\rangle\}$ del espacio \mathcal{H} de dimensión 2. Sea $A = B + C$ un operador que actúa sobre el producto $\mathcal{H} \otimes \mathcal{H}$, donde B y C son operadores factorizables dados por

$$B = |0\rangle\langle 0| \otimes I, \quad C = |1\rangle\langle 1| \otimes (|0\rangle\langle 1| + |1\rangle\langle 0|) \tag{1.239}$$

Escribe los elementos B_{ij} y C_{ij} , $i, j = 1, 2, 3, 4$ y obtén A_{ij} . Comprueba si C , B y A son operadores unitarios o no.

1.5. Probabilidades

La Mecánica Cuántica es **intrínsecamente probabilística**. Por ello, es importante repasar algunos conceptos estadístico.

1.5.1. Variables aleatorias

1.5.1.1. Variable aleatoria

Denotamos con $(X, p(X))$ una **variable aleatoria** donde

- X es el **espacio muestral** de valores $\{x_1, x_2, \dots, x_n\}$ que pueden aparecer en una *consulta* a la variable aleatoria
- $p(X)$ es la **distribución de probabilidad**

1.5.1.2. Distribución de probabilidad

Una **distribución de probabilidad** es una función real $x \rightarrow p(x)$ que debe verificar las dos condiciones siguientes

$$p(x) \in [0, 1] \quad , \quad \sum_{x \in X} p(x) = 1 \quad (1.240)$$

Es decir, la suma de probabilidades de todos los sucesos posibles debe ser la unidad.

1.5.1.3. Media

La **media** de una variable aleatoria viene dada por la expresión

$$\bar{X} = \sum_i x_i p(x_i) \quad (1.241)$$

1.5.1.4. Varianza y desviación estándar

La **varianza**, σ_X^2 , es la *media de la desviación cuadrática* $\overline{(x_i - \bar{X})^2}$

$$\sigma_X^2 = \sum_j (x_j - \bar{X})^2 p(x_j) = \overline{X^2} - \bar{X}^2 \quad (1.242)$$

La cantidad σ_X se denomina **desviación estándar**

$$\sigma_X = \sqrt{\overline{X^2} - \bar{X}^2} \quad (1.243)$$

1.5.2. La conexión estadística

Nuestro conocimiento del mundo se basa en la realización de **experimentos**, el resultado de los cuales es (empíricamente) **aleatorio**. Podemos pensar en el hecho de medir un sistema como la consulta de una variable aleatoria $(X, p(X))$ donde la distribución de probabilidad incorpora todo nuestro conocimiento acerca del sistema.

1.5.2.1. Frecuencias e Histogramas

Cualquier consulta o medida da lugar a una *muestra* finita de valores $A_N = (a_1, a_2, \dots, a_N)$. Cada uno de estos resultados puede tomar cualquier valor x_j del espacio muestral, es decir, $a_i \in \{x_1, \dots, x_n\}$. A su vez, medidas diferentes (a_i diferentes) pueden tomar valores iguales x_j , con números de aparición $n(x_i)$ tales que $n(x_1) + \dots + n(x_p) = N$.

Estos datos se pueden agrupar en intervalos o *bins* que eliminan cierta precisión numérica. Por ejemplo, si truncamos nuestra precisión a las décimas de unidad, 13.10 y 13.19 pertenecerán al mismo *bin*.

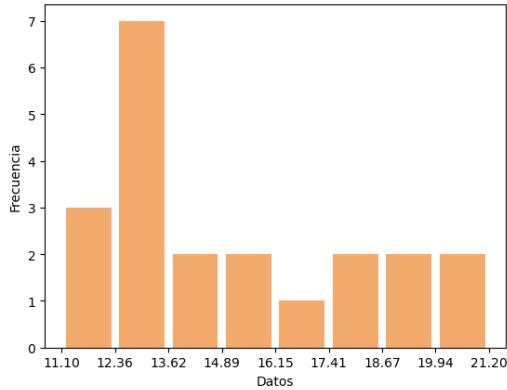


Figura 1.2: Ejemplo de histograma.

Un **histograma** es un diagrama en el que, por cada *bin*, hay una columna, cuya altura representa el número de sucesos que pertenecen a dicho *bin*. Podemos ver uno en la Fig. 1.2.

1.5.2.2. Ley de los grandes números

La conexión entre estadística y teoría de la probabilidad se da mediante la **ley de los grandes números**. Lo que nos dice esta ley es que cuando el número de muestras tiende a infinito, $N \rightarrow \infty$, las fracciones relativas tienden a un número fijo que hereda, de toda la dinámica del sistema, ciertas propiedades que no se desvanecen. Este número es la probabilidad, es decir,

$$f_N(x_i) = \frac{n(x_i)}{N} \xrightarrow{N \rightarrow \infty} p(x_i) \quad (1.244)$$

Un punto importante aquí es darnos cuenta de que experimentalmente sólo tenemos acceso a las frecuencias relativas $f_N(x_i)$ para un N grande aunque **finito**.

Igualmente, nuestro conocimiento de la media \bar{X} y la varianza σ_X^2 siempre es aproximado, y se realiza a través de las medias y varianzas muestrales

$$\bar{A}_N = \sum_i x_i f_N(x_i) \xrightarrow{N \rightarrow \infty} \bar{X} \quad (1.245)$$

$$\sigma_{A_N}^2 = \sum_i (x_i - \bar{A}_N)^2 f_N(x_i) \xrightarrow{N \rightarrow \infty} \sigma_X^2 \quad (1.246)$$

Nota

Este es el enfoque **frecuentista**.

1.5.2.3. La distribución de Bernouilli

Una **variable aleatoria de Bernouilli** $X = (x, p(x))$ tiene dos posibles resultados

- **Éxito:** $\rightarrow x = 1$ con probabilidad $p(1) = p$
- **Fracaso:** $\rightarrow x = 0$ con probabilidad $p(0) = 1 - p$

Podemos calcular fácilmente

$$\begin{aligned} \bar{X} &= \sum_i x_i p_i = 1 \cdot p + 0 \cdot (1 - p) = p \\ \sigma^2 &= \sum_i (x_i - \bar{X})^2 p_i = (1 - p)^2 p + (0 - p)^2 (1 - p) = p(1 - p) \end{aligned} \quad (1.247)$$

1.5.2.4. La distribución Binomial

La **variable aleatoria binomial** $X = (x, p(x))$ se define como

$$x = \text{número de éxitos obtenidos en } n \text{ pruebas de Bernouilli sucesivas} \quad (1.248)$$

Claramente $x \in (0, 1, 2, \dots, n)$.

Ahora es muy sencillo obtener la **probabilidad** de un suceso con x éxitos

$$p(x) = \binom{n}{x} p^x (1-p)^{n-x} \quad (1.249)$$

donde el primer factor tiene en cuenta las posibles ordenaciones en que aparecen x éxitos en n intentos.

Podemos ver esta distribución en la Fig. 1.3.

Un cálculo un poco más largo permite ver que, ahora

$$\begin{aligned} \bar{X} &= np \\ \sigma^2 &= np(1-p) \end{aligned} \quad (1.250)$$

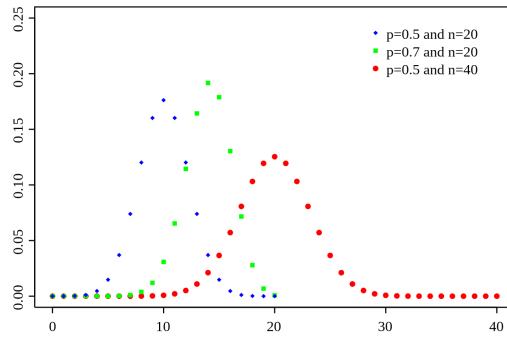


Figura 1.3: Distribución binomial

1.5.2.5. La distribución Normal o Gaussiana

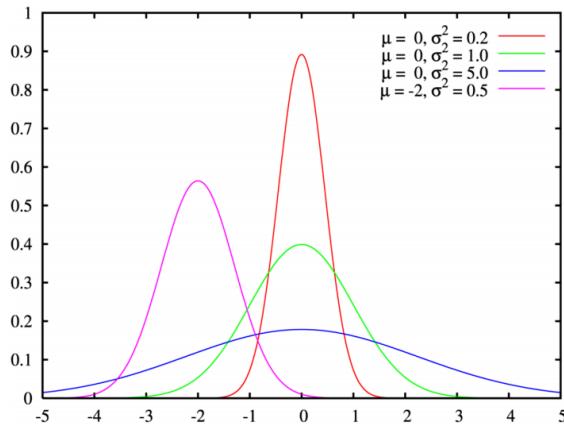
La **distribución normal** (o **gaussiana**) centrada en μ y con anchura σ viene dada por

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (1.251)$$

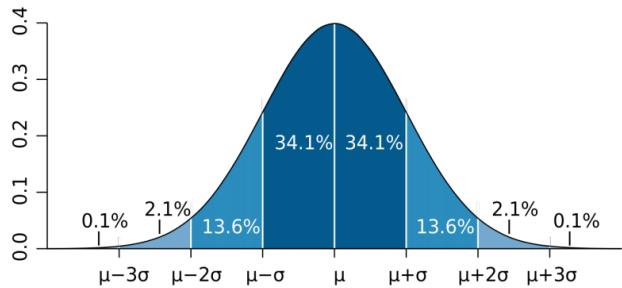
Nos encontramos ante una variable aleatoria con un espacio muestral continuo $x \in (-\infty, +\infty)$.

Podemos ver esta distribución en la Fig. 1.4. Tenemos además que

$$\begin{aligned} \bar{X} &= \int_{-\infty}^{+\infty} xp(x)dx = \mu \\ \overline{(x - \bar{X})^2} &= \int_{-\infty}^{+\infty} (x - \mu)^2 p(x)dx = \sigma^2 \end{aligned} \quad (1.252)$$



(a) Diferentes valores de μ y σ



(b) Distribución de probabilidad en torno a la media

Figura 1.4: Distribución normal (o gaussina)

1.5.3. Probabilidades combinadas

Las probabilidades combinadas son la base de las **correlaciones**. Es aquí donde la Mecánica Cuántica produce resultados inesperados clásicamente. Esto es porque cantidades que son independientes desde el punto de vista clásico, presentan correlaciones al hacer el experimento.

Ahora vamos a examinar variables aleatorias formadas por dos espacios muestrales X e Y . Dependiendo de la forma en que combinemos la observación de cada una tendremos distintas distribuciones de probabilidad.

1.5.3.1. Probabilidad combinada

Una forma de lidiar con las correlaciones es tratar el conjunto de varias variables aleatorias como una sola variable. Esta es la idea detrás de la **probabilidad combinada**,

La **distribución de probabilidad combinada** asocia un número $p(x, y)$ a la probabilidad de observación conjunta de x e y .

De esta forma, tratamos las parejas de eventos como un solo evento $a = (x, y)$. Por eso, la condición de normalización ahora es

$$\sum_a p(a) = \sum_{xy} p(x, y) = 1. \quad (1.253)$$

- **Distribución marginal:**

La suma parcial sobre una de las dos variables conduce a sendas **distribuciones marginales**

$$q(x) = \sum_y p(x, y) \quad \tilde{q}(y) = \sum_x p(x, y) \quad (1.254)$$

- **Variables independientes:**

Si dos variables aleatorias no presentan **ninguna correlación**, decimos que son **independientes**. En este caso las probabilidades combinadas factorizan

$$p(x, y) = p(x)p(y) \quad (1.255)$$

La distribución de cada variable coincide con la que se deduce de marginalizar la otra

$$\sum_y p(x,y) = p(x) \quad , \quad \sum_x p(x,y) = p(y) \quad (1.256)$$

1.5.3.2. Probabilidad condicionada

La distribución de **probabilidad condicionada** $p(X|Y)$ asigna un número $p(x|y)$ a la probabilidad de encontrar un suceso $X = x$ una vez *sabemos* que $Y = y$ ha sido el resultado de consultar Y .

La manera de acceder experimentalmente a estas distribuciones, es efectuar un muestreo $(a_i, b_i), i = 1, \dots, N$ de valores de (X, Y) y *seleccionar* sólo aquellos sucesos donde $b_i = y$ un valor concreto de Y .

Teorema 13 (de Bayes) *Las probabilidades condicionales y combinadas se relacionan de la forma siguiente*

$$p(x,y) = p(x|y)p(y) = p(y|x)p(x) \quad (1.257)$$

La segunda igualdad conduce al teorema de Bayes

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \quad (1.258)$$

1.5.4. Entropía de una variable aleatoria

Definición 22 *Dada una variable aleatoria $(X, p(X))$ definimos la **entropía** asociada mediante la expresión*

$$H = - \sum_x p(x) \log p(x) \quad (1.259)$$

- El signo negativo hace esta expresión positiva, debido a que $\log p(x) \leq 0$.
- El valor de H está acotado entre $H \in [0, \log(N)]$ donde N es el número de sucesos x posibles
- Si $p(x_i) = 0 \forall x_i$ excepto $p(x_0) = 1$ para un evento posible $\Rightarrow H = 0$
- Si $p(x_i) = 1/N$ es equiprobable $\Rightarrow H = \log(N)$ y este es el valor máximo.

Ejemplo

La entropía de la distribución de Bernoulli es

$$H = -p \log p - (1-p) \log(1-p) \quad (1.260)$$

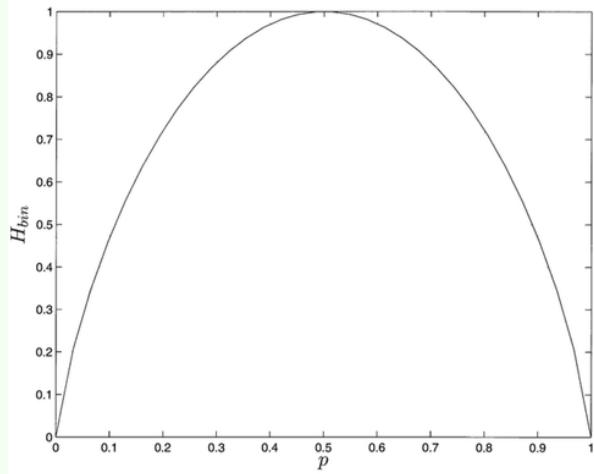


Figura 1.5: Entropía para una distribución de Bernouilli

Capítulo 2

Fundamentos de Mecánica Cuántica

En este capítulo vamos a hacer una introducción a los fundamentos de la Mecánica Cuántica. Este capítulo se basa en [1], que a su vez toma como referencias los capítulos 1 y 3 de [2], los capítulos 1, 3 y 4 de [3] y los capítulos 1-6 de [6].

Para más información sobre Mecánica Cuántica, se recomienda el famoso libro de Griffits [7].

2.1. Axiomas

2.1.1. ¿Qué es un axioma?

Axioma es una proposición tan clara y evidente que se admite sin demostración. Aplicado en matemáticas y otras ciencias, es cada uno de los principios indemostrables sobre los que, por medio de un razonamiento deductivo, se construye una teoría.

Ejemplo

Un ejemplo son los axiomas (postulados) de la geometría euclíadiana:

1. Es posible trazar una línea recta desde cualquier punto a cualquier otro punto.
2. Es posible prolongar un segmento de recta continuamente en ambas direcciones.
3. Es posible describir una circunferencia con cualquier centro y cualquier radio.
4. Es cierto que todos los ángulos rectos son iguales entre sí.
5. ("Postulado de las paralelas") Es cierto que, si una recta que cae sobre dos rectas hace que los ángulos interiores de un mismo lado sean menores que dos ángulos rectos, las dos rectas, si se producen indefinidamente, se intersecan en aquel lado en el que están los ángulos menores que los dos ángulos rectos.

2.1.2. Axiomas de la mecánica cuántica

La Mecánica Cuántica es una teoría fundamentada en unos pilares axiomáticos cuya selección admite cierta flexibilidad. La más aceptada constituye lo que se denomina la **interpretación de Copenhague**.

2.1.2.1. Vector de estado

En un instante, t , la **máxima información accesible** de un sistema está asociada a un **vector de estado** $|\psi\rangle$, de norma unidad, $\langle\psi|\psi\rangle = 1$, perteneciente a un espacio de Hilbert \mathcal{H}

- La dimensión de \mathcal{H} está relacionada con el número de grados de libertad del sistema.
- La **fase global** del vector de estado no contiene información: dos vectores que difieren en una fase global representan al mismo estado ningún experimento permite distinguirlos

$$|\psi\rangle \sim |\psi'\rangle = e^{i\varphi} |\psi\rangle . \quad (2.1)$$

- El vector de estado recibe también el nombre de **función de onda**.

2.1.2.2. Medidas

A una magnitud física medible, le está asociado un operador hermítico $A = A^\dagger$ que denominamos **observable**.

Los resultados de una medición no pueden dar como resultado nada más que uno de los valores propios de $A \Rightarrow \lambda_n$.

- Las magnitudes medibles son números reales, de ahí la exigencia de que A deba ser un operador hermítico.
- La información contenida en $|\psi\rangle$ es **intrínsecamente probabilística** (no hay variables ocultas). Los resultados de una medida son inciertos, pero ψ codifica una distribución de probabilidad.

2.1.2.3. Regla de Born

La probabilidad de obtener el autovalor λ_k como resultado de una cierta medición, viene dada por la expresión

$$p(\lambda_k) = |\langle \lambda_k | \psi \rangle|^2 \quad (2.2)$$

donde $|\lambda_k\rangle$ es el autovector asociado.

El valor $\langle \lambda_k | \psi \rangle$ se denomina **amplitud**.

La fórmula anterior es cierta cuando el autovalor en cuestión es *no degenerado*. La expresión correcta cuando λ_k tiene degeneración d_k es

$$p(\lambda_k) = \sum_{a=1}^{d_k} |\langle \lambda_k^a | \psi \rangle|^2 \quad (2.3)$$

2.1.2.4. Colapso de la función de ondas

Si el resultado de una medida ha sido λ_n , el estado del sistema, inmediatamente después de la medida, viene dado por el vector propio $|\lambda_n\rangle \in \mathcal{H}$, normalizado $|\langle \lambda_n | \lambda_n \rangle| = 1$

$$|\psi\rangle \xrightarrow{\lambda_n} |\lambda_n\rangle \quad (2.4)$$

2.1.2.5. Evolución en el tiempo

El cambio del estado del sistema $|\psi(t)\rangle$ es *determinista* y está gobernado por la **Ecuación de Schrödinger**

$$i\hbar \frac{\partial |\psi\rangle}{\partial t} = H |\psi\rangle , \quad (2.5)$$

donde el Hamiltoniano, H , es un operador hermítico que *caracteriza* el sistema, y \hbar es una *constante universal* denominada **constante de Planck**

$$\hbar = 1.054 \times 10^{-34} \text{ J} \cdot \text{s} \quad (2.6)$$

- J =Julio: Unidad en el Sistema Internacional para la Energía
- s =segundos: Unidad en el Sistema Internacional para el tiempo.

2.2. Bases ortogonales

En Mecánica Cuántica especificamos un estado usando las componentes de la expansión del mismo en una base ortogonal. Existen infinitas bases posibles para expresar un vector. ¿Cuál es la mejor base? La respuesta es que depende del proceso que estudiemos.

Por ejemplo, si el proceso es la *medida* de un cierto observable $A = A^\dagger$, según los postulados el resultado de nuestra medición sólo puede ser uno de los *valores propios* λ_i

$$A |\lambda_k\rangle = \lambda_k |\lambda_k\rangle , \quad (2.7)$$

donde $|\lambda_k\rangle$ son los autovectores $k = 1, 2, \dots$. Estos autovectores forman una base ortonormal (ver Ecs. (1.160) y (1.158)). Esta es la base natural adaptaba al proceso de medida de A . En particular, para conocer con qué probabilidad se producirá cada resultado, expandiremos el estado de la forma

$$|\psi\rangle = \sum_{i=1}^N a_i |\lambda_i\rangle . \quad (2.8)$$

El axioma III (regla de Born) afirma que la *probabilidad de aparición* del resultado λ_i es precisamente

$$p(\lambda_i) = |\langle \lambda_i | \psi \rangle|^2 = |a_i|^2 . \quad (2.9)$$

La forma de tener acceso experimental a los números $p(\lambda_i)$ es mediante la repetición estadística. Si efectuamos la medida un número n de veces con $n \rightarrow \infty$, y contamos la frecuencia de aparición de los distintos λ_i , esto es

$$p(\lambda_i) \approx \frac{n(\lambda_i)}{n} , \quad (2.10)$$

en el límite $n \rightarrow \infty$ dicha frecuencia experimental convergerá a la probabilidad teórica

$$\lim_{n \rightarrow \infty} \frac{n(\lambda_i)}{n} = p(\lambda_i) = |a_i|^2 . \quad (2.11)$$

Vemos que, mediante la repetición estadística, solamente podemos recuperar el módulo de las amplitudes

$$|a_i| = \lim_{n \rightarrow \infty} \sqrt{\frac{n(\lambda_i)}{n}} \quad (2.12)$$

Ejercicio 12 Generalizar las expresiones anteriores al caso en que los autovalores λ_k puedan ser d_k veces degenerados.

2.3. Medidas Estadísticas

2.3.1. Valor esperado

Vamos a reescribir la Ec. (2.3)

$$\begin{aligned} p(\lambda_k) &= \sum_{a=1}^{d_k} |\langle \lambda_k^a | \psi \rangle|^2 = \sum_{a=1}^{d_k} \langle \psi | \lambda_k^a \rangle \langle \lambda_k^a | \psi \rangle = \langle \psi | \left(\sum_{a=1}^{d_k} \lambda_k^a \langle \lambda_k^a | \right) | \psi \rangle \\ &= \langle \psi | P_k | \psi \rangle \end{aligned} \quad (2.13)$$

donde

$$P_k = \sum_{a=1}^{d_k} \lambda_k^a \langle \lambda_k^a | \quad (2.14)$$

es el **proyector sobre el subespacio propio** (ver Ec. (1.146)).

Por el **valor esperado** $\langle A \rangle$ del observable A en el estado $|\psi\rangle$, entendemos el *valor medio* de la variable aleatoria λ_n obtenida por el método descrito.

$$\begin{aligned} \langle A \rangle &= \sum_k \lambda_k p(\lambda_k) = \sum_k \lambda_k \langle \psi | P_k | \psi \rangle \\ &= \langle \psi | \left(\sum_k \lambda_k P_k \right) | \psi \rangle \end{aligned} \quad (2.15)$$

Reconocemos entre paréntesis la descomposición espectral de A . Llegamos así a la siguiente expresión para el valor esperado de un observable A en un estado $|\psi\rangle$:

Teorema 14 (*Valor esperado de un operador*) *El valor esperado de un operador es*

$$\langle A \rangle = \langle \psi | A | \psi \rangle \quad (2.16)$$

- En particular, las probabilidades mismas se pueden expresar como *valores esperados del proyector asociado*

$$p(\lambda_k) = \langle \psi | P_k | \psi \rangle = \langle P_k \rangle \quad (2.17)$$

- El valor esperado de un observable en uno de sus autoestados coincide con el autovalor asociado

$$\langle \lambda_i | A | \lambda_i \rangle = \langle \lambda_i | \lambda_i | \lambda_i \rangle = \lambda_i \langle \lambda_i | \lambda_i \rangle = \lambda_i \quad (2.18)$$

El espectro de un observable está formado por números reales que podemos ordenar $\lambda_{min} < \dots < \lambda_{max}$.

Teorema 15 *El valor esperado de un observable A está acotado entre sus valores propios mínimo y máximo*

$$\lambda_{min} \leq \langle \psi | A | \psi \rangle \leq \lambda_{max} \quad (2.19)$$

y las desigualdades anteriores se saturan en los autoestados correspondientes

2.3.2. Varianza y Desviación estándar

La **varianza** es la media de la desviación cuadrática de la variable aleatoria $(\lambda, p(\lambda))$ es decir

$$\sigma^2 = \overline{(\lambda_i - \bar{\lambda}_i)^2} = \bar{\lambda^2} - \bar{\lambda}^2 \quad (2.20)$$

de aquí se sigue, para la **desviación estándar**

$$\sigma = \sqrt{\langle \psi | A^2 | \psi \rangle - \langle \psi | A | \psi \rangle^2} \quad (2.21)$$

2.4. Evolución temporal

2.4.0.1. Ecuación de Schrödinger

El estado de un sistema $|\psi(t)\rangle$ posee la máxima información instantánea accesible de un sistema, es decir, a un tiempo dado t . Dado $|\psi(t_0)\rangle$ en un instante inicial t_0 , la evolución posterior obedece a la ecuación diferencial de Schrödinger

$$i\hbar \frac{d}{dt} |\psi(t)\rangle = H(t) |\psi(t)\rangle . \quad (2.22)$$

Nota

- El operador Hamiltoniano $H(t)$ contiene toda la información sobre la dinámica del sistema.
- Hemos enfatizado el hecho de que, en el caso más general, $H(t)$ puede depender del tiempo.

2.4.0.2. Conservación de la probabilidad

La hermiticidad de H se relaciona directamente con la interpretación probabilística de $|\psi\rangle$. Efectivamente, en *la evolución temporal se conserva la norma del vector de estado*

$$\begin{aligned} \frac{d}{dt} \langle \psi(t) | \psi(t) \rangle &= \left(\frac{d}{dt} \langle \psi(t) | \right) |\psi(t)\rangle + \langle \psi(t) | \left(\frac{d}{dt} |\psi(t)\rangle \right) \\ &= \left(\langle \psi(t) | \frac{i}{\hbar} H^\dagger \right) |\psi(t)\rangle + \langle \psi(t) | \left(-\frac{i}{\hbar} H |\psi(t)\rangle \right) \\ &= \frac{i}{\hbar} \langle \psi(t) | (H^\dagger - H) |\psi(t)\rangle = 0 \end{aligned} \quad (2.23)$$

Esto implica que la norma de $|\psi(t)\rangle$ se conserva $\Rightarrow 1 = \| |\psi(0)\rangle \| = \| |\psi(t)\rangle \|$

2.4.0.3. Operador de evolución

La conservación de la norma implica que la evolución $\| |\psi(0)\rangle \| = \| |\psi(t)\rangle \|$ es un *proceso unitario*. En otras palabras, debe existir un operador unitario

$$U(t, t_0)^{-1} = U(t, t_0)^\dagger \quad (2.24)$$

que lleve el estado inicial al actual a tiempo t

$$U(t, t_0) : |\psi(t_0)\rangle \rightarrow |\psi(t)\rangle = U(t, t_0) |\psi(t_0)\rangle \quad (2.25)$$

Nota: Propiedades de $U(t, t_0)$

El operador de evolución satisface las siguientes propiedades

- $U(t_0, t_0) = I$.

- transitividad: $U(t, t_1)U(t_1, t_0) = U(t, t_0)$
- invertibilidad: $U(t, t_0)^{-1} = U(t_0, t)$

Evidentemente, el operador de evolución debe estar relacionado con el Hamiltoniano, que es quien gobierna la dinámica. Veamos a ver que hay una ecuación de Schrödinger también para el operador de evolución

Teorema 16 *El operador de evolución satisface la siguiente ecuación de evolución*

$$i\hbar \frac{d}{dt} U(t, t_0) = H(t) U(t, t_0) \quad (2.26)$$

Demostración: Tomando la derivada temporal de la ecuación $|\psi(t)\rangle = U(t, t_0) |\psi(t_0)\rangle$, que define $U(t, t_0)$, tenemos para cada miembro

$$\begin{aligned} i\hbar \frac{d}{dt} |\psi(t)\rangle &= H(t) |\psi(t)\rangle = H(t) U(t, t_0) |\psi(t_0)\rangle \\ i\hbar \frac{d}{dt} |\psi(t)\rangle &= i\hbar \frac{d}{dt} U(t, t_0) |\psi(t_0)\rangle = i\hbar \left(\frac{d}{dt} U(t, t_0) \right) |\psi(t_0)\rangle \end{aligned} \quad (2.27)$$

Igualando ambas expresiones, y teniendo en cuenta que $|\psi_0\rangle$ es arbitrario, obtenemos la ecuación deseada. ■

2.4.0.4. Caso de H independiente del tiempo

Cuando H no depende del tiempo podemos dar una expresión analítica para el operador de evolución

Teorema 17 *Para un hamiltoniano H independiente del tiempo, el operador de evolución $U(t, t_0)$ es*

$$U(t, t_0) = \exp \left(-\frac{i}{\hbar} (t - t_0) H \right) \quad (2.28)$$

Demostración: Basta con demostrar que se satisface la ecuación de evolución y la condición de contorno

$$\begin{aligned} \frac{d}{dt} U(t, t_0) &= \frac{d}{dt} \exp \left(-\frac{i}{\hbar} (t - t_0) H \right) \\ &= \frac{d}{dt} \left(I + (t - t_0) \left(-\frac{i}{\hbar} H \right) + \frac{1}{2!} (t - t_0)^2 \left(-\frac{i}{\hbar} H \right)^2 + \dots \right) \\ &= 0 + \left(-\frac{i}{\hbar} H \right) + (t - t_0) \left(-\frac{i}{\hbar} H \right)^2 + \dots \\ &= \left(-\frac{i}{\hbar} H \right) \left(I + (t - t_0) \left(-\frac{i}{\hbar} H \right) + \dots \right) \\ &= \left(-\frac{i}{\hbar} H \right) \exp \left(-\frac{i}{\hbar} (t - t_0) H \right) \\ &= -\frac{i}{\hbar} H U(t, t_0) \end{aligned} \quad (2.29)$$

Además

$$U(t_0, t_0) = \exp\left(-\frac{i}{\hbar}(t_0 - t_0)H\right) = \exp(0) = I \quad (2.30)$$

■

Nota

La cantidad $-\frac{i}{\hbar}(t - t_0)H$ es adimensional

$$\frac{1}{J \cdot s} s \cdot J = 1 \quad (2.31)$$

Es decir, es un número puro. Por eso podemos exponenciarla. El operador de evolución $U(t, t_0)$ es, por tanto, también, una magnitud adimensional.

2.4.0.5. Evolución en la base de eutoestados de H

Sea $\{|n\rangle\}$ una base de autoestados de $H \Rightarrow H|n\rangle = E_n|n\rangle$, $n = 1, \dots, N$. En esta base H es una matriz H_{mn} diagonal

$$H_{mn} = \begin{bmatrix} E_0 & & & \\ & E_1 & & \\ & & \ddots & \\ & & & \ddots \end{bmatrix} = E_m \delta_{mn}. \quad (2.32)$$

Entonces la matriz de evolución es también diagonal

$$U_{mn} = \exp\left(-\frac{i}{\hbar}tH_{mn}\right) = \begin{bmatrix} e^{-\frac{i}{\hbar}tE_0} & & & \\ & e^{-\frac{i}{\hbar}tE_1} & & \\ & & \ddots & \\ & & & \ddots \end{bmatrix} = e^{-\frac{i}{\hbar}tE_m} \delta_{mn}. \quad (2.33)$$

En conclusión:

- La evolución temporal de un autoestado de la energía es trivial (es una fase global)

$$U(t, 0)|n\rangle = e^{-\frac{i}{\hbar}tE_n}|n\rangle \quad (2.34)$$

Es decir, un autoestado del hamiltoniano es un **estado estacionario**. Si nuestro sistema está en un autoestado del hamiltoniano, este no cambiará con el tiempo.

- Esta fase deja de ser trivial cuando afecta a una combinación lineal. La forma más eficaz de calcular la evolución de un estado arbitrario es expresarla en la base $\{|n\rangle\}$. Es decir, si a $t = 0$

$$|\psi(t=0)\rangle = \sum_n c_n |n\rangle \quad (2.35)$$

entonces, a tiempo t

$$|\psi(t)\rangle = \sum_n c_n e^{-\frac{i}{\hbar}E_n t} |n\rangle. \quad (2.36)$$

En esta base, la evolución temporal lo que hace es que las fases relativas de los autoestados evolucionen a velocidades diferentes.

Jupyter Notebook: Fundamentos de Mecánica Cuántica

Puede verse el Notebook [Fundamentos de Mecánica Cuántica](#).
El Notebook puede descargarse de [Github](#).

2.5. Estados Mezcla y Matriz Densidad

2.5.1. Estado puro y mezcla

Hasta ahora hemos estado estudiando casos **ideales**, donde los estados que tenemos son *perfectos*, en el sentido de que tenemos el estado que creemos tener con una probabilidad del 100 %. Esto en la vida real no es así. Cuando intentamos generar un estado cuántico, nunca vamos a generar estos estados perfectos, pues nuestros aparatos no son perfectos y nunca lo serán. Todo aparato lleva asociada una **incertidumbre**, es decir, una probabilidad de error.

A estos estados *perfectos* se los denomina **estados puros**, mientras que al resto de estados se los denomina **estados mezcla**.

2.5.1.1. Estos puros

Si sabemos *con seguridad* que el sistema se encuentra en un estado $|\psi\rangle \in \mathcal{H}$ *toda la incertidumbre* que queda es cuántica.

Estado puro:

Si con *certeza total* podemos afirmar que el estado de un sistema está descrito por un vector $|\psi\rangle \in \mathcal{H}$ decimos que nuestro sistema se encuentra en un **estado puro**.

Tomemos un estado **superposición**

$$|\psi\rangle = a|0\rangle + b|1\rangle \quad (2.37)$$

con $a, b \neq 0$. La probabilidad de medir el estado $|0\rangle$ será positiva

$$p(0) = |\langle 0|\psi\rangle|^2 = |\langle 0|(a|0\rangle + b|1\rangle)|^2 = |a\langle 0|0\rangle + b\langle 0|1\rangle|^2 = |a|^2 \quad (2.38)$$

la de medir el estado $|1\rangle$ también

$$p(1) = |\langle 1|\psi\rangle|^2 = |\langle 1|(a|0\rangle + b|1\rangle)|^2 = |a\langle 1|0\rangle + b\langle 1|1\rangle|^2 = |b|^2 \quad (2.39)$$

y la de medir el estado $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ será

$$\begin{aligned} p(+) &= |\langle +|\psi\rangle|^2 = \frac{1}{2} |(\langle 0| + \langle 1|)(a|0\rangle + b|1\rangle)|^2 = \frac{1}{2} |(a\langle 0|0\rangle + b\langle 1|1\rangle)|^2 \\ &= \frac{1}{2}|a+b|^2 = \frac{1}{2}(a+b)(a^*+b^*) \\ &= \frac{1}{2}(|a|^2 + |b|^2 + 2\text{Re}(ab^*)) \end{aligned} \quad (2.40)$$

El término que no tiene signo definido $2\text{Re}(ab^*)$ se denomina *interferencia*. Es el responsable de que $p(+) = 0$ se pueda anular aun cuando $a, b \neq 0$. Concretamente cuando $a = 1 = -b$

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle \quad (2.41)$$

y los estados son ortogonales $|\langle \psi|+\rangle|^2 = |\langle -|+\rangle|^2 = p(|-\rangle \rightarrow |+)) = 0$, obteniendo $p(+) = 0$.

2.5.1.2. Estados mezcla

¿Qué ocurre si no podemos tener la certeza de que el estado que describe el sistema sea $|\psi\rangle$?

Por ejemplo: supongamos que,

- a la salida de un polarizador de Stern Gerlach, encontramos que el estado es $|+\rangle$,
- pero la fiabilidad nominal de dicho aparato es del 90 %.
- Eso quiere decir que, con un 10 % de probabilidad, el estado emergente ha sido, en realidad, $|-\rangle$.

Nuestro polarizador se comporta como una *variable aleatoria* $\{ \{ |+\rangle, 0.9 \}, \{ |-\rangle, 0.1 \} \}$ de la que obtendremos, cada vez, un estado u otro con distinta probabilidad.

Nota

Véase que esta incertidumbre debida al error del polarizador que genera el estado es clásica, no cuántica.

Estados mezcla:

En la situación más general, un sistema estará en un **estado mezcla** asociado a una variable aleatoria

$$\{ |\psi_\alpha\rangle, p_\alpha \} \quad (2.42)$$

que indica que, con probabilidad p_α , el estado real del sistema es $|\psi_\alpha\rangle$

- Un estado mezcla no es un vector en el espacio de Hilbert, sino que es una variable aleatoria donde el espacio muestral son vectores en el espacio de Hilbert.
- Es importante recalcar que las probabilidades $p_\alpha \in \mathbb{R}$ son incertidumbres clásicas, y por tanto $p_k \in [0, 1]$, $\sum_\alpha p_\alpha = 1$.
- El estado mixto contiene al estado puro como caso particular en el que para un cierto valor $\alpha = i$, tenemos $p_i = 1$ y el resto son cero.

2.5.2. Operador densidad

El formalismo matemático que nos permite lidiar con los estados mezcla pasa por usar la **matriz densidad**.

Supongamos que a nuestro sistema le podemos asignar un estado mixto $\{ |\psi_\alpha\rangle, p_\alpha \}$. La probabilidad, $p(\lambda_n)$, de encontrar un autovector $|\lambda_n\rangle$ como resultado de la medida de un observable A , debe ser la *suma ponderada* de probabilidades de esa medida en cada uno de los estados $|\psi_\alpha\rangle$

$$\begin{aligned} p(\lambda_n) &= \sum_\alpha p_\alpha |\langle \lambda_n | \psi_\alpha \rangle|^2 = \sum_\alpha p_\alpha \langle \lambda_n | \psi_\alpha \rangle \langle \psi_\alpha | \lambda_n \rangle \\ &= \langle \lambda_n | \left(\sum_\alpha p_\alpha |\psi_\alpha\rangle \langle \psi_\alpha| \right) | \lambda_n \rangle \\ &\equiv \langle \lambda_n | \rho | \lambda_n \rangle \end{aligned} \quad (2.43)$$

El resultado anterior muestra la aparición en escena de un *nuevo operador* formado con los datos del colectivo aleatorio $\{ |\psi_\alpha\rangle, p_\alpha \}$

$$\rho = \sum_\alpha p_\alpha |\psi_\alpha\rangle \langle \psi_\alpha| = \sum_\alpha p_\alpha P_\alpha \quad (2.44)$$

ρ se denomina **matriz densidad** y consiste en una suma ponderada de proyectores sobre cada uno de los subespacios generados por los estados posibles. Este es el objeto matemático que caracteriza un estado mezcla.

En particular, como acabamos de ver, la probabilidad de medir el autovalor λ es el *valor esperado* del operador densidad en dicho estado

$$p(\lambda_n) = \langle \rho \rangle_{|\lambda_n\rangle} = \langle \lambda_n | \rho | \lambda_n \rangle \quad (2.45)$$

Operador densidad:

Un operador, ρ , podrá ser el operador densidad de un sistema si cumple los siguientes requisitos

- es hermítico $\rho = \rho^\dagger$
- es semidefinido positivo
- tiene traza unidad $\text{tr}\rho = 1$

Claramente la expresión dada anteriormente cumple con todos estos requisitos.

- Hermiticidad: se ve gracias a que los p_α son probabilidades (números reales)

$$\rho^\dagger = \left(\sum_\alpha p_\alpha |\psi_\alpha\rangle\langle\psi_\alpha| \right)^\dagger = \sum_\alpha p_\alpha^* |\psi_\alpha\rangle\langle\psi_\alpha| = \rho \quad (2.46)$$

- De hecho escrito en esta base, la matriz que representa ρ es diagonal

$$\rho = \begin{bmatrix} p_1 & 0 & \dots & 0 \\ 0 & p_2 & \dots & 0 \\ & & \vdots & \\ 0 & 0 & \dots & p_n \end{bmatrix} \quad (2.47)$$

Por tanto los números $p_\alpha \geq 0$ son los autovalores, que son no-negativos, lo cuál caracteriza un operador semidefinido positivo.

- Finalmente la traza es la unidad debido a que p_α son probabilidades

$$\text{Tr } \rho = \sum_\alpha p_\alpha = 1 \quad (2.48)$$

Nota

Es importante darse cuenta de que un estado mezcla **no se puede escribir como un vector de estado**. Solo los estados puros pueden ser escritos así.

Por ejemplo, en el caso que veíamos antes de un polarizador que generaba el estado $|+\rangle$ con una probabilidad del 90 %, este se comportaba como una variable aleatoria $\{|+\rangle, 0.9\}, \{|-\rangle, 0.1\}$. El estado que sale de este polarizador es un estado mezcla que se puede expresar con siguiente matriz densidad

$$\rho = [0.9|+\rangle\langle+|] + [0.1|-\rangle\langle-|] \quad (2.49)$$

Este estado **no es el mismo** que el estado puro

$$|\psi\rangle = \sqrt{0.9}|+\rangle + \sqrt{0.1}|-\rangle \quad (2.50)$$

Es más, podemos escribir la matriz densidad de este segundo estado para ver que es diferente:

$$\begin{aligned} \rho_{|\psi\rangle} &= |\psi\rangle\langle\psi| = (\sqrt{0.9}|+\rangle + \sqrt{0.1}|-\rangle)(\sqrt{0.9}\langle+| + \sqrt{0.1}\langle-|) \\ &= [0.9|+\rangle\langle+|] + [0.1|-\rangle\langle-|] + [\sqrt{0.9}\sqrt{0.1}|+\rangle\langle-|] + [\sqrt{0.9}\sqrt{0.1}|-\rangle\langle+|] \end{aligned} \quad (2.51)$$

Nota

La matriz densidad de un estado mezcla **no tiene porque ser diagonal**. La Ec. (2.44) es simplemente el caso en el que estamos en una base donde la matriz es diagonal.

2.5.2.1. Matriz densidad de un estado puro

La matriz densidad es un formalismo más general que el del vector estado, que sólo se aplica en el caso de estados puros. En efecto, la matriz densidad asociada a un estado puro $|\psi_0\rangle$ se obtiene haciendo todas las probabilidades $p_{\alpha \neq 0} = 0$ y $p_0 = 1$. Entonces el colectivo $\{|\psi_0\rangle, p_0 = 1\}$ tiene asociado el operador

$$\rho = |\psi_0\rangle\langle\psi_0| = \begin{bmatrix} 1 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix} \quad (2.52)$$

Es evidente que esta expresión cumple, a mayores, la propiedad que caracteriza a un **proyector**

$$\rho^2 = \rho \quad (2.53)$$

y esto es una ecuación que *sólo verifican las matrices densidad asociadas a estados puros*.

Por el contrario, para un estado mezcla podemos ver que $\rho^2 \neq \rho$

$$\rho^2 = \sum_{\alpha} p_{\alpha}^2 |\psi_{\alpha}\rangle\langle\psi_{\alpha}| \neq \sum_{\alpha} p_{\alpha} |\psi_{\alpha}\rangle\langle\psi_{\alpha}| = \rho \quad (2.54)$$

Podemos extraer esta información de forma *independiente de la base* usando la función traza Tr

- Si $|\psi_{\alpha}\rangle$ son vectores ortonormales, $\text{Tr}\rho = \sum p_{\alpha} = 1$ pero $\text{Tr}\rho^2 = \sum_{\alpha} p_{\alpha}^2 \leq 1$
- La igualdad $\text{Tr}\rho^2 = 1$ sólo se producirá si $p_{\alpha} = 1$ para un solo α .

Teorema 18 *Un estado ρ será puro (mezcla) si y solo si $\text{Tr}\rho^2 = 1$ ($\text{Tr}\rho^2 < 1$)*

2.5.2.2. Estado maximalmente mezclado

En el extremo opuesto de un estado puro, encontramos un estado **maximalmente mezclado**, en el cual *todos* los proyectores aparecen de manera equiprobable $\{|\psi_{\alpha}\rangle, p_{\alpha} = \frac{1}{d}\}, \alpha = 1, \dots, d$

$$\rho = \sum_{\alpha=1}^d \frac{1}{d} |\psi_{\alpha}\rangle\langle\psi_{\alpha}| = \frac{1}{d} I = \begin{bmatrix} \frac{1}{d} & & & \\ & \frac{1}{d} & & \\ & & \ddots & \\ & & & \frac{1}{d} \end{bmatrix} \quad (2.55)$$

Por tanto, es una matriz diagonal proporcional a la identidad.

2.5.2.3. Estado mezcla parcial

Entre los dos extremos mencionados anteriormente, estado puro y estado maximalmente mezclado, se sitúa cualquier estado ρ genérico. Si escribimos

$$\rho = \sum_{\alpha} p_{\alpha} |\psi_{\alpha}\rangle\langle\psi_{\alpha}| \quad (2.56)$$

el grado de mezcla es proporcional a la uniformidad en la distribución de valores p_{α} .

2.5.2.4. Estado de Gibbs

Un caso muy importante de mezcla parcial es el **estado de Gibbs**, que describe un sistema cuando alcanza el *equilibrio térmico a una temperatura T*. En este caso, los estados $\{|\psi_\alpha\rangle = |E_\alpha\rangle\}$ son la *base de autoestados del operador Hamiltoniano*,

$$H|E_\alpha\rangle = E_\alpha|E_\alpha\rangle \quad (2.57)$$

y los autovalores E_α , con $\alpha = 1, 2, \dots, d$, son los *niveles de energía* del sistema.

Cuando un sistema se encuentra en contacto con un baño térmico a temperatura T el estado de energía no está bien definido, sino que es una mezcla denominada **colectivo canónico**

$$\left\{ |E_\alpha\rangle, p_\alpha = e^{-E_\alpha/k_B T} \right\} \quad (2.58)$$

Los coeficientes $p_\alpha = e^{-E_\alpha/k_B T}$ se denominan *coeficientes de Boltzmann* codifican la probabilidad de hallarse en el estado (nivel de energía) $|E_\alpha\rangle$.

La matriz densidad que describe el estado de este sistema es el denominado **estado de Gibbs**

$$\rho(T) = \frac{1}{Z} \sum_{\alpha=1}^d e^{-E_\alpha/k_B T} |E_\alpha\rangle \langle E_\alpha| = \frac{1}{Z} \begin{bmatrix} e^{-E_0/k_B T} & & \\ & e^{-E_1/k_B T} & \\ & & \ddots \end{bmatrix}, \quad (2.59)$$

donde k_B es la constante de Boltzmann, y $Z = \sum_\alpha e^{-E_\alpha/k_B T}$ es la normalización necesaria para que $\text{tr}\rho(T) = 1$. La **energía del sistema** a cada temperatura, vendrá dada por un promedio sobre todos los autoestados pesados por la matriz densidad a dicha temperatura

$$\langle E \rangle_T = \text{tr}(\rho(T)H) \quad (2.60)$$

Nota: autoestados del Hamiltoniano

Como ya comentamos, un autoestado de un operador es un estado donde el observable (operador) toma un valor concreto (el autovalor). Los autoestados del operador hamiltoniano son especiales, pues son los **niveles de energía**. Es decir, los autovalores son las **energías**.

Ejercicio 13 Probar que recuperamos los casos puro y maximalmente mezclado en los límites siguientes

- $\rho(T=0) = |E_0\rangle\langle E_0|$
- $\rho(T=\infty) = \frac{1}{d}I$

Ejercicio 14 Genera aleatoriamente un estado de Gibbs a temperatura T y grafica los valores de $p_\alpha(T)$ para distintos valores de T (toma $k_B = 1$).

2.5.2.5. Entropía de Von Neumann

La entropía de Von Neumann $S(\rho)$ es una medida del *grado de mezcla* que hay en una matriz densidad. Se define como sigue

$$S(\rho) = -\text{tr}(\rho \log \rho). \quad (2.61)$$

Usando la descomposición espectral de $\rho = \sum_\alpha p_\alpha |\psi_\alpha\rangle\langle\psi_\alpha|$, donde $\sum_i p_\alpha = 1$, la entropía de Von Neumann resulta ser

$$S(\rho) = -\sum_\alpha p_\alpha \log p_\alpha \quad (2.62)$$

es decir, la entropía de la variable aleatoria $\{|\psi_\alpha\rangle, p_\alpha\}$.

- Vemos que $S = 0$ cuando el estado que representa ρ es puro ya que $p_\alpha = \delta_{\alpha 1}$.
- Por el contrario, en un estado máximamente mezclado $p_\alpha = 1/d$, $S = \log d$ alcanza su máximo valor.

2.5.3. Medidas en estados mezcla

El escenario ahora es un estado general ρ sobre el que efectuamos una medida asociada a un observable que admite una descomposición espectral

$$A = \sum_n \lambda_n |\lambda_n\rangle\langle\lambda_n| = \sum_n \lambda_n P_n \quad (2.63)$$

Según el axioma del colapso de la función de onda, después de una medida, si el resultado ha sido λ_n , el estado mezcla ρ colapsa al estado puro $\rho \rightarrow |\lambda_n\rangle\langle\lambda_n|$. En términos de matriz densidad

$$\rho \rightarrow |\lambda_n\rangle\langle\lambda_n| = \frac{P_n \rho P_n}{p(\lambda_n)}. \quad (2.64)$$

Desarrollando un poco el numerador

$$P_n \rho P_n = |\lambda_n\rangle\langle\lambda_n| \rho |\lambda_n\rangle\langle\lambda_n| = p(\lambda_n) |\lambda_n\rangle\langle\lambda_n| \quad (2.65)$$

Vamos a manipular un poco la expresión de la probabilidad que hemos visto en la Ec. (2.43):

$$\begin{aligned} p(\lambda_n) &= \langle\lambda_n|\rho|\lambda_n\rangle = \sum_i \langle\lambda_n|i\rangle \langle i|\rho|\lambda_n\rangle = \sum_i \langle i|\rho|\lambda_n\rangle \langle\lambda_n|i\rangle = \sum_i \langle i|\left(\rho|\lambda_n\rangle\langle\lambda_n|\right)|i\rangle \\ &\equiv \text{tr}(\rho P_n) \end{aligned} \quad (2.66)$$

Es decir,

$$p(\lambda_n) = \langle\lambda_n|\rho|\lambda_n\rangle = \text{tr}(\rho P_n) \quad (2.67)$$

Ahora podemos hallar el valor esperado de A

$$\langle A \rangle_\rho = \sum_n \lambda_n p(\lambda_n) = \sum_n \lambda_n \text{tr}(\rho P_n) = \text{tr} \left(\rho \sum_n \lambda_n P_n \right) = \text{tr}(\rho A) \quad (2.68)$$

Teorema 19 *El valor esperado de un observable A en un estado general ρ es*

$$\langle A \rangle_\rho = \text{tr}(\rho A) \quad (2.69)$$

2.5.4. Estado mezcla bipartito

De forma similar, podemos definir la matriz densidad en sistemas multipartitos. Si consideramos por simplicidad un espacio $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2$, el siguiente operador $\rho \in \text{L}(\mathcal{H})$

$$\rho = \sum_{ia,jb} \rho_{ia,jb} |e_{ia}\rangle\langle e_{jb}| \quad (2.70)$$

podrá describir el estado mezcla de un sistema bipartito si la matriz $\rho_{ia,jb}$ verifica las tres condiciones que definen un operador densidad.

Nota

No confundir estado **mezcla** con estado **entrelazado** Un estado $|\psi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$ puede ser entrelazado, y aun así $\rho = |\psi\rangle\langle\psi|$ ser puro.

En el caso de vectores $|\psi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$, ya hemos visto la división que existe entre vectores factorizables y entrelazados. Para estados $\rho \in L(\mathcal{H}_1 \otimes \mathcal{H}_2)$ tenemos más posibilidades:

- **Factorizable:** si ρ es un operador factorizable

$$\rho = \rho_1 \otimes \rho_2 . \quad (2.71)$$

En este caso, $\rho_{ia,jb} = \rho_{1,ij}\rho_{2,ab}$ y decimos que, en este estado, los subsistemas 1 y 2 están *descorrelacionados*.

- **Separable:** Si es combinación lineal de operadores factorizables

$$\rho = \sum_a p_a \rho_{1a} \otimes \rho_{2a} \quad (2.72)$$

con $p_a \in [0, 1]$ y $\sum_a p_a = 1$. En este caso existen correlaciones clásicas entre ambos sistemas.

- **No-separable:** en cualquier otro caso. En este caso hay correlaciones clásicas y cuánticas entre ambos sistemas.

2.5.4.1. Trazas parciales

A partir de un estado bipartito $\rho \in L(\mathcal{H}_1 \otimes \mathcal{H}_2)$ podemos definir estados $\rho_1 \in L(\mathcal{H}_1)$ y $\rho_2 \in L(\mathcal{H}_2)$ mediante la operación de *traza parcial*

$$\rho_1 = \text{Tr}_{\mathcal{H}_2} \rho = \sum_{a=1}^{d_2} \left\langle e_a^{(2)} \right| \rho \left| e_a^{(2)} \right\rangle , \quad \rho_2 = \text{Tr}_{\mathcal{H}_1} \rho = \sum_{a=1}^{d_1} \left\langle e_a^{(1)} \right| \rho \left| e_a^{(1)} \right\rangle \quad (2.73)$$

Sobre las matriz densidad $\rho_{ia,jb}$ las trazas parciales dan matrices

$$\rho_{1,ij} = \sum_{a=1}^{d_2} \rho_{ia,ja} , \quad \rho_{2,ab} = \sum_{i=1}^{d_1} \rho_{ia,ib} \quad (2.74)$$

Ejercicio 15 Sea un estado bipartito $\rho \in L(\mathcal{H}_1 \otimes \mathcal{H}_2)$ y $A = A_1 \otimes I$ un observable que sólo depende del subsistema 1. Demuestra que

$$\langle A \rangle_\rho = \text{tr}(\rho_1 A_1) \quad (2.75)$$

Supongamos un *estado puro bipartito* $\rho = |\psi\rangle\langle\psi| \in L(\mathcal{H}_1 \otimes \mathcal{H}_2) \Rightarrow \rho = \rho^2$ es un proyector. Después de tomar la traza parcial $\rho \rightarrow \rho_1 = \text{tr}_{\mathcal{H}_2} \rho$ hay dos opciones

- Si $|\psi\rangle$ es *factorizable* $\Rightarrow \rho_1$ permanece *puro*

$$|\psi\rangle = |\varphi\rangle \otimes |\phi\rangle \Rightarrow \rho_1 = \text{Tr}_{\mathcal{H}_2} |\psi\rangle\langle\psi| = |\varphi\rangle\langle\varphi| = \rho_1^2 \quad (2.76)$$

- Si $|\psi\rangle$ es *entrelazado* $\Rightarrow \rho_1$ se vuelve *mezclado*. Podemos ver esto con la descomposición de Schmidt

$$|\psi\rangle = \sum_{a=1}^r \sqrt{s_a} |f_a\rangle \left| \tilde{f}_a \right\rangle \Rightarrow \rho_1 = \text{Tr}_{\mathcal{H}_2} |\psi\rangle\langle\psi| = \sum_{a=1}^r s_a |f_a\rangle\langle f_a| \neq \rho_1^2 \quad (2.77)$$

En resumen: **bajo traza parcial, el entrelazamiento induce mezcla.**

Parte II

Fundamentos de Computación Cuántica

Capítulo 3

Qubits

3.1. Definición y bases

3.1.1. Definición de Qubit

Un concepto fundamental en computación cuántica (QC) es el concepto de **qúbit** o *bit cuántico*. En esta sección veremos que es un qúbit, que es la base computación y como se representa un qubit en la esfera de Bloch.

Como sabemos, la computación clásica se de basa en el **bits**, es decir, en ceros o unos. Con estos bits construimos cadenas de ceros y unos sobre las que se aplican puertas lógicas (AND, OR, NOT,...) que transforman esta cadena en otra. Definamos de forma matemática un bit:

Definición 23 *Un bit es una variable real a que puede tomar valores en $\mathbb{Z}_2 = \{0, 1\}$.*

Un qúbit es el equivalen en computación cuántica de un bit. Un qúbit es un **sistema cuántico de dos niveles de energía o dos estados no degenerado**.

Nota: Estado degenerado

Dentro de un sistema cuántico dos estados se denominan **degenerados** si tiene la misma energía

El estado de menor energía será el equivalente al cero y el estado de mayor energía será equivalente al uno. Definiéndolo de forma rigurosa:

Definición 24 *Un cúbít es un espacio de Hilbert complejo de dimensión dos, $|v\rangle \in \mathcal{H} \sim \mathbb{C}^2$*

Quizás estas dos definiciones parecen muy pomposas, pero tienen la capacidad de transmitirnos en dos simples frases la abismal diferencia entre un bit y un qúbit. Mientras un bit es simplemente un número entero que puede tomar dos valores, un qúbit puede tomar los infinitos valores del espacio del Hilbert de dos dimensiones. Es decir, bit es una variable discreta (digitalizada), mientras que un qúbit es una variable continua (analógica). Como veremos mas adelante, la digitalización de la computación cuántica se da en el momento de la **medida**.

Cabe destacar que cualquier sistema cuyo estado instantáneo se pueda describir mediante un vector $|v\rangle \in \mathbb{C}^2$ puede albergar físicamente un **qúbit**. A efectos prácticos, se trata de sistemas que pueden estar y conmutar entre dos niveles de energía bien definidos de manera controlada. Esto lleva a que haya varias implementaciones de los qubits (qubits superconductores, trampas de iones, ...).

Se está explorando también el uso de sistemas cuánticos de más niveles que dos, como lo **qútrit** de dimensión 3. De forma genérica, a un qubit de d dimensiones se le denomina **qúdit**.

Nota: El espín del electrón como un qubit

El ejemplo más típico es el de un espín de un electrón, que de forma natural tiene dos niveles (la proyección $1/2$ y la proyección $-1/2$). Sin embargo, como queremos que los dos estados estén bien diferenciados, debemos pensar en el espín de un electrón sumergido en un campo magnético, de forma que la proyección del espín a favor del campo se ve favorecida sobre la otra. Es decir, pasamos de tener un dos estados degenerados (de igual energía) a tener dos estados con diferente energía.

3.1.2. Bases computacionales y bases X e Y .

En esta sección vamos a ver tres bases para representar nuestro qubits: la base Z (o computacional), la X y la Y . Estas dos últimas bases estarán referidas a la primera.

3.1.2.1. Base computacional o base Z

Como acabamos de ver, los qubits viven en un espacio de Hilbert complejo de dos dimensiones, así que vamos a necesitar dos vectores ortonormales para definir una **base**.

Definir por primera vez una base tiene una cierta componente de arbitrariedad. Por ejemplo, cuando tenemos un vector en un plano podemos describirlo definiendo sus coordenadas respecto a unos ejes x e y . La gracia está en que estos ejes podemos elegirlos como queramos (siempre que sean independientes). Sin embargo, una vez fijados unos ejes, cualquier nuevo par de ejes que introduzcamos (cualquier nueva base) está definido a partir de los primeros.

Nota: Mas sobre el espín

Como comentamos en una nota anterior, los autoestados $|+\rangle$ y $|-\rangle$ de la matriz σ_z corresponden a las proyecciones $1/2$ y $-1/2$ del espín sobre el eje z . Cabe preguntarnos, porque es tan común medir las proyecciones sobre este eje? En realidad, es simplemente por convenio y por la arbitrariedad a la hora de elegir la dirección de este eje. Simplemente, se mide la proyección del espín sobre un eje cualquiera y se considera que este es el eje z . Como ahora tenemos definido uno de los ejes, si medimos en otro eje diferente, esta nueva medida ya no podrá ser la proyección sobre el eje z .

La base que se usa habitualmente en computación cuántica se denomina **base computacional** y se denota como $B = \{|0\rangle, |1\rangle\}$. Podemos ver la clara analogía con los bits. Esta base es **ortonormal**, es decir

$$\langle 0|1\rangle = \langle 1|0\rangle = 0 \quad \text{y} \quad \langle 0|0\rangle = \langle 1|1\rangle = 1. \quad (3.1)$$

Una manera de especificar una base es decir cuál es el operador que diagonaliza. A la base computacional se la suele denominar habitualmente como **base Z** , pues es la base que diagonaliza la matriz de Pauli σ_z

$$|+\rangle = |0\rangle_Z \equiv \boxed{|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}}, \quad |-\rangle = |1\rangle_Z \equiv \boxed{|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}} \quad (3.2)$$

Es decir, esta base la forman los autoestados del operador $Z = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ con autovalores ± 1 .

3.1.2.2. Base X

También podemos usar la base de autoestados del operador $X = \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ con autovalores ± 1 :

$$\begin{aligned} |x_+\rangle &= |0\rangle_X \equiv \boxed{|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}}, \\ |x_-\rangle &= |1\rangle_X \equiv \boxed{|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}}. \end{aligned} \quad (3.3)$$

3.1.2.3. Base Y

También podemos usar la base de autoestados del operador $Y = \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ con autovalores ± 1 :

$$\begin{aligned} |y_+\rangle &= |0\rangle_Y \equiv \boxed{|+i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix}}, \\ |y_-\rangle &= |1\rangle_Y \equiv \boxed{|-i\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix}}. \end{aligned} \quad (3.4)$$

Ejercicio 16 Escribe las matrices de cambio de la base $Z \rightarrow X$, $Z \rightarrow Y$ y $X \rightarrow Y$.

3.2. El estado de un qúbit y la esfera de Bloch

3.2.1. Parametrización del estado de un qúbit.

Como ya sabemos, en un bit clásico podemos almacenar un valor, o 0 o 1. La pregunta ahora es, ¿cuánta información podemos almacenar en un qúbit? De forma genérica, el **estado de un qubit** se puede representar como

$$|u\rangle = a|0\rangle + b|1\rangle, \quad \text{con } a, b \in \mathbb{C} \quad (3.5)$$

Nota

Habitualmente, se denomina simplemente **qúbit** al **estado del qúbit**.

En principio, como a y b son complejos tenemos 4 grados de libertad. Sin embargo, tenemos una ligadura, la normalización del estado

$$|a|^2 + |b|^2 = 1, \quad (3.6)$$

lo que nos deja 3 grados de libertad.

Nota: Normalización de un estado

El estado completo de un sistema cuántico debe estar normalizada a 1, pues la probabilidad de medir algo alguno de los estados debe de ser el 100 %

Además, como en mecánica cuántica las fases globales no tienen significado, perdemos otro grado de libertad. Nos quedamos pues con dos grados de libertad (reales), es decir, solo necesitamos dos parámetros reales para parametrizar el estado de un qubit. La parametrización más habitual se presenta en el siguiente lemma:

Nota: las fases gobales

Al multiplicar un estado por una fase de la forma $e^{i\phi}$, el estado es el mismo

Lemma 4 El qúbit $|u\rangle$ más general se puede representar, en la base computacional $\{|0\rangle, |1\rangle\}$, usando dos números reales (ángulos) $\theta \in [0, \pi)$ y $\varphi \in [0, 2\pi)$

$$|u\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle \quad (3.7)$$

Demostración: Para demostrar el lema escribimos un vector de estado general usando la representación polar para las componentes complejas

$$\begin{aligned} |u\rangle &= u_0 |0\rangle + u_1 |1\rangle \\ &= a_0 e^{ib_0} |0\rangle + a_1 e^{ib_1} |1\rangle \\ &= e^{ib_0} (a_0 |0\rangle + a_1 e^{i(b_1 - b_0)} |1\rangle) \\ &\sim (a_0 |0\rangle + a_1 e^{i(b_1 - b_0)} |1\rangle) \end{aligned} \quad (3.8)$$

donde hemos hecho uso de la irrelevancia de una fase global para descartar e^{ib_0} en la última línea.

Ahora a_1 y a_0 no son números independientes sino que verifican $a_0^2 + a_1^2 = 0$ para que $\| |u\rangle \| = 1$. Esta ecuación se puede resolver en términos de un ángulo $\theta/2$ tal que

$$a_0 = \cos \frac{\theta}{2}, \quad a_1 = \sin \frac{\theta}{2} \quad (3.9)$$

Por su parte, de los números b_1, b_2 sólo la diferencia

$$\varphi = b_1 - b_2 \quad (3.10)$$

es relevante para especificar $|u\rangle$. ■

En componentes escribimos equivalentemente

$$|u\rangle = \begin{bmatrix} \cos \frac{\theta}{2} \\ e^{i\varphi} \sin \frac{\theta}{2} \end{bmatrix} \quad \text{con } \theta \in [0, \pi) \text{ y } \varphi \in [0, 2\pi) \quad (3.11)$$

3.2.2. La esfera de Bloch

Viendo la Ec. (3.11) vemos que podemos interpretar los parámetros θ y φ como ángulos y representar el vector de estado del qúbit en sobre una *esfera unidad*. Dicho de otro modo, podemos representar cada estado de un qúbit como un **punto** sobre una esfera de radio unidad: la **esfera de Bloch**.

- $\theta \in [0, \pi)$ el ángulo **azimutal** se mide desde el eje z . De modo que $\theta = \pi/2$ es el **ecuador de la esfera de Bloch**.
- $\varphi \in [0, 2\pi)$ el ángulo **polar** en torno al eje z , medido a partir del plano XZ

Veamos donde se sitúan en la esfera los elementos de las bases Z , X e Y que vimos anteriormente

- $(\theta, \phi) = (0, \phi) \Rightarrow e^{i\phi} |0\rangle = |0\rangle$
- $(\theta, \phi) = (\pi, \phi) \Rightarrow e^{i\phi} |1\rangle = |1\rangle$

- $(\theta, \phi) = (\pi/2, 0) \Rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |0\rangle_X \equiv |+\rangle$
- $(\theta, \phi) = (\pi/2, \pi) \Rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |1\rangle_X \equiv |-\rangle$
- $(\theta, \phi) = (\pi/2, \pi/2) \Rightarrow \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) = |0\rangle_Y \equiv |+i\rangle$
- $(\theta, \phi) = (\pi/2, 3\pi/2) \Rightarrow \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) = |1\rangle_Y \equiv |-i\rangle$

Podemos ver la representación de estos estados sobre la esfera de Bloch en la Fig. 3.1.

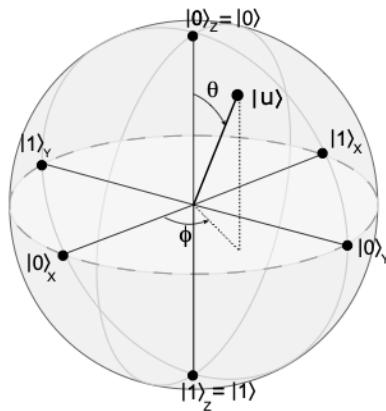


Figura 3.1: Esfera de Bloch. Figura tomada de [1]

Véase que no estamos representando vectores en un espacio tridimensional, sino algo completamente diferente. En el espacio \mathbb{R}^3 , dos vectores son ortogonales si forman un ángulo de 90° . Sin embargo, en la esfera de Bloch *dos vectores son ortonormales (en realidad, ortonormales) si apuntan en direcciones opuestas*. Por ejemplo, ya comentamos que los estados $|0\rangle$ y $|1\rangle$ forman una base ortonormal. Si nos fijamos, en la esfera de Bloch el estado $|0\rangle$ está en el polo norte y el estado $|1\rangle$ en el polo sur. Lo mismo pasa con $|0\rangle_X$ y $|1\rangle_X$ y con $|0\rangle_Y$ y $|1\rangle_Y$.

Jupyter Notebook: Vectores de estado y la esfera de Bloch

Puede verse el Notebook [Vectores de estado y la esfera de Bloch](#). En el se muestra como dibujar estados en la esfera de Bloch usando Qiskit en Python.

El Notebook puede descargarse de [Github](#).

Capítulo 4

Puertas simples

Ya comentamos que la computación clásica se usan circuitos con puertas lógicas para hacer los cálculos. En computación cuántica que sigue la misma filosofía. En este capítulo vamos a empezar a ver las **puertas cuánticas** y como actúan sobre los qúbit.

4.1. Rotaciones en la esfera de Bloch

Como todos los estados de un qúbit se pueden representar en la esfera de Bloch, cualquier operación que hagamos sobre el qúbit se puede interpretar como una rotación en la esfera de Bloch.

Nota: operadores unitarios

Esto esta relacionado con que todas las puertas (operadores) que se usan en computación cuántica son **operadores unitarios**, es decir, transformaciones que preservan la norma del vector de estados. Esto es lógico, pues las probabilidades tienen que seguir sumando el 100 %.

Teorema 20 *El operador que efectúa una **rotación de ángulo** $\alpha \in [0, 2\pi)$ en torno al **eje que marca un vector unitario** $\hat{\mathbf{n}}$ es el siguiente*

$$R_{\hat{\mathbf{n}}}(\alpha) = \exp\left(-i\frac{\alpha}{2}\hat{\mathbf{n}} \cdot \boldsymbol{\sigma}\right) = \cos\frac{\alpha}{2}I - i\sin\frac{\alpha}{2}\hat{\mathbf{n}} \cdot \boldsymbol{\sigma} \quad (4.1)$$

donde $\boldsymbol{\sigma} = (\sigma_x, \sigma_y, \sigma_z)$ son las matrices de Pauli.

Como podemos ver en el teorema anterior, solo necesitamos un ángulo y un eje (un vector unitario) para definir una rotación en la esfera de Bloch. Podemos ver una imagen de esto en la Fig. 4.1.

Nota: Sentido de rotación

El sentido de la rotación que produce $R_{\hat{\mathbf{n}}}(\alpha)$ en torno al eje $\hat{\mathbf{n}}$, viene dado por la **regla de la mano derecha** o, también, **anti-horario**.

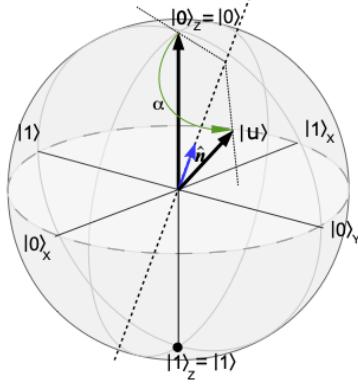


Figura 4.1: Una rotación en la esfera de Bloch representada por un eje de rotación y el ángulo que se rota entorno al mismo. Figura tomada de [1]

En la Ec. (4.1) el término $\hat{\mathbf{n}} \cdot \boldsymbol{\sigma}$ se refiere a la multiplicación de un vector por un vector de matrices. Esto es, multiplicar cada matriz por un elemento del vector y sumar las 3 matrices resultantes. Es decir, en la Ec. (4.1) tenemos la suma de 4 matrices 2×2 . La matriz resultante es

$$R_{\hat{\mathbf{n}}}(\alpha) = \begin{pmatrix} \cos \frac{\alpha}{2} - i n_z \sin \frac{\alpha}{2} & (-i n_x - n_y) \sin \frac{\alpha}{2} \\ (-i n_x + n_y) \sin \frac{\alpha}{2} & \cos \frac{\alpha}{2} + i n_z \sin \frac{\alpha}{2} \end{pmatrix} \quad (4.2)$$

Ejercicio 17 Calcular la matriz (4.2) a partir de (4.1)

4.1.1. Rotaciones en X, Y, Z

Veamos los casos particulares de las rotaciones entorno a los ejes x , y y z .

$$\begin{aligned} \hat{\mathbf{n}} = (0, 0, 1) &\Rightarrow R_z(\alpha) = \begin{pmatrix} e^{-i\alpha/2} & 0 \\ 0 & e^{i\alpha/2} \end{pmatrix} \\ \hat{\mathbf{n}} = (0, 1, 0) &\Rightarrow R_y(\alpha) = \begin{pmatrix} \cos \alpha/2 & -\sin \alpha/2 \\ \sin \alpha/2 & \cos \alpha/2 \end{pmatrix} \\ \hat{\mathbf{n}} = (1, 0, 0) &\Rightarrow R_x(\alpha) = \begin{pmatrix} \cos \alpha/2 & -i \sin \alpha \\ -i \sin \alpha/2 & \cos \alpha/2 \end{pmatrix} \end{aligned} \quad (4.3)$$

4.1.2. Parametrización de Euler

Hay otra parametrización para las rotaciones mucho más común en física y es la **parametrización de Euler**. A diferencia de la anterior, esta no necesita definir ningún eje extra usando un vector, sino que simplemente consiste en tres rotaciones, con tres ángulos, entorno a dos ejes coordinados: primero entorno al eje z , después en torno al eje y y finalmente entorno al eje z otra vez. Es decir

$$R_z(\phi)R_y(\theta)R_z(\varphi) = e^{-\frac{i}{2}(\phi+\varphi)}U(\theta, \phi, \varphi), \quad (4.4)$$

donde

$$U(\theta, \phi, \varphi) = \begin{pmatrix} \cos \frac{\theta}{2} & -e^{i\varphi} \sin \frac{\theta}{2} \\ e^{i\phi} \sin \frac{\theta}{2} & e^{i(\varphi+\phi)} \cos \frac{\theta}{2} \end{pmatrix} \quad (4.5)$$

y donde θ , ϕ y φ se denominan **ángulos de Euler**. Véase que nuevamente podemos ignorar la fase global que nos sale en la Ec. (4.4) y quedarnos solo con la matriz $U(\theta, \phi, \varphi)$.

Nota

Podría usarse otro orden de ejes, como por ejemplo xyx o yzy .

Podemos ver que la acción del operador (4.5) sobre el estado $|0\rangle$ nos da la expresión genérica del estado de un qúbit que vimos en la Ec. (3.7):

$$U(\theta, \phi, \varphi)|0\rangle = U(\theta, \phi, \varphi) \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \cos \frac{\theta}{2}|0\rangle + e^{i\varphi} \sin \frac{\theta}{2}|1\rangle \quad (4.6)$$

También podemos ver que la acción del operador de este operador sobre la base computacional $\{|0\rangle, |1\rangle\}$ nos la una base alineada con el eje (θ, ϕ) :

$$U(\theta, \phi, \varphi)|0\rangle = \begin{bmatrix} \cos \frac{\theta}{2} \\ e^{i\phi} \sin \frac{\theta}{2} \end{bmatrix}, \quad U(\theta, \phi, \varphi)|1\rangle = \begin{bmatrix} -e^{i\varphi} \sin \frac{\theta}{2} \\ e^{i(\varphi+\phi)} \cos \frac{\theta}{2} \end{bmatrix} \quad (4.7)$$

4.2. Puertas simples

La computación clásica se basa en la descomposición de algoritmos complejos en una serie de puertas lógicas elementales. Veremos que lo mismo ocurre con la computación cuántica.

Por puertas simples entendemos un conjunto de **operadores unitarios** que se utilizan con frecuencia en la computación cuántica. Vamos a ver las puertas simples sobre 1 qúbit.

4.2.1. Dos formas de escribir las matrices 2x2.

Primero hagamos un breve alto en el camino para vez una segunda forma de escribir las matrices 2×2 en computación cuántica. Dada una matriz genérica A podemos escribirla como

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = a_{11}|0\rangle\langle 0| + a_{12}|0\rangle\langle 1| + a_{21}|1\rangle\langle 0| + a_{22}|1\rangle\langle 1| \quad (4.8)$$

Esta segunda forma es cómoda para hacer operaciones.

Nota: producto de interno de estado de la base

Como los elemtos de la base son ortonormales cumplen:

$$\langle 0|0\rangle = \langle 1|1\rangle = 1, \quad \langle 0|1\rangle = \langle 1|0\rangle = 0. \quad (4.9)$$

Ahora podemos ver como actúa una matriz genérica A sobre un elemento de la base, por ejemplo, $|0\rangle$

$$\begin{aligned} A|0\rangle &= a_{11}|0\rangle\langle 0|0\rangle + a_{12}|0\rangle\langle 1|0\rangle + a_{21}|1\rangle\langle 0|0\rangle + a_{22}|1\rangle\langle 1|0\rangle \\ &= a_{11}|0\rangle\langle 0|0\rangle + a_{21}|1\rangle\langle 0|0\rangle = a_{11}|0\rangle + a_{21}|1\rangle = \begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix} \end{aligned}$$

4.2.2. Puertas de fase

4.2.2.1. P_α con $\alpha \in [0, 2\pi)$.

Se trata de una rotación entorno al eje Z y se escribe de la forma

$$P(\alpha) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix} = |0\rangle\langle 0| + e^{i\alpha}|1\rangle\langle 1| \quad (4.10)$$

Aplicando sobre un estado genérico tenemos

$$P(\alpha)|u\rangle = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix} \begin{bmatrix} \cos \theta/2 \\ e^{i\phi} \sin \theta/2 \end{bmatrix} = \begin{bmatrix} \cos \theta/2 \\ e^{i(\phi+\alpha)} \sin \theta/2 \end{bmatrix} = |v\rangle$$

Ya vimos en la Ec. (4.3) una puerta para rotar entorno al eje z . Podemos ver que a efectos prácticos la puerta P_α es lo mismo que la puerta $R_z(\alpha)$, pues se diferencian solo en una fase global

$$P_\alpha \equiv e^{i\alpha/2}R_z(\alpha)$$

4.2.2.2. K_α

Esta puerta es trivial pero a veces se usa en algunas demostraciones. Se trata simplemente de una puerta de fase global

$$K(\alpha) = e^{i\alpha} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = e^{i\alpha}(|0\rangle\langle 0| + |1\rangle\langle 1|) = e^{i\alpha}I \quad (4.11)$$

4.2.3. Puertas discretas

4.2.3.1. X, Y, Z

Tres puertas muy usadas en computación cuántica con las siguientes

$$\boxed{X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \sigma_x = |1\rangle\langle 0| + |0\rangle\langle 1|},$$

$$\boxed{Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = \sigma_y = i|1\rangle\langle 0| - i|0\rangle\langle 1|}, \quad (4.12)$$

$$\boxed{Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \sigma_z = |0\rangle\langle 0| - |1\rangle\langle 1|},$$

donde hemos remarcado la igualdad con las *matrices de Pauli* (ver sección 1.3.10).

Ejercicio 18 Relacionar X, Y, Z con $R_x(\alpha), R_y(\alpha)$ y $R_z(\alpha)$ para algún valor de α .

Nota

Como las puertas X, Y y Z son las matrices de Pauli, estas son hermíticas ($A = A^\dagger$) y además cumplen que son iguales a su inversa (ver Ec. (1.182)). Esto implica que aplicar dos veces seguidas una de estas puertas es lo mismo que aplicar la identidad. Además, los autovalores de las matrices de Pauli son ± 1 (ver Ec. (1.183)).

4.2.3.2. S, T

Cualquier potencia U^k de un operador unitario es otro operador unitario. Esto es fácil de demostrar cuando $k = 2$ pero es cierto en el caso general $k \in \mathbb{R}$. Así obtenemos

$$\boxed{S = Z^{1/2} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{pmatrix}} \quad , \quad \boxed{T = S^{1/2} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}}. \quad (4.13)$$

4.2.3.3. H

La puerta de Hadamard, H , es la primera puerta **genuinamente cuántica** en el sentido de que lleva un estado de la base a una superposición coherente

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle \quad , \quad H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle \quad (4.14)$$

Podemos escribir este operador en la base $H = H_{ij}|i\rangle\langle j|$

$$\begin{aligned} H &= |+\rangle\langle 0| + |-\rangle\langle 1| \\ &= \frac{1}{\sqrt{2}}(|0\rangle\langle 0| + |1\rangle\langle 0| + |0\rangle\langle 1| - |1\rangle\langle 1|) \end{aligned}$$

de lo que obtenemos la representación matricial

$$\boxed{H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}} \quad (4.15)$$

Nota: otra expresión para la Hadamard

En cálculos posteriores encontraremos muy útil la siguiente representación de la acción de H

$$\boxed{H|x\rangle = \frac{1}{\sqrt{2}} \sum_{y=0,1} (-1)^{xy} |y\rangle} \quad (4.16)$$

Como cualquier puerta, la acción de H puede visualizarse como una rotación en la esfera de Bloch. En este caso una es una rotación de π radianes en torno a un eje diagonal situado a 45° entre el eje x y el eje y . Esta rotación permuta los ejes x y z y cambia de sentido el eje y .

$$\hat{\mathbf{n}} = \frac{1}{\sqrt{2}}(1, 0, 1) \Rightarrow R_{\hat{\mathbf{n}}}(\pi) = -i \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = -iH \sim H \quad (4.17)$$

Nota: La puerta de Hadamard es hermítica

Puede verse fácilmente que la puerta de Hadamard es hermítica e igual a su inversa. Es decir, aplicar dos veces seguidas la puerta de Hadamard es como aplicar la identidad.

4.2.4. Descomposición

Una noción muy importante en computación cuántica es la descomposición de una puerta en producto de otras más simples.

Para el caso de H , un poco de visión espacial muestra que su acción equivale a la composición de

- una rotación de $\pi/2$ radianes sobre el eje y
- seguida de una rotación de π radianes en torno al eje x .

Lo demostramos algebraicamente (despreciando fases globales)

$$\begin{aligned} R_x(\pi)R_y\left(\frac{\pi}{2}\right) &= \begin{pmatrix} 0 & -i \\ -i & 0 \end{pmatrix} \begin{pmatrix} \cos \pi/4 & -\sin \pi/4 \\ \sin \pi/4 & \cos \pi/4 \end{pmatrix} = \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} = \frac{-i}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = -iH \sim H \end{aligned}$$

Ejercicio 19 Encontrar los ángulos θ, ϕ, φ que hay que verifican las siguientes idendidades

$$U(\theta, \phi, \varphi) = H \quad , \quad U(\theta, \phi, \varphi) = SH \quad (4.18)$$

4.3. Circuitos Cuánticos (1 qubit)

En un circuito cuántico un qubit se representa como una linea horizontal y las puertas que se aplican sobre el mismo se representan como cajas que contiene los datos del operador asociado. Por ejemplo, la aplicación del operador $U(\theta, \phi, \varphi)$ sobre un qubit en un estado $|\psi\rangle$, i.e.,

$$|\psi\rangle \rightarrow U(\theta, \phi, \varphi) |\psi\rangle ,$$

se representa mediante el circuito siguiente

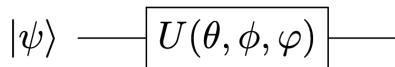


Figura 4.2: Circuito para la operación $|\psi\rangle \rightarrow U(\theta, \phi, \varphi) |\psi\rangle$. Figura tomada de [1]

La concatenación de puertas se corresponde con la **composición de operadores**, es decir, con la **multiplicación de las matrices** asociadas.

Nota: Orden de las puertas

El **orden** en el que aparecen los operadores en la composición es el opuesto al que se aprecia en el circuito. Así por ejemplo a la composición de operadores

$$|\psi\rangle \rightarrow TH |\psi\rangle \quad (4.19)$$

le corresponde un circuito en el que H está a la izquierda de T . Recordemos que, por norma general, el producto de matrices no es conmutativo.

4.3.1. Matriz de un circuito.

Todo circuito se corresponde con un operador unitario que se obtiene componiendo todos los operadores que figuran en el. Por ejemplo, para el ejemplo $|\psi\rangle \rightarrow TH |\psi\rangle$, el circuito correspondiente representa el operador unitario U al que le corresponde la matriz obtenida por multiplicación

$$U = TH \quad \rightarrow \quad U_{ij} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ e^{i\pi/4} & -e^{i\pi/4} \end{bmatrix} \quad (4.20)$$

4.3.2. Simulador de un estado con qiskit.

Veremos como construir y simular un estado con qiskit en la sección 5.1.3, después de haber visto como medir.

Capítulo 5

Medidas, Parte I: Medida de 1 qúbit

5.1. En la base computacional

5.1.1. Superposiciones, medidas y colapso.

En mecánica cuántica podemos tener un estado que sea la **superposición** de varios estados. Por ejemplo, el estado de un qúbit puede ser la superposición de los estados $|0\rangle$ y $|1\rangle$, esto es

$$|u\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle \quad (5.1)$$

Sin embargo, al medir solo obtenemos **uno de estos estados de la superposición**. La probabilidad de medir cada uno de estos estados que forman la superposición es igual al **módulo cuadrado del coeficiente que lo acompaña en el vector $|\psi\rangle$** (el módulo cuadrado de la amplitud de ese estado). En este caso vemos que la probabilidad de medir el estado $|0\rangle$ es $|\alpha|^2$ y la de medir $|1\rangle$ es $|\beta|^2$.

El aparato de medida estándar en computación cuántica asigna valores $\{0, 1\}$ a los kets $|0\rangle$ y $|1\rangle$ de la base computacional. Su representación en un circuito podemos verla en la Fig. 5.1. En esta figura vemos que tenemos dos formas de medir: una destructiva y otra no destructiva. La explicación es simple: dependiendo de como esté construido el ordenador cuántico, nuestra medida destruirá o no el estado del qúbit al medirlo.

En el caso en el que el estado no se destruya, lo que pasará será que el qúbit **colapsará** al estado medido. Esto es los famosos **colapso de la función de ondas** en mecánica cuántica. Si, por ejemplo, al medir nuestro qúbit obtenemos que estado $|0\rangle$, la superposición desaparece y nuestro qubit pasa a estar en el estado $|0\rangle$.

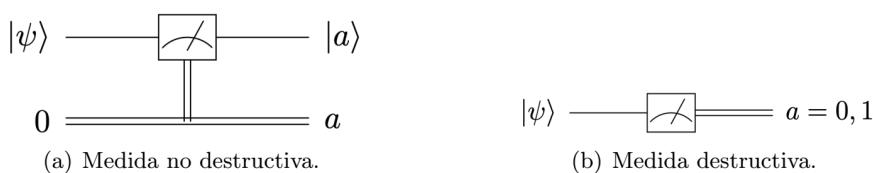


Figura 5.1: Medidas de un qubit en el estado $|\psi\rangle$ en un circuito cuántico. El resultado puede ser $a = 0$ o $a = 1$. La linea simple representa un qubit, mientras que la linea doble representa un bit clásico. Figura tomada de [1]

5.1.2. Sobre medir en la base computaciones.

Los elementos de la base computacional $|a\rangle \in \{|0\rangle, |1\rangle\}$, son autoestados del **observable** $Z = \sigma_z$, cuyos autovalores son $+1$ y -1 respectivamente, cumpliendo

$$Z|0\rangle = +|0\rangle , \quad Z|1\rangle = -|1\rangle \quad (5.2)$$

Podemos unificar ambos resultados como: $Z|a\rangle = (-1)^a|a\rangle$, con $a = \{0, 1\}$. Es decir, lo que estamos haciendo es medir el observable Z (por ejemplo, medir el espín en el eje z) y si obtenemos el autovalor $+1$ decimos que tenemos un 0 , mientras que si medimos el autovalor -1 , decimos que tenemos un 1 .

5.1.3. Código de Qiskit: simulación de un estado y medida.

Jupyter Notebook: [Circuitos de un qubit y HardWare real](#)

En [Circuitos de un qubit y HardWare real, sección 3.1: Creación y medición de circuitos de 1 qubit con Qiskit](#) puede verse como construir circuitos de un qubit en qiskit, como añadir puertas de un qubit, como añadir los medidores y como hacer la simulación. Además de como guardar figuras de los circuitos y de como generar histogramas con los resultados.

El Notebook puede descargarse de [Github](#).

5.1.4. Código de Qiskit: ejecución en un ordenador real.

Jupyter Notebook: [Circuitos de un qubit y HardWare real](#)

En [Circuitos de un qubit y HardWare real, sección 3.2: Mandar trabajos a un ordenador real de IBM](#) puede verse como mandar circuitos a ejecutar en ordenadores cuánticos reales de IBM. El Notebook puede descargarse de [Github](#).

5.2. La moneda cuántica

Vamos a ver aquí a modo de ejemplo un experimento simple: **la moneda cuántica**.

El resultado de tirar una moneda al aire es una variable aleatoria con dos resultados equiprobables: cara y cruz. Es irrelevante si analizamos el resultado cada tirada o cada dos, o tres tiradas. Las frecuencias relativas de caras y cruces, siempre serán próximas a $1/2$. Es decir, podemos tirar la moneda, recogerla sin mirarla, volver a tirar, y las probabilidades no cambian.

Podemos imaginar un experimento similar con un qubit, donde cara $\rightarrow 0$ y cruz $\rightarrow 1$ son los resultados posibles de la medida en la base Z . Como al tirar la moneda, mientras esta está en el aire podemos pensar que está en “una superposición equiprobable del cara y cruz”, el hecho de **tirar la moneda** en computación cuántica será aplicar el operador H (ver Ec. (4.15)).

Haciendo esta consideración, podemos ver que no es lo mismo tirar la moneda 1 vez y mirar

$$|0\rangle \xrightarrow{\text{tirar}} H|0\rangle = |+\rangle \xrightarrow{\text{medir}} p(0) = p(1) = 0.5 \quad (5.3)$$

que tirarla dos veces y mirar

$$|0\rangle \xrightarrow{\text{tirar}} H|0\rangle \xrightarrow{\text{tirar}} H^2|0\rangle = |0\rangle \xrightarrow{\text{medir}} p(0) = 1 , p(1) = 0 \quad (5.4)$$

El objetivo de este experimento es simplemente ver que ciertas puertas son sus propias inversas y que cuando aplicamos las aplicamos un número par de veces seguidas, es como si no aplicáramos nada. Otra cosa que podemos ver con este experimento es que **las medidas intermedias alteran el resultado**. Esto podemos verlo si colocmos un medidor en medio entre las puertas H y vemos como varía el resultado.

5.2.1. Código de Qiskit.

Jupyter Notebook: [Circuitos de un qubit y HardWare real](#)

En [Circuitos de un qubit y HardWare real](#), sección 3.3: La moneda cuántica puede verse el experimento de la moneda cuántica.

El Notebook puede descargarse de [Github](#).

5.3. Medidas en una base general

5.3.1. Bases X, Y, Z

A parte de la base computacional $\{|0\rangle, |1\rangle\}$, es muy necesario y frecuente el uso de otras bases ortonormales como $\{|+\rangle, |-\rangle\}$ o $\{|+i\rangle, |-i\rangle\}$. Todas ellas diagonalizan algún operador de Pauli y, por tanto, puede servir para construir aparatos de medida que discriminen entre sus estados

$$\begin{aligned} Z|0\rangle &= +|0\rangle, & Z|1\rangle &= -|1\rangle \\ X|+\rangle &= +|+\rangle, & X|-\rangle &= -|-\rangle \\ Y|+i\rangle &= +|+i\rangle, & Y|-i\rangle &= -|-i\rangle. \end{aligned} \quad (5.5)$$

En la práctica, solo solemos contar con un aparato de medida en la base computacional de autoestados de σ_z . La pregunta ahora es cómo podríamos utilizar dicho aparato para efectuar medidas en las bases de autoestados de σ_x y σ_y .

Por ejemplo desearíamos definir un aparato de medida que “leyese” los valores 0 y 1 para los autoestados $|\pm\rangle$ del operador $\sigma_x = X$ (ver Fig. 5.2).

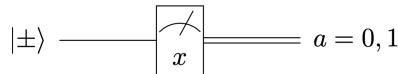


Figura 5.2: Medidor en la base X .

Análogamente, queremos lo mismo para los autoestados $|\pm i\rangle$ de $\sigma_y = Y$. La clave está en un cambio de base (en rotar la base en la que queremos medir a la base $\{|0\rangle, |1\rangle\}$).

$$\begin{aligned} |+\rangle &= H|0\rangle, & |-\rangle &= H|1\rangle \\ |+i\rangle &= SH|0\rangle, & |-i\rangle &= SH|1\rangle. \end{aligned} \quad (5.6)$$

Estas expresiones son fácilmente invertibles para deshacer el cambio de base:

$$\begin{aligned} H|+\rangle &= |0\rangle, & H|-\rangle &= |1\rangle \\ HS^\dagger|+i\rangle &= |0\rangle, & HS^\dagger|-i\rangle &= |1\rangle \end{aligned} \quad (5.7)$$

Es decir, la idea es llevar a cabo una operación que nos lleve los elementos de la base en la que queremos medir (los elementos $\{|+\rangle, |-\rangle\}$ o $\{|+i\rangle, |-i\rangle\}$) a los elementos de la base computacional $\{|0\rangle, |1\rangle\}$. Esto es precisamente lo que conseguimos con los operadores H y HS^\dagger en las Ecs. (5.7). De las relaciones anteriores se deduce que

$$\begin{aligned} X &= HZH \\ Y &= SHZHS^\dagger \end{aligned} \quad (5.8)$$

Nota

Verifiquemos por ejemplo

$$Y |-i\rangle = (SHZHS^\dagger)(SH|0\rangle) = SHZ|1\rangle = SH(-|1\rangle) = -SH|1\rangle = -|-i\rangle \quad (5.9)$$

Ejercicio 20 Comprueba estos cambios de base (las Ecs. (5.7) y (5.8)).

Estas relaciones nos permite definir el aparato de medida efectivo en las direcciones x e y . Podemos verlos en la Fig. 5.3.

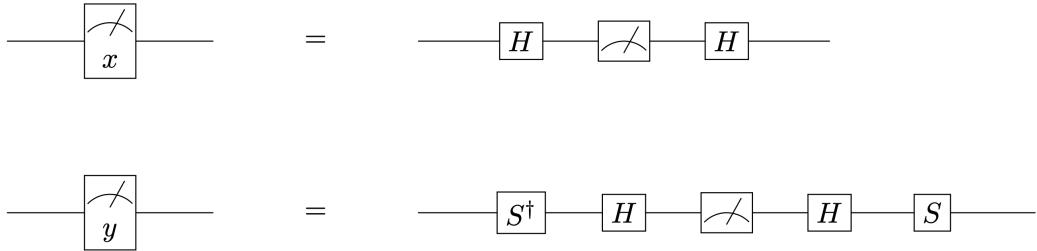


Figura 5.3: Equivalencias de los medidores en las bases X e Y con el medidor en Z . Cuando la medida es destructiva (la inmensa mayoría de las veces), no hace falta añadir los operadores después del aparato de medida.

5.3.2. Base arbitraria

Vamos a generalizar el análisis anterior. Para ello podemos usar la matriz de rotación en la parametrización de Euler de la Ec. (4.5), que nos lleva la base $\{|0\rangle, |1\rangle\}_{\hat{n}}$ asociado a un vector

$$\hat{n}(\theta, \phi) = \sin \theta \cos \phi \hat{x} + \sin \theta \sin \phi \hat{y} + \cos \theta \hat{z} \quad (5.10)$$

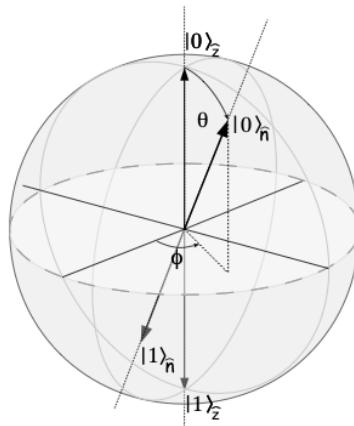


Figura 5.4: Base computacional y base \hat{n} en la esfera de Bloch

Es decir, tenemos el cambio de base

$$|a\rangle_{\hat{n}} = U(\theta, \phi, 0) |a\rangle_{\hat{z}}, \quad \text{con } a = 0, 1 \quad (5.11)$$

que verifica la ecuación de autovalores

$$\sigma_z |a\rangle_{\hat{z}} = (-1)^a \Rightarrow (\hat{n} \cdot \boldsymbol{\sigma}) |a\rangle_{\hat{n}} = (-1)^a |a\rangle_{\hat{n}} \quad (5.12)$$

Por esta razón, podemos etiquetar este operado como

$$U(\theta, \phi, 0) \equiv U(z \rightarrow \hat{n}) \quad (5.13)$$

para ver claramente que lo que hace es llevarnos de la base Z a una base definida por \hat{n} .

El circuito de la Fig. 5.5 simula un aparato de medición en la base $\{|a\rangle_{\hat{n}}\}_{a=0,1}$. Podemos ver que lo que se aplica antes del medidor en Z no es $U(z \rightarrow \hat{n})$ sino su inversa, $U(z \rightarrow \hat{n})^\dagger$. Esto es porque lo que tenemos que hacer es llevar la base \hat{n} a la base Z .

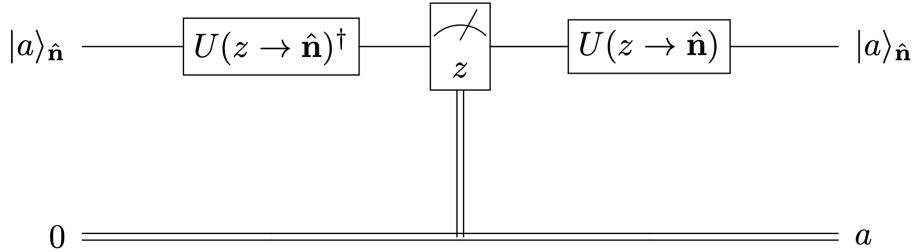


Figura 5.5: Medida en un eje arbitrario \hat{n} .

Nota

Puede verse fácilmente que el caso genérico que acabamos de ver incluye, como es lógico los dos casos anterior:

$$U(\pi/2, 0, \pi) = H : |a\rangle_{\hat{z}} \rightarrow |a\rangle_{\hat{x}} \quad (5.14)$$

$$U(\pi/2, \pi/2, \pi) = SH : |a\rangle_{\hat{z}} \rightarrow |a\rangle_{\hat{y}} \quad (5.15)$$

5.4. Valores esperados.

Un estado es un objeto probabilístico que tiene una interpretación a través de los coeficientes (de las amplitudes de probabilidad) y por tanto, esa interpretación probabilística está ligada a una base. Es decir, las amplitudes de probabilidad no tiene sentido hasta que definimos una base en la que medir. Si por ejemplo, vamos a medir en la base Z , lo conveniente es expandir el estado en la base Z

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle \quad (5.16)$$

de forma que los coeficientes adquieran una interpretación probabilística. Puede decirse que las amplitudes “no existen” hasta que decidimos en qué base medimos. El acceso experimental a estas amplitudes se da mediante estadística de las medidas

$$p_0 = \frac{n_0}{N} = |c_0|^2 = |\langle 0|\psi\rangle|^2 \quad p_1 = \frac{n_1}{N} = |c_1|^2 = |\langle 1|\psi\rangle|^2. \quad (5.17)$$

donde N es el número total de medidas, n_0 es el número de medidas que arrojaron el valor 0 (en realidad +1) y n_1 es el número de medidas que arrojaron el valor 1 (en realidad -1). Este procedimiento de reconstrucción es la base de la **tomografía cuántica**.

5.4.1. Valor esperado de un observable arbitrario (operador hermítico).

Vamos a ver como se mide el valor esperado de un observable arbitrario A en un estado arbitrario $|\Psi\rangle$. Estos valores esperados se denotan de la siguiente forma: $\langle Z \rangle_\Psi$.

Cualquier observable sobre un cíbit cumple $A = A^\dagger$ (es hermítico) con lo que puede expresarse en la base

$$A = aI + n_x X + n_y Y + n_z Z. \quad (5.18)$$

Los coeficientes se obtienen haciendo uso de las relaciones $\frac{1}{2} \text{tr}(\sigma_i \sigma_j) = \delta_{ij}$ y de $\text{tr}(\sigma_i) = 0$, de las cuales se obtiene

$$\boxed{a = \frac{1}{2} \text{tr}(A), \quad n_i = \frac{1}{2} \text{tr}(A\sigma_i)}, \quad (5.19)$$

Entonces, podremos obtener el valor esperado de A si somos capaces de medir los valores esperados de X , Y y Z .

$$\boxed{\langle A \rangle_\Psi = a + n_x \langle X \rangle_\Psi + n_y \langle Y \rangle_\Psi + n_z \langle Z \rangle_\Psi} \quad (5.20)$$

5.4.1.1. $\langle Z \rangle_\Psi$

Los estados de la base computacional son autoestados del operador Z con autovalor ± 1

$$Z |0\rangle = +|0\rangle \quad Z |1\rangle = -|1\rangle \quad (5.21)$$

Dado un estado $|\Psi\rangle = c_0 |0\rangle + c_1 |1\rangle$, la medida repetida arroja de forma aleatoria los valores propios de $Z \rightarrow \pm 1$ con frecuencias relativas n_0^Z y n_1^Z . Por definición, el valor esperado de dicha variable es,

$$\boxed{\langle Z \rangle_\Psi = (+1) \frac{n_0^Z}{N} + (-1) \frac{n_1^Z}{N}} \quad (5.22)$$

donde N es el número total de medidas, n_0^Z es el número de veces que hemos medido el estado $|0\rangle$ (el autovalor $+1$) y n_1^Z es el número de veces que hemos medido el estado $|1\rangle$ (el autovalor -1).

Para medir $\langle Z \rangle_\Psi$ necesitamos el **medidor habitual en computación cuántica** (ver Fig. 5.1)

5.4.1.2. $\langle X \rangle_\Psi$

Igualmente, si medimos el estado $|\Psi\rangle$ con un **medidor asociado al operador** $X = HZH$ (ver Fig. 5.3) la repetición arrojará igualmente una muestra aleatoria de valores propios de $X \rightarrow \pm 1$ con frecuencias relativas n_0^X y n_1^X . El valor esperado de X se obtiene del promedio

$$\boxed{\langle X \rangle_\Psi = (+1) \frac{n_0^X}{N} + (-1) \frac{n_1^X}{N}} \quad (5.23)$$

donde N es el número total de medidas, n_0^X es el número de veces que hemos medido el valor 0 (el estado $|+\rangle$ con autovalor $+1$) y n_1^X es el número de veces que hemos medido 1 (el estado $|-\rangle$ con autovalor -1).

5.4.1.3. $\langle Y \rangle_\Psi$

Igualmente, si medimos el estado $|\Psi\rangle$ con un **medidor asociado al operador** $X = SHZHS^\dagger$ (ver Fig. 5.3) la repetición arrojará igualmente una muestra aleatoria de valores propios de $X \rightarrow \pm 1$ con frecuencias relativas n_0^Y y n_1^Y . El valor esperado de X se obtiene del promedio

$$\boxed{\langle Y \rangle_\Psi = (+1) \frac{n_0^Y}{N} + (-1) \frac{n_1^Y}{N}} \quad (5.24)$$

donde N es el número total de medidas, n_0^Y es el número de veces que hemos medido el valor 0 (el estado $|+i\rangle$ con autovalor $+1$) y n_1^Y es el número de veces que hemos medido 1 (el estado $| -i\rangle$ con autovalor -1).

Ejercicio 21 Genera un observable arbitrario hermítico $2 \times 2 A$ y obtén los coeficientes n_i de la descomposición de A . Inicializa un vector $|\Psi\rangle$ aleatorio y calcula el valor esperado $\langle A \rangle_\Psi$. (Todo a mano, con papel y bolígrafo)

5.4.2. Valor esperado de un operador unitario (no necesariamente hermítico)

En la sección anterior vimos como calcular el valor esperado de un operador hermítico aprovechando que los operadores de esta clase se pueden descomponer en función de las matrices de Pauli. Cuando el operador no es hermítico, no podemos llevar a cabo esta descomposición. Veamos como podemos calcular el valor esperado de un operador unitario genérico V .

En general, en computación cuántica partimos teniendo los qubit inicializados en el estado $|0\rangle$. Si queremos tener el estado $|\Psi\rangle$ debemos realizar una serie de operaciones sobre el qubit para tenerlo. Estas operaciones pueden representarse por un operador U tal que $|\Psi\rangle = U|0\rangle$.

Supongamos ahora que queremos calcular el valor esperado de un operador **unitario** V en este estado $|\Psi\rangle$. Podemos calcularlo de la siguiente forma:

$$\langle V \rangle_\Psi = \langle \Psi | V | \Psi \rangle = \langle 0 | U^\dagger V U | 0 \rangle = \langle 0 | \tilde{\Psi} \rangle \quad (5.25)$$

donde $|\tilde{\Psi}\rangle \equiv U^\dagger V U |0\rangle$ y la acción del operador unitario $U^\dagger V U$ (que no tiene porqué ser hermítico) se realiza mediante un circuito inicializado en $|0\rangle$. Midiendo $|\tilde{\Psi}\rangle$ en la base Z , la fracción relativa de resultados $+1 \rightarrow n_0/N$ nos da acceso al **módulo del valor esperado**,

$$\sqrt{\frac{n_0(\tilde{\Psi})}{N}} = \sqrt{p_0} = |\langle 0 | \tilde{\Psi} \rangle| = |\langle \Psi | V | \Psi \rangle| = |\langle V \rangle_\Psi| \quad (5.26)$$

Podemos ver el circuito necesario para calcular este valor esperado en la Fig. 5.6.

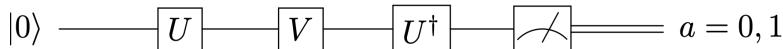


Figura 5.6: Circuito necesario para medir $\langle V \rangle_\Psi$ donde $|\Psi\rangle = U|0\rangle$ es un estado preparable.

Nota: operadores unitarios y hermíticos

Si V además de ser **unitario**, fuese **hermítico**, entonces tendríamos acceso al valor esperado completo, al tratarse de una cantidad real. Operadores de 1 cúbbit unitarios y hermíticos son, por ejemplo, los operadores $V = X, Y, Z, H$. Este argumento nos permite calcular de otra manera

$$\left. \begin{array}{l} \langle Z \rangle_\psi \\ \langle X \rangle_\psi \\ \langle Y \rangle_\psi \end{array} \right\} = \langle 0 | \tilde{\psi} \rangle = \sqrt{\frac{n_0(\tilde{\psi})}{N}} \quad \text{con} \quad \left\{ \begin{array}{l} |\tilde{\psi}\rangle = U^\dagger Z U |0\rangle \\ |\tilde{\psi}\rangle = U^\dagger H Z H U |0\rangle \\ |\tilde{\psi}\rangle = U^\dagger S H Z H S^\dagger U |0\rangle \end{array} \right. \quad (5.27)$$

Jupyter Notebook: Circuitos de un qubit y HardWare real

En [Circuitos de un qubit y HardWare real](#), sección 3.4: Valores esperados pueden verse los cálculos de valores esperados con Qiskit.

El Notebook puede descargarse de [Github](#).

Capítulo 6

Multi-qubit: definiciones y puertas.

Vamos ahora a empezar a ver que pasa cuando tenemos **más de un qúbit**.

6.1. Algunas definiciones

6.1.1. Estados multi-qubit.

Sea $\{|i\rangle\}_{i=0,1}$ la base computacional del espacio de Hilbert de un qúbit $\mathcal{H} = \mathbb{C}^2$

Definición 25 La base computacional de $\mathcal{H}^{\otimes n}$ está formada por todas las cadenas de elementos posibles

$$|i_{n-1}\rangle \otimes |i_{n-2}\rangle \otimes \dots \otimes |i_0\rangle \equiv |i_{n-1}i_{n-2}\dots i_0\rangle \quad (6.1)$$

donde $i_{n-1}, \dots, i_0 = 0, 1$ y donde \otimes representa el **producto tensorial** de espacio de Hilbert.

La **dimensión** $\dim(\mathcal{H}^{\otimes n}) = 2^n$ coincide con el número de combinaciones distintas posibles: $2 \times 2 \dots \times 2 = 2^n$.

Nota: Producto de Krönecker

Cuando tenemos que un espacio de Hilbert es un **producto tensorial** de espacios Hilbert, para construir los elementos de la base y los operadores tenemos que usar el **producto de Krönecker**.

Sean A y B dos operadores definidos respectivamente sobre \mathcal{H}_1 y \mathcal{H}_2 . El operador $C = A \otimes B$ actúa sobre $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2$ de la forma siguiente

$$C|v\rangle = A \otimes B \equiv |Av_1\rangle \otimes |Bv_2\rangle = |\omega_1\rangle \otimes |\omega_2\rangle \equiv |\omega\rangle \quad (6.2)$$

y linealmente sobre sumas de vectores de V .

Para el caso en que A y B son de dimensión 2×2 , el producto de Kronecker se puede escribir como

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B \\ a_{21}B & a_{22}B \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} & a_{11}b_{12} & a_{12}b_{11} & a_{12}b_{12} \\ a_{11}b_{21} & a_{11}b_{22} & a_{12}b_{21} & a_{12}b_{22} \\ a_{21}b_{11} & a_{21}b_{12} & a_{22}b_{11} & a_{22}b_{12} \\ a_{21}b_{21} & a_{21}b_{22} & a_{22}b_{21} & a_{22}b_{22} \end{pmatrix} \quad (6.3)$$

Para el caso de dos matrices columna (como son los vectores de la base):

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \otimes \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} a_1 b_1 \\ a_1 b_2 \\ a_2 b_1 \\ a_2 b_2 \end{bmatrix} \quad (6.4)$$

Puede verse más información en, por ejemplo, [8].

El multi-índice $i_{n-1}i_{n-2}\dots i_0$ interpretarse como un número entero escrito en base **binaria**

$$i_n i_{n-1} \dots i_1 \longleftrightarrow p = \sum_{k=1}^n 2^{k-1} i_k \quad (6.5)$$

que tomará 2^n valores $p = 0, 1, \dots, 2^n - 1$. El cambio de notación

$$\text{multi-índice} \leftrightarrow \text{entero en notación decimal} \quad (6.6)$$

será frecuente y se aplicará a cualquier elemento. Por ejemplo $|000\rangle = |0\rangle, |111\rangle = |7\rangle$ etc.

Ejemplo 2-qúbits

Por ejemplo, para un sistema de 2-qúbits, $n = 2$ y tendríamos $2^n = 2^2 = 4$ y entonces

$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad |01\rangle = |0\rangle \otimes |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad (6.7)$$

$$|10\rangle = |1\rangle \otimes |0\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad |11\rangle = |1\rangle \otimes |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad (6.8)$$

Las etiquetas de los vectores y, por tanto, de las componentes de las matrices, bi-índices $ij = 11, 12, 21, 22$ que adopta el mismo número, N^2 , de configuraciones distintas.

El vector de estado más general $|u\rangle \in \mathcal{H}^{\otimes n}$ será una combinación lineal de elementos de la base computacional $|i_n i_{n-1} \dots i_1\rangle$ en términos de unas componentes $u_{i_n i_{n-1} \dots i_1}$

$$|u\rangle = \sum_{i_0, i_1, \dots, i_{n-1}=0,1} u_{i_{n-1} i_{n-2} \dots i_0} |i_{n-1} i_{n-2} \dots i_0\rangle = \sum_{k=0}^{2^n-1} u_k |k\rangle, \quad (6.9)$$

donde hemos usado alternativamente la notación binaria y la decimal.

Ejemplo

Para $n = 2$ tendremos, en notación binaria

$$|u\rangle = \sum_{i,j=0,1} u_{ij} |ij\rangle = u_{00} |00\rangle + u_{01} |01\rangle + u_{10} |10\rangle + u_{11} |11\rangle$$

$$= u_{00} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + u_{01} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + u_{10} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + u_{11} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} u_{00} \\ u_{01} \\ u_{10} \\ u_{11} \end{bmatrix}$$

y en notación decimal, para el mismo vector

$$|u\rangle = \sum_{k=0}^{2^2-1=3} u_k |k\rangle = u_0 |0\rangle + u_1 |1\rangle + u_2 |2\rangle + u_3 |3\rangle$$

$$= u_0 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + u_1 \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + u_2 \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + u_3 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

No debería haber confusión entre ambas notaciones puesto que, en cuanto aparezca un número superior a 1 quiere decir que estamos tratando con la base decimal.

Ejercicio 22 Escribe el vector $|u\rangle = (1+i)|101\rangle - 2|010\rangle + 3|111\rangle$ normalizado

6.2. Entrelazamiento

De forma general, los estados $|u\rangle \in \mathcal{H}^{\otimes n}$ pertenecen a dos conjuntos disjuntos

- Estados **factorizables**, cuando $|u\rangle = |a\rangle \otimes |b\rangle \otimes \dots \otimes |c\rangle$
- Estados **entrelazados**, cuando $|u\rangle$ no es factorizable

Ejemplos de estados factorizables serían

$$\begin{aligned} |\Psi\rangle &= \frac{1}{\sqrt{4}} (|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \frac{1}{\sqrt{4}} (|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \\ |\Psi\rangle &= \frac{1}{\sqrt{2}} (|10\rangle + |11\rangle) = \frac{1}{\sqrt{2}} |1\rangle \otimes (|0\rangle + |1\rangle) \end{aligned}$$

Veremos un poco más sobre el entrelazamiento en la sección [6.2.2](#)

6.2.1. Base de Bell

Hasta ahora las bases que hemos usado eran todas de elementos factorizables. Podemos sin embargo usar también bases cuyo elementos sean vectores entrelazados. El ejemplo más común de base entrelazada es la **base del Bell**:

- **Base computacional (factorizable)**: $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$

■ **Base de Bell (entrelazada):**

$$\begin{aligned}
 |B_{00}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\
 |B_{01}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \\
 |B_{10}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\
 |B_{11}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)
 \end{aligned} \tag{6.10}$$

6.2.2. Medidas parciales

Una medida parcial afecta sólamente a un subconjunto de qubits de un multi-qubit. Es decir, solo afecta a uno o varios de los espacios de Hilbert, pero no a todos. Aquí encontramos una diferencia crucial entre estados factorizados y entrelazados.

6.2.2.1. Medidas parciales en un estado factorizable.

Consideremos el estado bi-qubit **factorizable**

$$|u\rangle = |a\rangle \otimes |b\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle). \tag{6.11}$$

Una medida sobre el primer qubit solo podrá resultar, con probabilidad 1/2, en uno de los dos posibles estados siguientes

$$|u\rangle \rightarrow \begin{cases} |0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |1\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \end{cases} \tag{6.12}$$

Vemos que, después de esta medición, el segundo qubit permanece intacto. Es decir, si medimos el segundo qubit seguimos teniendo una probabilidad 1/2 de medir |0⟩ o |1⟩.

6.2.2.2. Medidas parciales en un estado entrelazado.

Sin embargo, si el estado es **entrelazado**, por ejemplo,

$$|B_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \tag{6.13}$$

una medida sobre el primer qubit hace colapsar el segundo a uno de los dos siguientes posibles estados

$$|B_{00}\rangle \rightarrow \begin{cases} |0\rangle \otimes |0\rangle \\ |1\rangle \otimes |1\rangle \end{cases}. \tag{6.14}$$

también con probabilidad 1/2. Vemos que ahora, *el segundo qubit ha sufrido modificación correlacionada con el resultado obtenido de la medida del primero qubit*. Es decir, una medida parcial sobre uno de los espacios de Hilbert está afectando al otro espacio de Hilbert.

Nota: Entrelazamiento y acción fantasmal a distancia

Cuando se descubrió el entrelazamiento y las consecuencias que tenía sobre los estados al hecho de medir, a este efecto se lo denominó *acción fantasmal a distancia*, y fue muy criticado por personalidades de la talla de Albert Einstein.

Quizás con la explicación anterior no se entiende bien lo sorprendente del entrelazamiento y porque se lo denominó como acción fantasmal a distancia. Cuando hablamos de que tenemos *dos espacio de Hilbert* diferente, a lo que esto se (puede) traducir en la vida real es a que tenemos *dos partículas*. Lo que hemos visto es que sí nuestro par de partículas está en un estado

entrelazado *medir el estado de una partícula afecta el estado de la otra*. Esto es, cuanto menos, sorprendente.

Uno de los argumentos que usaban los detractores de la física cuántica para criticarla era precisamente el entrelazamiento. El argumento es simple: si tenemos un par de partículas en un estado entrelazado, nada nos impide separarlas y llevar cada una a una esquina del universo. Como están entrelazadas, si medimos una de ellas, inmediatamente el estado de la otra partícula se ve afectado y también colapsa. Es decir, parece que hay una interacción que es capaz de hacer un efecto instantáneo independientemente de la distancia que separe las partículas, lo cual viola un principio fundamental de la Relatividad Especial: *nada puede viajar más rápido que la velocidad de la luz en el vacío, ni siquiera las interacciones*.

Esta contradicción con la Relatividad se solventa teniendo en cuenta que *no existe ninguna forma de usar partículas entrelazadas para transmitir información más rápido que la velocidad de la luz*. Esto es simple de entender. Cuando tú tienes una de estas partículas entrelazadas no tienes forma de saber si la persona que tiene la otra partícula ha decidido medirla o no. En caso de que la otra persona midiera, la única forma de que tú sepas que ha medido es que te mande un mensaje y te lo diga. Como este mensaje no puede viajar más rápido que la velocidad de la luz, no se viola la causalidad.

- En ambos casos, las probabilidades de medir $|0\rangle$ ó $|1\rangle$ en el segundo cíbit, son idénticamente iguales a $1/2$.
- Eso implica que: mediciones sobre el segundo qúbit, *no permiten desvelar* si el estado original era entrelazado o no. Es decir, si tienes repetido cientos de veces el mismo par de qúbits y decides solo medir un qúbit en cada par, no tienes forma de saber qué está pasando en el otro qúbit del par, si ha colapsado al medir o no.
- Sin embargo, el entrelazamiento introduce un tipo de correlaciones muy sutiles que se pueden detectar haciendo medidas más sofisticadas, como las que conducen a las desigualdades de Bell.

Nota: Dos electrones entrelazados

Una forma intuitiva de pensar en el entrelazamiento es pensar que dos partículas están entrelazadas cuando las medidas de sus estados están correlacionadas. El ejemplo clásico es del espín de dos electrones. Dos electrones pueden entrelazarse de forma que si mides que uno de ellos tiene espín $+1/2$, el otro tiene espín $-1/2$.

La gracia de este entrelazamiento es que ambos electrones están en un estado de superposición, de forma que el estado global del sistema es

$$|\Psi\rangle = |\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle \quad (6.15)$$

Vemos que es una supersición, pero donde solo son posibles los estados en los que los electrones tienen estados al contrarios. Se ve fácilmente que este estado no es factorizable, con lo que es entrelazado.

6.3. Circuitos Multiqubit

Sea $|a_{n-1}a_{n-2}\dots a_1a_0\rangle$ un estado multiqubit de la base computacional $a_i = 0, 1$. Este estado se propaga a lo largo de un circuito de forma que *cada línea representa un espacio de Hilbert*.

La asignación que se hace en **Qiskit** coloca el qubit **menos relevante** a_0 en la línea superior. Esta ordenación en un circuito es la inversa de la que tradicionalmente se utiliza en la literatura (siguiendo

la influencia del libro de Nielsen Chuang [9]). Podemos ver esto en la Fig. 6.1

Convenio estandar	Qiskit
$ a_{n-1}\rangle$ ——	$ a_0\rangle$ ——
$ a_{n-2}\rangle$ ——	$ a_1\rangle$ ——
\vdots	\vdots
$ a_0\rangle$ ——	$ a_{n-1}\rangle$ ——

Figura 6.1: Convenios de ordenación de los qubits en la forma estándar, resaltando que Qiskit decide usar el convenio al revés.

6.4. Puertas (multi-qubit) no controladas

Vamos a ahora las puertas multi-qubit más famosas.

6.4.1. Walsh-Hadamard

El operador de Walsh-Hadamard es el producto tensorial de operadores de Hadamard

$$W = H^{\otimes n} = H \otimes H \dots \otimes H \quad (6.16)$$

La acción de $H^{\otimes n}$ sobre el estado de referencia $|00\dots 0\rangle$ produce una **superposición uniforme** de todos los estados de la base.

$$\begin{aligned} W|00\dots 0\rangle &= H|0\rangle \otimes H|0\rangle \otimes \dots \otimes H|0\rangle \\ &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \dots \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ &= \frac{1}{2^{n/2}}(|00\dots 00\rangle + |00\dots 01\rangle + \dots + |11\dots 11\rangle) \end{aligned}$$

Vemos que esta puerta es factorizable. Podemos ver en la segunda línea de la ecuación anterior que, efectivamente, al partir de un estado factorizable seguimos teniendo un estado factorizable después de aplicar la Walsh-Hadamard.

Es muy común que al principio de un circuito cuántico se aplique una Walsh-Hadamard a la mayoría de qubits para generar la superposición uniforme (recordemos que en computación cuántica ser parte teniendo todos los qubits en el estado $|0\rangle$, es decir, del estado $|0\dots 00\rangle$).

Para ver la acción de $H^{\otimes n}$ sobre un estado multiqubit es necesario recordar la Ec. (4.16) (la acción de H sobre un qubit). Tenemos pues

$$\begin{aligned} H^{\otimes n}|x\rangle &= H|x_{n-1}\rangle \otimes H|x_{n-2}\rangle \otimes \dots \otimes H|x_0\rangle \\ &= \sum_{y_{n-1}=0,1} (-1)^{y_{n-1}x_{n-1}}|y_{n-1}\rangle \otimes \sum_{y_{n-2}=0,1} (-1)^{y_{n-2}x_{n-2}}|y_{n-2}\rangle \otimes \dots \otimes \sum_{y_0=0,1} (-1)^{y_0x_0}|y_0\rangle \\ &= \sum_{y_{n-1}, y_{n-2}, \dots, y_0=0,1} (-1)^{x_{n-1}y_{n-1} + \dots + x_0y_0}|y_{n-1}\dots y_0\rangle \end{aligned}$$

es decir,

$$H^{\otimes n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} (-1)^{x \cdot y} |y\rangle \quad \text{donde } [x \cdot y = x_{n-1}y_{n-1} \oplus x_{n-2}y_{n-2} \oplus \dots \oplus x_0y_0]. \quad (6.17)$$

Para el caso particular en que lo aplicamos sobre el estado $|0\dots 00\rangle$, tenemos

$$|\Psi_0\rangle = H^{\otimes n}|0\rangle^n = \frac{1}{\sqrt{N}} \sum_{i=0}^{N=2^n} |i\rangle. \quad (6.18)$$

Si nos fijamos, estamos haciendo n operaciones y con ello estamos inicializando 2^n estados. Esta acción exponencialmente rápida es consecuencia de la superposición. No confundir superposición y entrelazamiento, son cosas diferentes. Aquí *no tenemos entrelazamiento*.

6.4.2. SWAP

La puerta SWAP es una puerta binaria fundamental (no factorizable), cuya acción consiste en permutar los estados existentes en los registros individuales sobre los que actúa. En particular, sobre los elementos de la base

$$U_{\text{SWAP}} : |00\rangle \rightarrow |00\rangle, \quad |01\rangle \rightarrow |10\rangle, \quad |10\rangle \rightarrow |01\rangle, \quad |11\rangle \rightarrow |11\rangle \quad (6.19)$$

Esto nos permite escribir el operador en la notación de producto exterior y posteriormente como matriz

$$U_{\text{SWAP}} = |00\rangle\langle 00| + |01\rangle\langle 10| + |10\rangle\langle 01| + |11\rangle\langle 11| \Rightarrow U_{\text{SWAP}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (6.20)$$

Podemos ver la representación gráfica asociada al operador SWAP en la Fíg. 6.2.

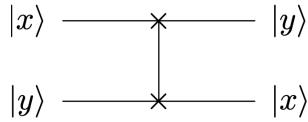


Figura 6.2: Puerta SWAP

6.5. Puertas controladas

En las puertas controladas, un operador se aplica sobre un qubit dependiendo del estado en el que se encuentra otro. Este segundo qubit se denomina **controlador**, mientras que el primero es el **controlado**. Las puertas controladas son eficientes para generar entrelazamiento. La puerta controlada se representa como sigue

$$CU = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U \quad (6.21)$$

donde U es un operador unitario de 1-qubit general. Si por el primer qubit (el controlador)

- entra $|0\rangle$, sale $|0\rangle$ y en el segundo qubit (controlado) no se hace nada (se aplica I).
- entra $|1\rangle$, sale $|1\rangle$ y en el segundo qubit (controlado) se aplica el operador U .

La **representación matricial** de CU es fácil de obtener como suma de **productos de Kronecker**

$$CU = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \otimes I + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \otimes U = \begin{bmatrix} 1 \times I & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \times U \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & U \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & U_{11} & U_{12} \\ 0 & 0 & U_{21} & U_{22} \end{bmatrix} \quad (6.22)$$

Nota: Las puertas controladas son unitarias

Escrito de esta manera es evidente que, si U es una matriz unitaria, CU también lo es $\Rightarrow (CU)^\dagger CU = I$. Esto no es algo trivial ya que la combinación lineal de operadores unitarios, en general, no es unitaria.

La acción de CU sobre elementos de la base $\{|x\rangle\}$ donde $x = 0, 1$ admite una forma compacta

$$CU : |x\rangle \otimes |y\rangle \rightarrow |x\rangle \otimes U^x |y\rangle \quad (6.23)$$

Podemos ver la representación gráfica asociada al operador CU en la Fig. 6.3

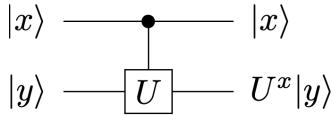


Figura 6.3: Puerta controlada

Ejercicio 23 Escribe un operador controlado y las matrices asociadas cuando

- el qúbit de control es el segundo sobre el primero.
- el operador U se aplica sobre el segundo cíbit, si el estado del primero es $|0\rangle$.

Operadores como éste existen en circuitos clásicos. La fascinante novedad es que, ahora, por el primer qúbit podría circular una superposición $a|0\rangle + b|1\rangle$. En estos casos, se efectuarían virtualmente las dos operaciones. El resultado de una acción controlada también conduce a una superposición, de forma tal que **se genera entrelazamiento**.

Para verlo, hagamos actuar CU sobre un estado de la forma $|\phi\rangle = (a|0\rangle + b|1\rangle) \otimes |v\rangle$, que es **factorizable**

$$\begin{aligned} CU \left[(a|0\rangle + b|1\rangle) \otimes |v\rangle \right] &= \left(|0\rangle \langle 0| \otimes I + |1\rangle \langle 1| \otimes U \right) \left[(a|0\rangle + b|1\rangle) \otimes |v\rangle \right] \\ &= a|0\rangle \otimes |v\rangle + b|1\rangle \otimes U|v\rangle \end{aligned}$$

Vemos que, efectivamente, el resultado es un estado entrelazado. Podemos ver esto en la Fig. 6.4.

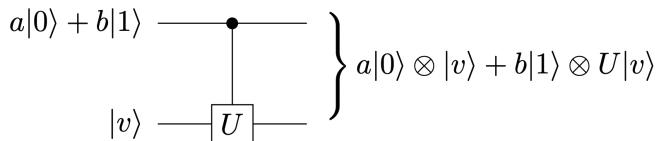


Figura 6.4: Entrelazamiento a partir de una puerta controlada.

6.5.1. CNOT

El caso más frecuente de una puerta binaria controlada es la puerta $CNOT = CX$

$$CNOT = CX = |0\rangle \langle 0| \otimes I + |1\rangle \langle 1| \otimes X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (6.24)$$

Sobre elementos de la base computacional $|xy\rangle = |x\rangle \otimes |y\rangle$ donde $x, y = 0, 1$, su acción se puede representar de manera compacta usando la suma módulo dos

$$CX : |x\rangle \otimes |y\rangle \rightarrow |x\rangle \otimes X^x |y\rangle = |x\rangle \otimes |y \oplus x\rangle \quad (6.25)$$

Podemos ver la representación gráfica asociada al operador CNOT en la Fíg. 6.5

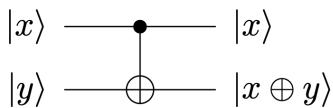


Figura 6.5: Puerta CNOT, CX o control-X

Ejercicio 24 El estado factorizable más general de dos cúbits es (a es la normalización)

$$|\psi\rangle = a(|0\rangle + b_1 e^{i\phi_1} |1\rangle) (|0\rangle + b_0 e^{i\phi_0} |1\rangle) \quad (6.26)$$

Escribe la condición más general que deben satisfacer b_0, b_1, ϕ_0 y ϕ_1 para que $CNOT|\psi\rangle$ sea un vector entrelazado. Nota: aplicar que para que un estado sea entrelazado el determinante de los coeficientes es cero.

Nota: CNOT en Qiskit

La puerta CNOT en qiskit aparece invertida con respecto a la que dibujamos en la Fig. 6.5. Ello se debe a que qiskit ordena los qubits en $|q_1 q_0\rangle$ poniendo el asociado al bit menos relevante q_0 arriba.

Cuando en un circuito en Qiskit aparece una puerta como la de ls Fig. 6.5 (con el control en el qúbit de arriba y aplicándose sobre el de abajo), como Qískit pone arriba el bit menos significativos en realidad lo que tenemos en *una CNOT donde el qúbit de control es el segundo*, es decir

$$\text{CNOT} = \text{CX} = I \otimes |0\rangle\langle 0| + X \otimes |1\rangle\langle 1| = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (6.27)$$

Jupyter Notebook: Circuitos multiqúbit

En las secciones 4.1 Circuitos de dos qúbits con una CNOT y 4.2. Entrelazamiento: Crear estados entrelazados con la CNOT y la H del notebook Circuitos multiqúbit pueden verse ejemplos de CNOTs y como crear entrelazamiento con ellas.

El Notebook puede descargarse de [Github](#).

6.5.2. Control-SWAP

Si el qúbit de control está en el estado $|1\rangle$ los dos controlados se intercambian.

$$U_{\text{CSWAP}} = |0\rangle\langle 0| \otimes I_4 + |1\rangle\langle 1| \otimes U_{\text{SWAP}}. \quad (6.28)$$

Podemos verla en la Fig. 6.6.

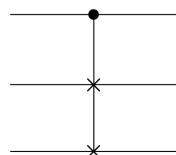


Figura 6.6: Puerta CSWAP

Jupyter Notebook: Circuitos multiqubit

En las secciones 4.3. Puerta Swap y 4.4. Puerta CSWAP y del notebook Circuitos multiqubit pueden verse ejemplos de SWAPs y CSWAPs
El Notebook puede descargarse de [Github](#).

6.6. Puertas multicontroladas

6.6.1. CCNOT o Toffoli

La puerta CCNOT, también llamada puerta de Toffoli, es un operador sobre $\mathcal{H}^{\otimes 3}$, en el que dos qubits controlan la acción de X sobre un tercero. **Sólo si ambos cùbits** de control están en el estado $|11\rangle$ el operador X actuará sobre el tercero. De nuevo, su representación es muy sencilla

$$\text{CCNOT} = (|00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 10|) \otimes I + |11\rangle\langle 11| \otimes X \quad (6.29)$$

Podemos ver la representación gráfica asociada la puerta de Toffoli en la Fíg. 6.7

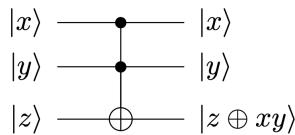


Figura 6.7: Puerta de Toffoli o CCX.

Jupyter Notebook: Circuitos multiqubit

En la sección 4.5. CCNOT o Toffoli del notebook Circuitos multiqubit pueden verse ejemplos de puertas Toffoli.

El Notebook puede descargarse de [Github](#).

Ejercicio 25 a) Obtener la matriz que representa la puerta de Toffoli en la base computacional. Reproducirla usando Qiskit. b) Obtener la matriz de un circuito de 3 cùbits con una puerta CNOT en la que el tercer cùbit controla el primero. Reproducirla usando qiskit.

6.6.2. X multi-controlada.

Cuando tenemos una puerta X con tres o más controles, la denominamos X multi-controlada (**MCX**). La puerta X se activa sí y solo sí los qubits de control están en una configuración deseada. Podemos ver un ejemplo de MCX que se activa con el estado $|1100\rangle$ en la Fig. 6.8 . El operador asociado esta puerta sería

$$MCX = |1100\rangle\langle 1100| \otimes X + (I - |1100\rangle\langle 1100|) \otimes I \quad (6.30)$$

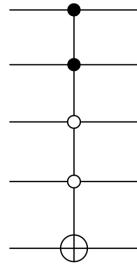


Figura 6.8: Puerta MCX (multi-controled X) para el estado de control $|1100\rangle$. Recordemos que con Qiskit sería el revés.

Nota: controles con $|0\rangle$

Los botones blancos denotan controladores que se activan si el cúbbit es $|0\rangle$. Esencialmente son iguales a un controlador negro con una puerta X antes y otra después.

Jupyter Notebook: [Circuitos multiqúbit](#)

En las sección [4.6. Puerta MCX \(multicontrolada X\)](#) del notebook [Circuitos multiqúbit](#) pueden verse ejemplos de puertas Toffoli.

El Notebook puede descargarse de [Github](#).

Capítulo 7

Medidas, Parte II: Medida de estados multi-Qubit

7.1. Medidas en la base computacional

Un aparato de medida en la base asociada al operador hermítico $\sigma_z^{\otimes n} = Z \otimes \dots \otimes Z$ hace colapsar el estado que mide a un elemento $|x\rangle$ de la **base computacional**, que identificamos mediante una cadena de bits $a_{n-1}...a_0$ con $a_i = 0, 1$, donde $x = a_{n-1}2^{n-1} + \dots + 2^0a_0$. Podemos ver esto en la Fig. 7.1



Figura 7.1: Medidor en la base Z . En este caso, midiendo destructivamente. Podría no ser el caso.

7.2. Medidas en bases generales

Vamos a suponer que queremos medir en una base ortonormal arbitraria $\{|x'\rangle\}$, $x = 0, \dots, 2^n - 1$. Buscamos un circuito que, a la llegada de un vector concreto de la base $|x'\rangle' = |a_{n-1}...a_0\rangle'$, devuelva exactamente la **misma colección** de bits $a_{n-1}...a_0$. Para ello, al igual que vimos en el caso de un qúbit, debemos conocer el operador U que relaciona esta base con la base computacional

$$|x'\rangle' = U|x\rangle \quad \Rightarrow \quad U^\dagger|x'\rangle' = |x\rangle . \quad (7.1)$$

Entonces es evidente que sólo tenemos que añadir el operador U^\dagger antes de usar el medidor estándar. Podemos ver esto en la Fig. 7.2.

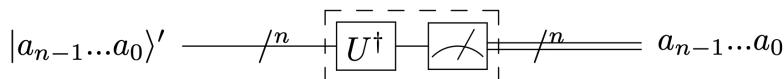


Figura 7.2: Medidor en una base arbitraria.

7.2.1. Medidas de Pauli

En caso más frecuente consiste en medir diferentes qúbits en diferentes bases de Pauli, X , Y ó Z . En este caso, $U = R_1 \otimes \dots \otimes R_n$ es un producto de rotaciones locales. Esto sigue la misma pauta que se explicó para el caso de un sólo qúbit (ver sección 5.3.1). Podemos ver un ejemplo en la Fig. 7.3.

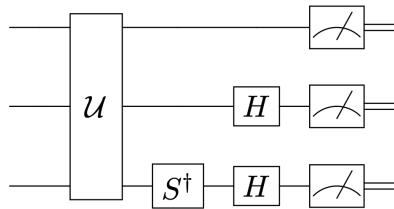


Figura 7.3: Ejemplo de un circuito que mide en la base $Z_0X_1Y_2$ (el qubit de arriba se mide en la base Z , el de en medio e la base X y el de abajo en la base Y).

Jupyter Notebook: 5 - Medidas II

Ver la sección 5.1. Medidas de Pauli del notebook 5 - Medidas II

El Notebook puede descargarse de [Github](#).

7.2.2. Medidas de Bell

Podemos medir también en bases entrelazadas, por ejemplo, en la base de Bell. En la Ec. (6.10) podemos ver la base de Bell

$$|B_{xy}\rangle = |xy\rangle' = U|xy\rangle. \quad (7.2)$$

En esta última expresión, $|xy\rangle$ está expresado en la base computacional. El circuito que genera esta base puede verse en la Fig. 7.4. Sabemos que el circuito de medida debe de llevar antes de los medidores el inverso de este circuito. Podemos ver el circuito de medida en la base de Bell en la Fig. 7.5.

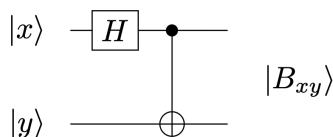


Figura 7.4: Circuito que genera la base de Bell a partir de la base computacional (ver Ec. (7.2)).

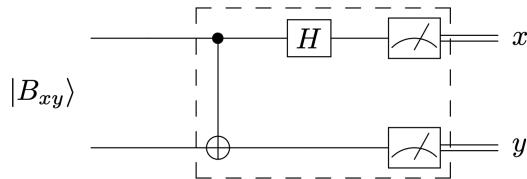


Figura 7.5: Medidor en la base de Bell.

Jupyter Notebook: 04-Medidas II sección 2

Ver la sección 2 del notebook 04-Medidas_II.

Jupyter Notebook: 5 - Medidas II

Ver la sección 5.2. Medidas de Bell del notebook 5 - Medidas II

El Notebook puede descargarse de [Github](#).

7.3. Valores esperados

Como ya vimos en el caso de un qubit, para calcular el valor esperado de un observable $A \in \text{L}(H^{\otimes n})$ debemos expandirlo en una base de **cadenas de Pauli**

$$A = \sum_{i_1, \dots, i_n=0}^3 a_{i_1 \dots i_n} \sigma_{i_1} \otimes \dots \otimes \sigma_{i_n} \quad (7.3)$$

donde $\sigma_i = (I, X, Y, Z)$. Los coeficientes se pueden obtener haciendo las trazas

$$a_{i_1 \dots i_n} = \frac{1}{2^n} \text{tr} (A \sigma_{i_1} \otimes \dots \otimes \sigma_{i_n}) \quad (7.4)$$

Por lo tanto, para hallar el valor esperado de un operador A , solo tenemos que hallar los valores esperados de las cadenas de Pauli

$$\langle A \rangle_\psi = \sum_{i_1, \dots, i_n=0}^3 a_{i_1 \dots i_n} \langle \sigma_{i_1} \otimes \dots \otimes \sigma_{i_n} \rangle \quad (7.5)$$

Jupyter Notebook: 5 - Medidas II

Ver la sección 5.3. Valor esperado de una cadena de Pauli del notebook 5 - Medidas II
El Notebook puede descargarse de [Github](#).

Ejercicio 26 Calcula el valor esperado de $\langle X \otimes Y \otimes Z \rangle_\Psi$, donde

$$|\psi\rangle = \frac{i}{4}|000\rangle + \frac{1}{\sqrt{8}}|001\rangle + \frac{1+i}{4}|010\rangle + \frac{1+2i}{\sqrt{8}}|101\rangle + \frac{1}{4}|110\rangle \quad (7.6)$$

Ejercicio 27 Considera el hamiltoniano $H = A(XX + YY + ZZ)$ siendo $A = 1.47 \cdot 10^{-6} eV$. Calcular el valor esperado de la energía $E = \langle H \rangle_\Psi$ en los cuatro estados de Bell $|\Psi\rangle = |B_{ij}\rangle$.

7.4. Medida de Hadamard

7.4.1. Valor esperado de un operador a partir de $\langle X \rangle$ y $\langle Y \rangle$

Al final, el valor esperado de un operador es un simple número que se obtiene a partir de una distribución aleatoria de valores. ¿No podríamos diseñar una variable aleatoria cuyo valor medio coincida con ese resultado? La medida de Hadamard hace precisamente esto aprovechando el entrelazamiento.

Teorema 21 Sea el circuito de la Fig. 7.6, donde por uno de los qubits (el qubit ancilla) entra el estado $|+\rangle$, por el otro entra el estado $|\psi\rangle$, se aplica el operador U sobre qubit en el estado $|\psi\rangle$ controlado por el qubit en el estado $|+\rangle$ y se mide este último qubit. Esta medida se hace bien usando un medidor en la base X (ver Fig. 7.6) o un medidor en la base Y (ver Fig. 7.6). Calculando los valores esperados de $\langle X \rangle$ e $\langle Y \rangle$ en este qubit ancilla (en la base computacional)

$$\langle X \rangle_{\text{ancilla}} = (+1) \frac{n_0^x}{n_0^x + n_1^x} + (-1) \frac{n_1^x}{n_0^x + n_1^x} \quad , \quad \langle Y \rangle_{\text{ancilla}} = (+1) \frac{n_0^y}{n_0^y + n_1^y} + (-1) \frac{n_1^y}{n_0^y + n_1^y} \quad (7.7)$$

podemos calcular el valor esperado del operador U en el estado $|\psi\rangle$ usando

$$\langle X \rangle_{\text{ancilla}} = \text{Re}\langle U \rangle_\psi , \quad \langle Y \rangle_{\text{ancilla}} = \text{Im}\langle U \rangle_\psi \quad (7.8)$$

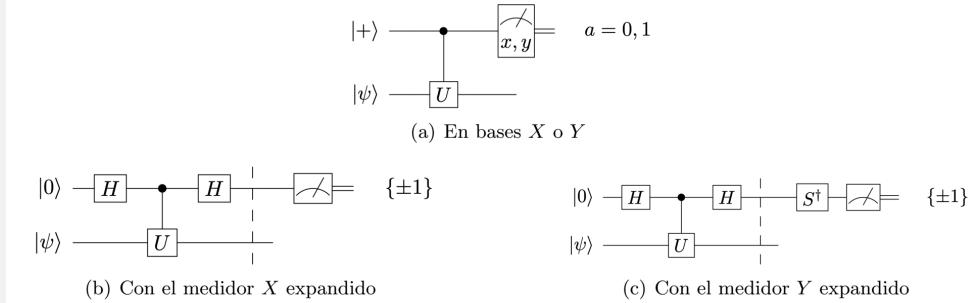


Figura 7.6: Medidas de Hadammard

Demostración: Para el caso X , el circuito anterior será el de la Fig. 7.6. Un cálculo explícito nos da el estado que llega al aparato de medida (el estado que llega a las líneas punteadas verticales)

$$|0\rangle|\psi\rangle \rightarrow |\Psi\rangle = \frac{1}{2} \left[|0\rangle \otimes (1+U)|\psi\rangle + |1\rangle \otimes (1-U)|\psi\rangle \right] \quad (7.9)$$

Si medimos el qubit ancilla (el de arriba), obtendremos como resultados $\{0, 1\}$ con probabilidades

$$\begin{aligned} p_0 &= \left| \frac{1}{2}(1+U)|\psi\rangle \right|^2 = \frac{1}{4} \langle \psi | (1+U^\dagger)(1+U) |\psi \rangle = \frac{1}{2}(1 + \text{Re}\langle \psi | U | \psi \rangle) \\ p_1 &= \left| \frac{1}{2}(1-U)|\psi\rangle \right|^2 = \frac{1}{4} \langle \psi | (1-U^\dagger)(1-U) |\psi \rangle = \frac{1}{2}(1 - \text{Re}\langle \psi | U | \psi \rangle) \end{aligned}$$

El valor esperado $\langle X \rangle_{\text{ancilla}}$ será

$$\langle X \rangle_{\text{ancilla}} = (+1) \frac{n_0^x}{n_0^x + n_1^x} + (-1) \frac{n_1^x}{n_0^x + n_1^x} = \text{Re} \langle \psi | U | \psi \rangle \quad (7.10)$$

La demostración para la medida en $|Y\rangle$ y la obtención de la $\text{Re}\langle \psi | U | \psi \rangle$ es análoga. ■

Ejercicio 28 Verificar que la parte imaginaria viene de medir $\langle Y \rangle$ en la ancilla

$$\langle Y \rangle_{\text{ancilla}} = \text{Im} \langle \psi | U | \psi \rangle . \quad (7.11)$$

Vemos que aquí estamos usando el poder del entrelazamiento, pues midiendo en un qubit que **no tiene el estado $|\psi\rangle$ ni se ha aplicado el operador U** , somos capaces de calcular $\langle U \rangle_\psi$.

Nota: operadores hermíticos

Si estamos en el caso en que U es hermítico, entonces solo hace falta medir $\langle X \rangle$, pues los operadores hermíticos tienen todos sus autovalores reales.

Ejercicio 29 Define una función `add_Hadamard_measure` que reciba un circuito y una cadena de Pauli y añada al circuito el medidor de Hadamard asociado.

Jupyter Notebook: 04-Medidas II sección 4

Ver la sección 4 del notebook **04-Medidas II**.

Jupyter Notebook: 5 - Medidas II

Ver la sección [5.4. Medida de Hadamard](#) del notebook **5 - Medidas II**

El Notebook puede descargarse de [Github](#).

7.4.2. Proyección de Hadamard

Supongamos el operador U es un operador sobre 1 qubit **a la vez hermítico y unitario**. Por tanto puede ser considerado, a la vez,

- un observable con autovalores reales $\lambda = \pm 1$ y
- una puerta cuántica con autovalores de módulo unidad

Ello deja a $\lambda = \pm 1$ como los únicos autovalores posibles para un operador así. Los operadores X, Y, Z y H son ejemplos de ello.

Denominemos $|a\rangle_U$, $a = 0, 1$ los autovectores de U con autovalores $(-1)^a$, es decir $U|a\rangle_U = (-1)^a|a\rangle_U$. En este caso, los factores $(1 \pm U)$ que aparecen en la medida de Hadamard son proyectores ortogonales sobre los autoestados de U . La imagen bajo este circuito de un estado de entrada $|0\rangle|\psi\rangle$ ahora será

$$|0\rangle|\psi\rangle = |0\rangle \otimes (\alpha|0\rangle_U + \beta|1\rangle_U) \rightarrow \alpha|0\rangle|0\rangle_U + \beta|1\rangle|1\rangle_U. \quad (7.12)$$

Podemos ver esto en la Fig. 7.7 Al igual que con los estados de Bell, cada resultado de medida en la ancilla está correlacionado con un autoestado del operador U .

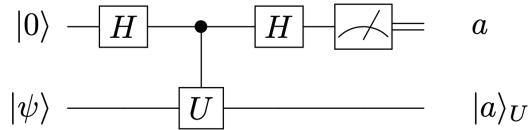


Figura 7.7: Proyección de Hadamard

Capítulo 8

Entrelazamiento en acción

8.1. Desigualdades de Bell

Con el nacimiento de la Mecánica Cuántica de mano de grandes físicos como Schrödinger o Dirac, también nacieron sus detractores. Uno de los más emblemáticos es el archiconocido Albert Einstein. Estos detractores defendían que la Mecánica Cuántica era una teoría incompleta. El argumento se basaba en que toda esta “parafernalia” de la Cuántica de las superposiciones, las indeterminaciones, el colapso de los estados,... no eran más que consecuencia de tener una teoría incompleta.

Argumentaban así que había **variables ocultas**, es decir, variables que no conocíamos pero en el caso de que las conociéramos, la Cuántica sería una teoría **determinista** (como la Mecánica Clásica). En esta batalla entre dos bandos enfrentados, entre los aférrimos defensores del determinismo (del **realismos local**) y los defensores de la cuántica, el tiempo y los experimentos dieron la razón a los segundos: la física cuántica no es una teoría determinista y no hay variables ocultas.

En este capítulo vamos a explicar el argumento que dio el golpe final a las variables ocultas y demostró el no determinismo inherente a la cuántica: **la violación de las desigualdades de Bell**.

Para ello, empecemos viendo un poco en detalle a que nos referimos cuando hablamos de **realismo local**. Las teorías con realismo local asumen que los valores que adquieren las magnitudes que se miden en un experimento **pertenecen** al sistema medido. Así, en una teoría con realismo local la posición de una partícula es algo bien definido aunque no la estemos midiendo. (En una teoría con realismo local, si un árbol cae en el bosque este hace ruido aunque nadie lo escuche.) La palabra **local** hace referencia a que ningún agente puede propagar su acción a mayor velocidad que la luz. Podría usarse la palabra **causal** en su lugar.

Con esta definición en mente, tanto la Mecánica Clásica como un teoría cuántica con variables ocultas serían teorías con realismo local. Sin embargo, la Mecánica Cuántica no lo sería. Cuando hablamos del espín de un electrón, no es correcto decir que **la proyección del espín a lo largo del eje \hat{z} es $+\hbar/2$** . Lo correcto es decir que, **al medir** la proyección sobre el eje \hat{z} , la medida obtenida es $+\hbar/2$. Dicho de otro, cualquiera de los valores $\pm\hbar/2$ se **adquiere o pone de manifiesto** de forma aleatoria al hacer una medida de la proyección elegida.

En 1964 físico nor-irlandés John Bell, trabajando en el CERN demostró [10] que **todas las teorías que respetan el realismo local** satisfacen ciertas desigualdades matemáticas. En particular, todas las magnitudes que evolucionan siguiendo las leyes de la física clásica las satisfacen.

Por el contrario, John Bell mostró cómo, en Mecánica Cuántica, el proceso de medida incorpora correlaciones sutiles que permiten **traspasar** dichas desigualdades. La discusión, pasó de ser puramente

filosófica a ser objeto de investigación experimental, culminando con el experimento de Alain Aspect y colaboradores en 1982 [11]. Se observó que la Mecánica Cuántica viola las desigualdades de Bell y, por tanto, **no es una teoría con realismo local: las propiedades no pertenecen al sistema, se generan en la interacción entre el sistema y el medidor.**

La propuesta de John Bell dio pie a una familia de desigualdades que ponen en evidencia la imposibilidad de obtener ciertas correlaciones en un mundo clásico. Vamos a examinar la desigualdad en la forma estudiada por Clauser, Horne, Shimony y Holt (**CHSH**) [12]. Posteriormente estudiaremos el **experimento de GHZ**, el cuál, también pone de manifiesto las correlaciones sutiles que introduce el entrelazamiento de una forma determinista, en lugar de estadística.

8.1.1. Perfecta anticorrelación

Central en esta discusión es la presencia de entrelazamiento. Vamos a seleccionar el denominado **singlete** de la Base de Bell.

$$|B_{11}\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \quad (8.1)$$

Una de las partículas del par entrelazado estaría en poder de Alice y la otra en poder de Bob.

Nota: creación del singlete

Desde un punto de vista experimental, para obtener este par entrelazado de electrones lo que se hace es coger los productos de una desintegración. Hay desintegraciones radioactivas que emiten estos pares entrelazados de electrones en direcciones opuestas.

Supongamos que Alice y Bob poseen sendos medidores de Stern Gerlach apuntando en la dirección $\hat{\mathbf{z}}$. De esta forma, lo que miden es el espín en la dirección z . Tenemos pues que

- Si Alice registra +1 el estado colapsa a $|01\rangle$ y, por tanto, Bob solo podrá medir -1
- Si Alice registra -1 el estado colapsa a $|10\rangle$ y, por tanto, Bob solo podrá medir +1

En definitiva hay una anticorrelación perfecta que se pone de manifiesto en el valor medio del producto de las medidas. Si las mediciones de Alice son $a_i = \pm 1$, las de Bob son $b_i = \mp 1$ respectivamente, con lo cual el valor medio

$$\langle Z \otimes Z \rangle = \frac{1}{N} \sum_{i=1}^N a_i b_i = \frac{1}{N} \sum_{i=1}^N (-1) = -1 \quad (8.2)$$

Nota: Veámoslo

Vamos a ver cómo la predicción teórica confirma este hecho. El estado $|B_{11}\rangle$ ya es autoestado del observable asociado a dicha pareja

$$Z \otimes Z |B_{11}\rangle = Z |0\rangle Z |1\rangle - Z |1\rangle Z |0\rangle = -|01\rangle + |10\rangle = -|B_{11}\rangle \quad (8.3)$$

Y el valor esperado satura en este autovalor, con lo que, la probabilidad de medida es 1

$$\langle Z \otimes Z \rangle = \langle B_{11} | Z \otimes Z | B_{11} \rangle = -\langle B_{11} | B_{11} \rangle = -1. \quad (8.4)$$

Nota

Hasta aquí, no se observa nada cuántico. La anti correlación que hemos hallado parece natural y presente en un experimento clásico hecho con una bolsa que contiene dos calcetines de dos colores: si Alice saca el blanco, el que saca Bob tiene que ser negro.

La cosa se pone más divertida cuando los dos polarizadores de Stern Gerlach **no se orientan en la misma dirección**. Es decir, cuando medimos las proyecciones del spín en dos direcciones diferentes. El observable asociado ahora a la dirección $\hat{\mathbf{n}}$ será $\hat{\mathbf{n}} \cdot \boldsymbol{\sigma}$. Aun así, los autovalores de este operador y, por ello, la proyección del espín seguirá siendo ± 1 . Como ya comentamos, da igual en que eje se mida, los valores de la proyección del espín que podemos medir son los mismos. Sin embargo, ahora el valor esperado cambia:

Teorema 22 *El valor medio del producto de las proyecciones de espín a lo largo de sendos ejes $\hat{\mathbf{m}}$ y $\hat{\mathbf{n}}$ viene dada por el coseno del ángulo θ que forman los ejes de los dos detectores*

$$\langle B_{11} | (\hat{\mathbf{m}} \cdot \boldsymbol{\sigma} \otimes \hat{\mathbf{n}} \cdot \boldsymbol{\sigma}) | B_{11} \rangle = -\cos \theta = -\hat{\mathbf{m}} \cdot \hat{\mathbf{n}} \quad (8.5)$$

Ejercicio 30 Prueba el resultado del teorema 22

Cuando los ejes son paralelos recuperamos la anticorrelación, independientemente de la dirección

$$-\hat{\mathbf{n}} \cdot \hat{\mathbf{n}} = -\cos 0 = -1 \quad (8.6)$$

mucho más interesante es cuando las direcciones de los detectores de Alice y Bob no coinciden ($\hat{\mathbf{n}} \neq \hat{\mathbf{m}}$).

8.1.2. Desigualdad CSCH

En 1970 Clauser, Horne, Shimony y Holt [12] propusieron una figura de mérito fácilmente accesible para verificar las desigualdades de Bell. La idea es que Alice y Bob pueden orientar sus detectores en **dos direcciones arbitrarias** cada uno. Para Alice Alice denotamos $\hat{\mathbf{n}}_A, \hat{\mathbf{n}}'_A$ y para Bob $\hat{\mathbf{n}}_B, \hat{\mathbf{n}}'_B$. Los pasos a seguir son los siguientes:

1. Alice y Bob seleccionan cada uno una orientación (de las dos posibles de cada uno), p. ej. $\hat{\mathbf{n}}_A$ y $\hat{\mathbf{n}}'_B$, para sus detectores. Hay cuatro parejas posibles dependiendo de que selecciones cada uno

Alice	Bob
$\hat{\mathbf{n}}_A$	$\hat{\mathbf{n}}_B$
$\hat{\mathbf{n}}_A$	$\hat{\mathbf{n}}'_B$
$\hat{\mathbf{n}}'_A$	$\hat{\mathbf{n}}_B$
$\hat{\mathbf{n}}'_A$	$\hat{\mathbf{n}}'_B$

(8.7)

2. Alice y Bob reciben un electrón cada uno de un par entrelazado en el estado $|B_{11}\rangle$
3. Alice y Bob realizan la medida de la proyección del espín a lo largo del eje elegido y anotan el resultado de la medición $(a, b) = (\pm 1, \pm 1)$
4. Repiten el paso anterior un número $i = 1, \dots, N$ grande de veces, y con los datos obtenidos $(a_i, b_i) = (\pm 1, \pm 1)$ donde $i = 1, \dots, N$ pueden reconstruir la cantidad

$$C(\hat{\mathbf{n}}_A, \hat{\mathbf{n}}'_B) = \frac{1}{N} \sum_{i=1}^N a_i b'_i \in [-1, 1] \quad (8.8)$$

Esta vez, como miden en direcciones diferentes, los dos pueden medir el mismo valor, así que esta cantidad estará entre -1 y $+1$.

5. Repiten todo el proceso anterior para las cuatro posibles orientaciones elegidas de forma aleatoria. Con las $4N$ mediciones construyen la cantidad

$$R = |C(\hat{\mathbf{n}}_A, \hat{\mathbf{n}}_B) + C(\hat{\mathbf{n}}_A, \hat{\mathbf{n}}'_B) + C(\hat{\mathbf{n}}'_A, \hat{\mathbf{n}}_B) - C(\hat{\mathbf{n}}'_A, \hat{\mathbf{n}}'_B)| \quad (8.9)$$

En un mundo clásico, supondríamos que los valores a_i, b_i proceden de **valores predefinidos** para cada sistema individual, sobre el que efectuamos simplemente un promedio estadístico de muchos sistemas. Entonces podemos probar la **desigualdad de CSCH**:

Teorema 23 *La desigualdad de CSCH afirma que*

$$R \leq 2 \quad (8.10)$$

Demostración: Es fácil ver que se cumple para cada colección $a_i, a'_i, b_i, b'_i \in \pm 1$ la desigualdad

$$a_i(b_i + b'_i) + a'_i(b_i - b'_i) = \pm 2 \quad (8.11)$$

porque si $b_i + b'_i = \pm 2$ entonces $b_i - b'_i = 0$ y viceversa. Ahora podemos demostrar la desigualdad

$$\begin{aligned} R &= \lim_{N \rightarrow \infty} \left| \frac{1}{N} \sum_{i=1}^N (a_i b_i + a_i b'_i + a'_i b_i - a'_i b'_i) \right| \\ &= \lim_{N \rightarrow \infty} \frac{1}{N} \left| \sum_{i=1}^N (a_i(b_i + b'_i) + a'_i(b_i - b'_i)) \right| \\ &\leq \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N |(a_i(b_i + b'_i) + a'_i(b_i - b'_i))| \\ &= \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N 2 \\ &= 2 \end{aligned}$$

■

La Mecánica Cuántica nos proporciona una respuesta teórica para R que sólo depende de los ángulos relativos $\cos \theta_{AB} = \cos(\theta_A - \theta_B) = \hat{\mathbf{n}}_A \cdot \hat{\mathbf{n}}_B$.

$$R = |\cos \theta_{AB} + \cos \theta_{A'B} + \cos \theta_{AB'} - \cos \theta_{A'B'}|. \quad (8.12)$$

8.1.2.1. Ejemplo particular.

Ahora sólo hace falta jugar un poco con los detectores. Por ejemplo, podemos situarlos en el plano (y, z) , perpendicular al eje de propagación x , de manera que los vectores $\hat{\mathbf{n}}'_A, \hat{\mathbf{n}}_A, \hat{\mathbf{n}}_B$ y $\hat{\mathbf{n}}'_B$ estén ordenados correlativamente en sentido horario. Finalmente tomaremos dos ejes coincidentes $\hat{\mathbf{n}}_A = \hat{\mathbf{n}}_B$ paralelos $\Rightarrow \theta_{AB} = 0$, y apertura igual para ambos, $\theta_{A'A} = \theta_{BB'} = \varphi$, de modo que $\theta_{A'B'} = 2\varphi$. Podemos ver esta disposición en la Fig. 8.1.

La expresión de R nos queda

$$R = |1 + 2 \cos \varphi - \cos 2\varphi|. \quad (8.13)$$

Derivando vemos que esta expresión alcanza su máximo cuando $\sin \varphi = \sin 2\varphi$ lo cual tiene solución $\varphi = \pi/3 = 60^\circ$. Sustituyendo encontramos $R = 2.5 > 2$, violando la desigualdad CHSH de la Ec. (8.10).

Jupyter Notebook: 6. Entrelazamiento

Ver la sección 6.1. La desigualdad CSCH del notebook 6. Entrelazamiento.

El Notebook puede descargarse de [Github](#).

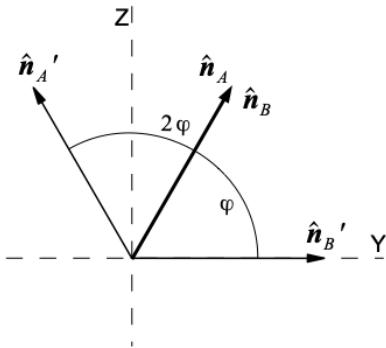


Figura 8.1: Ejemplo particular de unas orientaciones de los medidores de Alice y Bob

Ejercicio 31 Prueba con otros estados de la base de Bell. Realiza este experimento en un ordenador real.

Ejercicio 32 Usar bases perpendiculares de Alice A y A' y Bob B y B' , y formando un ángulo φ entre sí. Sin pérdida de generalidad puedes tomar $B = X$ y $B' = Z$. Variar φ en el intervalo $(0, \pi)$ y hallar el valor de la máxima violación de la desigualdad de Bell.

8.2. Experimento GHZ

Las desigualdades de Bell-CHCS demuestran que hay una manera de distinguir una teoría con realismo local de una en la que este postulado no exista. Lo hacen de una forma estadística: es necesario formar ciertas medias de datos sobre estados de 2-cúbits, los estados de Bell. Vamos a ver ahora una forma alternativa de llegar a la misma conclusión, pero esta vez de una forma **determinista**, no probabilística.

Supongamos un sistema, compuesto de tres subsistemas, A, B y C . En cada uno de ellos hay dos **magnitudes** observables, X e Y , que al ser medidas adquieren valores binarios $x, y = \pm 1$. Pretendemos saber si existe un estado del sistema tal que, con el resultado de medidas simultáneas X o Y de cada una de sus partes A, B y C , se obtengan resultados x e y que verifiquen las siguientes ecuaciones:

medimos		obtenemos x, y tales que
XYY	\rightarrow	$xyy = 1$
YXY	\rightarrow	$yxy = 1$
YYX	\rightarrow	$yyx = 1$
XXX	\rightarrow	$xxx = -1$

Si la teoría satisface los axiomas de realismo local, los valores de X es Y estarán bien definidos independientemente de la medida que efectuemos. Dicho de otra forma, las medidas que efectuamos son compatibles y no afectan a los resultados posibles. Entonces podemos utilizar conclusiones que extraigamos de los resultados de un experimento para los de otro (notar que se supone que el estado es el mismo en todos los casos)

- Las tres primeras afirman que $x = 1$ puesto que $y^2 = 1$.
- La última afirma que $x = -1$

vemos que hay una contradicción. Concluimos que las medidas efectuadas **no son compatibles entre sí**. Como vemos, la paradoja existe en tanto en cuanto atribuyamos a las magnitudes X e Y una noción de realidad independiente de la medición. Es decir, cuando pensamos desde la perspectiva del realismo local en la que los valores de x e y están predefinidos y el hecho de medir no los altera.

Concluimos que para que se cumplan las 4 afirmaciones tenemos que aceptar que las variables X e Y , no pueden tener valores predefinidos simultáneamente en los cuatro experimentos. La pregunta ahora es, hay algún estado cuántico que si cumpla esas 4 condiciones? La respuesta es sí, el estado GHZ.

En 1997 Greenberger, Horne y Zeilinger estudiaron las propiedades de una serie de estados de 3-qubits. Consideremos el siguiente estado entrelazado

$$|\text{GHZ}\rangle = \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle). \quad (8.14)$$

Este estado se denomina **estado GHZ**. Cuánticamente es fácil ver que el estado GHZ proporciona una solución al conjunto de ecuaciones. Supongamos que x e y son los resultados de aplicar $X = \sigma_x$ y $Y = \sigma_y$, los operadores hermíticos usuales que miden la componente del espín

$$X|0\rangle = |1\rangle, \quad X|1\rangle = |0\rangle, \quad Y|0\rangle = i|1\rangle, \quad Y|1\rangle = -i|0\rangle. \quad (8.15)$$

Por un lado,

$$xyy|\text{GHZ}\rangle = X \otimes Y \otimes Y \left(\frac{1}{\sqrt{2}}(|000\rangle - |111\rangle) \right) = \frac{1}{\sqrt{2}}(i^2|111\rangle - (-i)^2|000\rangle) = +|\text{GHZ}\rangle, \quad (8.16)$$

y, análogamente, obtenemos $xyy = yxy = yyx = +1$. Por otro, $xxx = -1$ se sigue de

$$xxx|\text{GHZ}\rangle = X \otimes X \otimes X \left(\frac{1}{\sqrt{2}}(|000\rangle - |111\rangle) \right) = \frac{1}{\sqrt{2}}(|111\rangle - |000\rangle) = -|\text{GHZ}\rangle. \quad (8.17)$$

Podemos ver en la Fig. 8.2 el circuito que genera el estado GHZ de la Ec. (8.14).

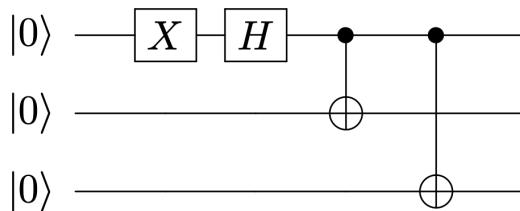


Figura 8.2: Circuito que genera el estado GHZ de la Ec. (8.14)

Jupyter Notebook: 6. Entrelazamiento

Ver la sección 6.2. Experimento GHZ del notebook 6. Entrelazamiento.
El Notebook puede descargarse de [Github](#).

8.3. Teleportación

El entrelazamiento conlleva un tipo nuevo de correlación que acaba constituyendo un recurso importante. Podemos usar esta correlación para **teleportar** estados. Lo primero que debemos entender es que cuando hablamos de “teleportar” estamos hablando de **teleportar un estado cuántica**. Es decir, no estamos teleportando un particula, sino que estamos aprovechando el entrelazamiento para transferir el estado de una partícula a otra. Como las partículas de la misma clase (como los electrones)

son indistinguibles, si conseguimos teleportar el estado de una partícula a otra, a efectos prácticos es como si teleportaramos la partícula en sí.

Supongamos que Alice y Bob tienen dos qubits que se encuentran en el estado entrelazado $|B_{00}\rangle$. Alice tiene, un segundo qubit inicializado en un estado arbitrario $|\phi\rangle$, y se plantea la posibilidad de transferirlo o clonarlo en el laboratorio de Bob. El circuito de la Fig. 8.3 permite efectuar esa tarea

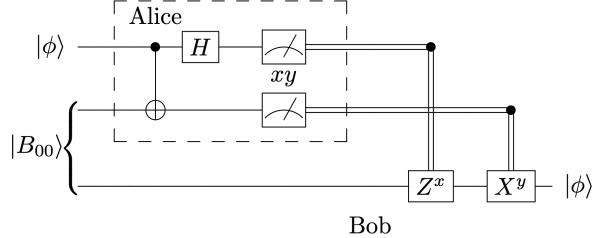


Figura 8.3: Circuito del protocolo de teleportación.

1. El estado inicial es

$$|\phi\rangle|B_{00}\rangle = |\phi\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle). \quad (8.18)$$

Alice tiene acceso a los dos primeros qubits y Bob al tercero. El estado que queremos teleportar es $|\phi\rangle$. De forma genérica, este estado será de la forma

$$|\phi\rangle = a|0\rangle + b|1\rangle \quad (8.19)$$

2. Alice realiza una *medida de Bell* a sus dos qubits. Esto implica un desentrelazador $U_{\text{desent}} = (H \otimes I) \cdot U_{\text{CNOT}}$. Un cálculo sencillo da el resultado

$$\begin{aligned} (H \otimes I \otimes I)(U_{\text{CNOT}} \otimes I)|\phi\rangle|B_{00}\rangle &= \frac{1}{2} \left[|00\rangle(a|0\rangle + b|1\rangle) + |01\rangle(a|1\rangle + b|0\rangle) \right. \\ &\quad \left. + |10\rangle(a|0\rangle - b|1\rangle) + |11\rangle(a|1\rangle - b|0\rangle) \right] \end{aligned}$$

Podemos ver que al estar entrelazados, las operaciones sobre los qubits de Alice afectan al estado del qubit de Bob.

3. Alice mide el estado que obra en su poder, y obtiene un 2-bit clásico, xy de manera equiprobable para las 4 posibilidades. De forma correlacionada, el qubit de Bob colapsa a uno de los 4 estados $|\varphi_{xy}\rangle$, pero no sabe a cuál.
4. Alice envía el resultado de su medida xy por un canal clásico a Bob.
5. Bob efectúa sobre su qubit, una operación controlada por este 2-bit, $U_{xy} = X^y Z^x$.

$$xy = \begin{cases} 00 \\ 01 \\ 10 \\ 11 \end{cases} \implies X^y Z^x |\varphi_{xy}\rangle = \begin{cases} I : (a|0\rangle + b|1\rangle) \\ X : (a|1\rangle + b|0\rangle) \\ Z : (a|0\rangle - b|1\rangle) \\ XZ = -iY : (a|1\rangle - b|0\rangle) \end{cases} \rightarrow a|0\rangle + b|1\rangle = |\phi\rangle \quad (8.20)$$

Como resultado de esta operación, el qubit de Bob es finalmente $|\phi\rangle$.

Ejercicio 33 Calcula $(H \otimes I \otimes I)(U_{\text{CNOT}} \otimes I)|\phi\rangle|B_{00}\rangle$ y verifica que la ecuación del paso 2 es correcta.

Jupyter Notebook: 6. Entrelazamiento

Ver la sección [6.3. Protocolo de teleportación](#) del notebook [6. Entrelazamiento](#).
El Notebook puede descargarse de [Github](#).

El siguiente ejercicio ilustra el principio de la medida diferida.

Ejercicio 34 Modifica y ejecuta el circuito de teleportación de dos formas distintas

- a) sustituyendo los controles clásicos por controles cuánticos
- b) permutando el orden de los controles y los aparatos de medida

Discute la sutileza que distingue estas posibilidades.

Ejercicio 35 Cambia el estado que comparten Alice y Bob por $|B_{11}\rangle$ y modifica el circuito para que teleporte igualmente.

Nota: causalidad y clonación.

El protocolo de teleportación parece poner en riesgo conceptos fundamentales. Sin embargo, no es así:

- **No clonación:**

El protocolo tiene como ingrediente esencial la medida y, por tanto, la destrucción del estado de Alice. Como consecuencia, el estado ha sido teleportado pero no clonado. De no ser así, entraríamos en conflicto con el **Teorema de No Clonación**

- **Causalidad:**

El protocolo de teleportación **no viola causalidad**. Es necesario mandar una información clásica (como muy rápido a la velocidad de la luz) para resolver la ambigüedad que le queda a Bob. Esta parte es la que hace que la teleportación no sea un proceso instantáneo de acción a distancia.

8.4. Intercambio de Entrelazamiento

Ya hemos visto cómo el ejemplo más sencillo de entrelazamiento tiene una aplicación muy interesante en la teleportación. Vamos a ver que el entrelazamiento se puede “contagiar” a terceras partes. En inglés se denomina **entanglement swapping**.

Consideremos el circuito de la Fig. 8.4. El circuito podemos interpretarlo de la siguiente forma:

1. Alice (A) entrelaza un qubit con Charles (C) y otro con Bob (B).
2. Después, Alice hace una medida de Bell, y comunica el resultado xy a Charles y a Bob respectivamente.
3. Charles y Bob efectúan las puertas controladas Z^x y X^y respectivamente. El resultado final es que los qubits de Bob y Charlie están entrelazados.

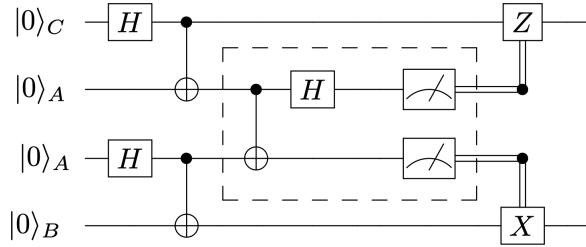


Figura 8.4: Circuito para el intercambio de entrelazamiento (entanglement swapping)

Ejercicio 36 Completa los siguientes apartados:

- Programa el circuito de la Fig. 8.4.
- Ejecuta varias veces el circuito y muestra que el estado final que comparten Bob y Charles está entrelazado. ¿Es siempre el mismo estado?

Ejercicio 37 A partir del circuito de la Fig. 8.4, diseña y ejecuta un protocolo capaz de teleportar un qubit arbitrario entre Charles y Bob.

8.5. Teorema de no-clonación

En un principio, que el protocolo de teleportación exija la destrucción del estado inicial para teleportarlo puede parecer una particularidad de este protocolo. Sin embargo, nada más lejos de la realidad. El **Teorema de No Clonación** es uno de los resultados más sencillos y a la vez más importante del formalismo de la Mecánica Cuántica. De hecho su formalización completa es bastante reciente, 1982, debida a Wootters, Zurek [13] y Dieks [14].

Teorema 24 (de No Clonación) *No existe un operador unitario U (clonador) que, para un estado arbitrario $|\psi\rangle$, realice la siguiente operación*

$$U|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle \quad (8.21)$$

Demostración: Supondremos que U existe y llegaremos a una contradicción. Tratemos de clonar el estado $\alpha|\psi\rangle + \beta|\phi\rangle$. Esto implica evaluar

$$U(\alpha|\psi\rangle + \beta|\phi\rangle) \otimes |0\rangle = (\alpha|\psi\rangle + \beta|\phi\rangle) \otimes (\alpha|\psi\rangle + \beta|\phi\rangle)$$

Sin embargo, la linealidad de U nos permite seguir otro camino

$$\begin{aligned} U(\alpha|\psi\rangle + \beta|\phi\rangle) \otimes |0\rangle &= \alpha U|\psi\rangle \otimes |0\rangle + \beta U|\phi\rangle \otimes |0\rangle \\ &= \alpha|\psi\rangle \otimes |\psi\rangle + \beta|\phi\rangle \otimes |\phi\rangle. \end{aligned}$$

Los dos resultados son diferentes y el teorema queda demostrado por reducción al absurdo. ■

Nota

- El teorema de no clonación pone de manifiesto la *tensión* que hay entre linealidad y tensorialidad cuando se trata de aplicar operadores.
- Es muy importante recalcar que la validez de este teorema sólo aplica a estados *genéricos*. Si por ejemplo nos restringimos a estados de la base $|0\rangle$ y $|1\rangle$, entonces la mera puerta

CNOT es un operador de clonación.

$$CX |00\rangle \rightarrow |00\rangle \quad , \quad CX |10\rangle \rightarrow |11\rangle \quad (8.22)$$

Capítulo 9

Hardware: Técnicas de control y computación en RMN.

En este capítulo vamos a ver un poco más en detalle como se implementa un computador cuántico. En concreto, veremos de manera relativamente profunda el formalismo detrás del control de los qubits en **RMN** (Resonancia Magnética Nuclear). Esto es debido a que, a pesar de que actualmente las implementaciones de los qubits son variopintas, la teoría detrás del control de los qubits es más o menos igual.

9.1. Introducción.

El problema del control de sistemas cuánticos acoplados múltiples es un tema emblemático de la RMN, y puede resumirse como sigue: dado un sistema con Hamiltoniano

$$\mathcal{H} = \mathcal{H}_{sys} + \mathcal{H}_{control} \quad (9.1)$$

donde \mathcal{H}_{sys} es el Hamiltoniano del sistema en ausencia de cualquier control, y $\mathcal{H}_{control}$ describe términos que están bajo control externo, ¿cómo puede aplicarse una transformación unitaria U deseada, en presencia de imperfecciones y utilizando un mínimo de recursos? De forma similar a otros escenarios en los que el control cuántico es una idea bien desarrollada, como en la excitación láser de reacciones químicas, $\mathcal{H}_{control}$ surge de secuencias sincronizadas con precisión de múltiples pulsos de radiación electromagnética, aplicados de forma coherente en fase, con diferentes anchuras, frecuencias, fases y amplitudes de pulso.

En RMN lo que se controla con estos pulsos es el **espín nuclear**. Tenemos que ver entonces que es el espín nuclear (sección 9.2), como se puede usar para formar qubits (construir \mathcal{H}_{sys} , sección 9.3.1) y como se manipulan usando pulsos electromagnéticos (construir $\mathcal{H}_{control}$, sección 9.3.2). Finalmente, veremos más en detalle como se implementan estos pulsos (sección 9.4).

Para más información en general sobre física nuclear, puede consultarse un libro clásico como es [15]. Para más información sobre como controlar qubits de RMN puede consultarse [16]. Gran parte de este capítulo se basa en intentar explicar de una forma más simple este último artículo.

9.2. El espín nuclear

En el núcleo, cada nucleón (protones o neutrones) posee un momento angular orbital y un momento angular intrínseco o espín. Al igual que el electrón, los nucleones son fermiones de espín $\hbar/2$.

A cada **estado nuclear** se asigna un único número cuántico de espín I , representando el **momento angular total** (orbital más intrínseco) de todos los nucleones en el núcleo. El vector \vec{I} puede considerarse como la suma de las contribuciones orbital y intrínseca (espín intrínseco de protones y neutrones) de los momento angulares de los nucleones

$$\begin{aligned}\vec{I} &= \sum_{i=1}^A (\vec{l}_i + \vec{s}_i) \\ &= \vec{L} + \vec{S} \\ &= \sum_{i=1}^A \vec{j}_i\end{aligned}$$

donde A es el **número másico**

$$A = Z + N \quad (9.2)$$

donde Z es el **número de protones** y N el **número de neutrones**.

El *número cuántico* I tiene la conexión usual con el *vector* \vec{I} :

$$\begin{aligned}|\vec{I}| &= \sqrt{I(I+1)}\hbar \\ I_i &= m_i\hbar \quad (m_i = I, I-1, \dots, -I+1, -I)\end{aligned} \quad (9.3)$$

Al vector \vec{I} se lo denomina **espín nuclear**.

Nota: S_z e I_Z

Aunque aquí hayamos decidido usar una notación diferenciadora para el spín de una partícula y un núcleo (\vec{S} y \vec{I}), a partir de aquí **usaremos \vec{S} para todo**.

Nota: Operador de espín para partículas de espín 1/2

En física cuántica todos los observables son operadores (matrices hermíticas). Ya hemos visto el vector de espín, \vec{S} (o \vec{I} , pero ya comentamos que abandonamos esta notación), ahora nos falta ver el **operador espín**, \hat{S} o simplemente S . Para partículas de espín 1/2, este toma la forma

$$S = \frac{\hbar}{2} \vec{\sigma} \quad (9.4)$$

donde $\vec{\sigma}$ es el vector de matrices de Pauli ($\sigma_x, \sigma_y, \sigma_z$). También podemos escribirlo pues como

$$S = (S_x, S_y, S_z) \quad (9.5)$$

donde

$$\boxed{S_x = \frac{\hbar}{2} \sigma_x}, \quad \boxed{S_y = \frac{\hbar}{2} \sigma_y}, \quad \boxed{S_z = \frac{\hbar}{2} \sigma_z}. \quad (9.6)$$

Nota: Operador momento angular

Es común en física que se usen la notación $I = (I_x, I_y, I_z)$ para hablar del **operador de momento angular**. No confundirlo con el espín nuclear que vimos antes. Este operador es simplemente el operador de espín (9.4) pero sin el factor \hbar , es decir,

$$I = \frac{1}{2} \vec{\sigma} \quad (9.7)$$

Por ejemplo, en el paper de referencia de esta sección [16] se usa esta notación.

La Ec. (9.2) representa lo que en principio podría ser un acoplamiento muy complicado de muchos vectores para dar un solo resultado, y puede que no sea evidente por qué podemos despreciar esta estructura interna y tratar el núcleo como si fuera un momento angular una “partícula”. Esto es posible porque las interacciones a las que sometemos el núcleo, como los campos electromagnéticos, no son suficientemente fuertes como para cambiar la estructura interna o romper los acoplamientos de los nucleones que son responsables de la Ec. (9.2).

El valor del espín nuclear depende del valor del número másico:

Núcleos con A impar: \rightarrow Espín nuclear, I , semientero

Núcleos con A par: \rightarrow Espín nuclear, I , entero

Esto es debido a que los nucleones tienden a acoplarse en parejas de iguales (pp y nn) con el mismo momento angular orbital pero con los espines opuestos, situación consistente con el principio de exclusión de Pauli y que minimiza la energía potencial del sistema permitiendo un mayor solapamiento de las funciones de onda de los nucleones. Teniendo esto en cuenta, es de esperar que usualmente las propiedades magnéticas nucleares estén determinadas por el último nucleón desapareado (si existe), por el acoplamiento de la última pareja de nucleones, ó por el acoplamiento del espín del nucleón desapareado con el espín del core nuclear residual (bastante similar a lo que ocurre con los electrones más externos de los átomos).

Todos los núcleos que se conocen (estables e inestables) con Z par y N par tienen espín cero en el estado fundamental. Lo cual es una evidencia de que la interacción fuerte manifiesta una especie de fuerza de apareamiento tal que existe una tendencia a mantener los nucleones apareados. Como consecuencia, resulta que el espín del estado fundamental de un núcleo con A impar debe ser igual al $\vec{j}_i = \vec{s}_i + \vec{l}_i$ del protón o neutrón desapareado.

9.3. Qúbits de RMN

En esta sección vamos a describir como es el sistema con el que se construyen los qúbits en RNM, basándonos en su **Hamiltoniano del sistema** y su **Hamiltoniano de control**. El Hamiltoniano del sistema da la energía de los espines simples y acoplados en un campo magnético estático, y el Hamiltoniano de control surge de la aplicación de pulsos de radiofrecuencia al sistema en, o cerca de, sus frecuencias resonantes. Veremos que para describir el efecto de los pulsos es más conveniente usar un **sistema de referencia giratorio**.

9.3.1. Hamiltoniano del sistema

9.3.1.1. Espines simples.

Una partícula con espín constituye un **dipolo magnético**. En esencia, dipolo magnético es un pequeño imán, es decir, una fuente de campo magnético. El **momento dipolar magnético**, $\vec{\mu}$, es proporcional al momento angular de espín, \vec{S} :

$$\vec{\mu} = \gamma \vec{S}, \quad (9.8)$$

donde la constante de proporcionalidad, γ , se denomina ratio giromagnético (**gyromagnetic ratio**). Este toma la forma

$$\gamma = g \frac{q}{2m}, \quad (9.9)$$

donde q es la carga eléctrica de la partícula, m su masa y g se denomina **factor g (g-factor)**.

Cuando un dipolo magnético se sitúa en un campo magnético \vec{B} , este dipolo experimenta un torque, $\vec{\mu} \times \vec{B}$, que tiende a alinearlo paralelo al campo (como la aguja de una brújula). La energía asociada con este torque es

$$E = -\vec{\mu} \cdot \vec{B} \quad (9.10)$$

con lo que el Hamiltoniano para una partícula cargada con espín, en reposo en un campo magnético \vec{B} , es

$$\mathcal{H} = -\gamma \vec{B} \cdot \vec{S} \quad (9.11)$$

donde S es el operador de espín (ver Ec. (9.4) para el caso de partículas de espín 1/2).

Si tenemos una partícula (o un núcleo) de espín 1/2 (no consideraremos espines de mayor orden en estas notas) en un campo magnético \vec{B}_0 a lo largo del eje \hat{z} , entonces su evolución temporal está gobernada por el Hamiltoniano

$$\boxed{\mathcal{H} = -\gamma B_0 S_z = -\omega_0 S_z = \begin{bmatrix} -\hbar\omega_0/2 & 0 \\ 0 & \hbar\omega_0/2 \end{bmatrix}}, \quad (9.12)$$

donde

- γ es ratio giromagnético del núcleo (ver Ec. (9.9)),
- $\omega/2\pi = \gamma B_0$ es la **frecuencia de Larmor**
- S_z es el operador de espín en la dirección \hat{z} (ver Ec. (9.6)).

Nota: frecuencia y frecuencia angular

En Física, la **frecuencia** se define como el inverso del periodo de rotación: $f = 1/T$ y se mide en Hz (1/segundos). Además, se define la **frecuencia angular** como $\omega = 2\pi f$ y se mide en radianes/segundo. Muchas veces se denomina a las dos, simplemente, frecuencia.

La interpretación de la Ec. (9.12) es simple. Cuando tenemos una partícula de espín 1/2 aislada, las dos proyecciones del espín son **degeneradas**, es decir, tiene la misma energía. Esto es fácil de entender si pensamos en que no hay ninguna dirección privilegiada en el sistema. Cuando introducimos un campo magnético, ahora sí hay una dirección privilegiada: aquella con el momento dipolar magnético $\vec{\mu}$ apuntando en la misma dirección que el campo magnético. Esto es debido a que el espín es como un pequeño imán y tiende a orientarse con el campo magnético. En este momento, el sistema pasa a ser **no-degenerado**, y un estado tiene más energía que el otro.

Esto último se traduce matemáticamente en la Ec. (9.12). Como el Hamiltoniano es diagonal, los elementos de la diagonal son las energías de los estados. Vemos que el estado con el espín apuntando en la misma dirección (es estado $|0\rangle$ o $|\uparrow\rangle$) que el campo magnético tiene menos energía que el estado que apunta en la dirección contraria (el estado $|1\rangle$ o $|\downarrow\rangle$). La diferencia de energía entre los estados es de $\hbar\omega_0$, como se puede ver en la Fig. 9.1. Esta separación de energías (*energy splitting*) se conoce como **Zeeman splitting**.

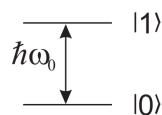


Figura 9.1: Diagrama de energías de una partícula de espín 1/2 en un campo magnético.

Podemos entender gráficamente la evolución temporal $U = e^{-i\mathcal{H}t/\hbar}$ bajo el Hamiltoniano de la Ec. (9.12) como un movimiento de precesión en la esfera de Bloch alrededor de \vec{B}_0 , como se muestra en

la Fig. 9.2. Como es habitual, definimos el eje \hat{z} de la esfera de Bloch como el eje de cuantización del Hamiltoniano, con $|0\rangle$ a lo largo de $+\hat{z}$ y $|1\rangle$ a lo largo de $-\hat{z}$.

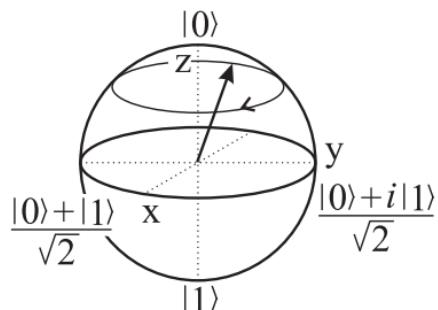


Figura 9.2: Precesión del valor esperado del espín entorno al campo magnético.

Nota: Esfera de Bloch para el espín

Véase que para el espín la representación en la esfera de Bloch no es más que la representación en 3D del vector de espín.

Nota: orientación del espín

En las Ecs. (9.8) y (9.9) vemos que el vector de espín y el momento dipolar magnético se relacionan por una constante que γ que puede ser positiva o negativa, dependiendo del valor de la carga eléctrica de la partícula. Es decir, estos vectores pueden apuntar en la misma dirección o en dirección opuesta.

Hemos comentado que cuando tenemos un campo magnético, el estado de menor energía es aquel con el **momento dipolar magnético** apuntando en la dirección del campo. Como podemos ver en la Fig. 9.3, tenemos dos opciones dependiendo de la carga de la partícula.

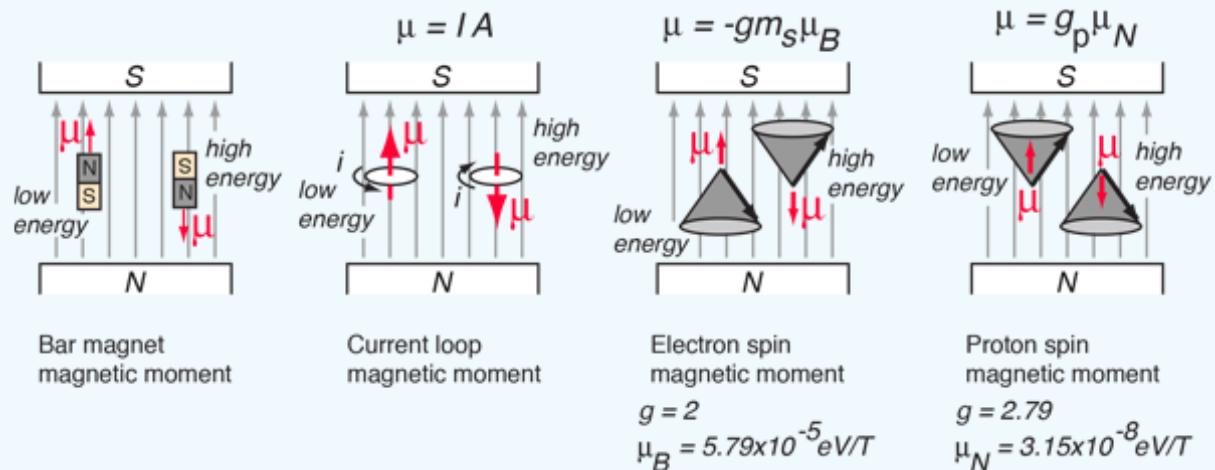


Figura 9.3: Momentos dipolares magnéticos en presencia de un campo magnético externo. En las dos figuras de la derecha vemos que, dependiendo de la carga de la partícula, el momento dipolar puede apuntar en la misma dirección o en la contraria al espín. Figura tomada de <http://hyperphysics.phy-astr.gsu.edu/hbase/Nuclear/nmr.html>

Vemos a la demostración de que el valor esperado del espín precesa entorno al campo magnético:

Demostración: (Precesión de Larmor, Griffiths [7] ejemplo 4.3)

Supongamos una partícula con espín 1/2 sometida a un campo magnético en la dirección \vec{z} , i.e. $\vec{B}_0 = B_0 \hat{z}$. El Hamiltoniano estará dado por

$$\mathcal{H} = -\gamma B_0 S_z = -\frac{\gamma B_0 \hbar}{2} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (9.13)$$

Los autovectores y autovalores (energías) de este Hamiltoniano son

$$\begin{cases} |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ con energía } E_{|0\rangle} = -(\gamma B_0 \hbar)/2 \\ |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \text{ con energía } E_{|1\rangle} = +(\gamma B_0 \hbar)/2 \end{cases} \quad (9.14)$$

La energía es menor cuando el momento dipolar es paralelo al campo magnético, al igual que en el caso clásico.

Como el Hamiltoniano es independiente del tiempo, la solución general para la ecuación de Schrödinger dependiente de tiempo,

$$i\hbar \frac{\partial |\Psi\rangle}{\partial t} = \mathcal{H} |\Psi\rangle \quad (9.15)$$

puede expresarse en función de estados estacionarios:

$$|\Psi(t)\rangle = ae^{-iE_{|0\rangle}t/\hbar} |0\rangle + be^{-iE_{|1\rangle}t/\hbar} |1\rangle = \begin{bmatrix} ae^{i\gamma B_0 t/2} \\ be^{i\gamma B_0 t/2} \end{bmatrix} \quad (9.16)$$

Las constantes a y b se pueden determinar mediante las condiciones iniciales:

$$|\Psi(t=0)\rangle = \begin{bmatrix} a \\ b \end{bmatrix} \quad (9.17)$$

(por supuesto, $|a|^2 + |b|^2 = 1$). Sin pérdida de generalidad, podemos elegir $a = \cos(\alpha/2)$ y $b = \sin(\alpha/2)$, donde α es un ángulo fijo cuyo significado físico veremos dentro de poco. Tenemos entonces

$$|\Psi(t)\rangle = \begin{bmatrix} \cos(\alpha/2)e^{i\gamma B_0 t/2} \\ \sin(\alpha/2)e^{-i\gamma B_0 t/2} \end{bmatrix} \quad (9.18)$$

Para ver qué está pasando con el espín bajo la evolución de este Hamiltoniano, calculemos el calor esperado del espín \vec{S} como función del tiempo.

$$\begin{aligned} \langle S_x \rangle &= \langle \Psi(t) | S_x | \Psi(t) \rangle = \\ &= \begin{bmatrix} \psi_0^* & \psi_1^* \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \psi_0 \\ \psi_1 \end{bmatrix} \\ &= \begin{bmatrix} ae^{-i\gamma B_0 t/2} & be^{-i\gamma B_0 t/2} \end{bmatrix} \frac{\hbar}{2} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \cos(\alpha/2)e^{i\gamma B_0 t/2} \\ \sin(\alpha/2)e^{-i\gamma B_0 t/2} \end{bmatrix} \\ &= \frac{\hbar}{2} \sin \alpha \cos(\gamma B_0 t) \end{aligned}$$

De forma análoga,

$$\begin{aligned} \langle S_y \rangle &= -\frac{\hbar}{2} \sin \alpha \sin(\gamma B_0 t) \\ \langle S_z \rangle &= \frac{\hbar}{2} \cos \alpha \end{aligned} \quad (9.19)$$

Con lo cual, $\langle \vec{S} \rangle$ está inclinado un ángulo α respecto al eje \hat{z} , y precesa al rededor del campo con una frecuencia

$$f_0 = \omega_0/2\pi = \gamma B_0/2\pi \quad (9.20)$$

(la frecuencia de Larmor) al igual que en el caso clásico (una peonza inclinada). No tenemos ninguna sorpresa aquí, pues el teorema de Ehrenfest nos asegura que los valores esperados evolucionan de acuerdo a las leyes clásicas del movimiento (algunos autores limitan esta afirmación solo al par de ecuaciones $\langle p \rangle = md\langle x \rangle/dt$ y $\langle -\partial V/\partial x \rangle = d\langle p \rangle/dt$) Puede verse esto en el Griffiths, [7].

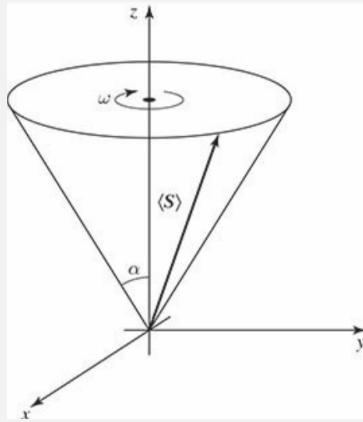


Figura 9.4: Precesión de Larmor en un campo magnético uniforme.

■

Ejercicio 38 Verifica que la Ec. (9.16) es solución de la ecuación de Schrödinger, Ec. (9.15)

El Hamiltoniano de espín para una molécula con n núcleos desacoplados viene dado por

$$\mathcal{H}_0 = - \sum_{i=1}^n \omega_0^i S_z^i \quad (9.21)$$

Véase que, los espines de especies nucleares diferentes (espines heteronucleares) tendrán diferentes valores de ω_0^i , así como también es posible que los espines de la misma especie nuclear (espines homonucleares) que formen una molécula tengan también diferentes valores de esta frecuencia.

9.3.1.2. Espines que interactúan.

Para espines nucleares en moléculas, la naturaleza nos proporciona dos mecanismos de interacción diferentes que vamos a describir:

- **Interacción directa dipolo-dipolo.**

La interacción magnética dipolo-dipolo es similar a la interacción entre dos barras magnéticas cercanas. Tiene lugar puramente a través del espacio (no se requiere ningún medio para esta interacción) y depende del vector internuclear \vec{r}_{ij} que conecta los dos núcleos i y j , tal y como describe el Hamiltoniano

$$\mathcal{H}_D = \sum_{i < j} \frac{\mu_0 \gamma_i \gamma_j}{4\pi |\vec{r}_{ij}|^3 \hbar} \left[\vec{S}^i \cdot \vec{S}^j - \frac{3}{|\vec{r}_{ij}|^2} (\vec{S}^i \cdot \vec{r}_{ij}) (\vec{S}^j \cdot \vec{r}_{ij}) \right] \quad (9.22)$$

donde μ_0 es la permeabilidad magnética habitual del espacio libre.

No vamos a entrar en dalles sobre esta interacción, pues usualmente se promedia y no tiene efecto.

- **Interacción del contrato de Fermi mediada por electrones (acoplamiento J).**

El segundo mecanismo de interacción entre los espines nucleares de una molécula es el acoplamiento J o acoplamiento escalar. Esta interacción está mediada por los electrones compartidos en los enlaces químicos entre los átomos, y es debido al solapamiento de la función de onda del electrón compartido con los dos núcleos acoplados. Es decir, una interacción de contacto de Fermi. La fuerza de acoplamiento de enlace J depende de la especie nuclear respectiva y disminuye con el número de enlaces químicos que separan los núcleos.

El Hamiltoniano es

$$\mathcal{H} = \frac{1}{\hbar} \sum_{i < j} 2\pi J_{ij} S^i S^j = \frac{1}{\hbar} \sum_{i < j} 2\pi J_{ij} (S_x^i S_x^j + S_y^i S_y^j + S_z^i S_z^j) \quad (9.23)$$

donde J_{ij} es la fuerza de acoplamiento entre los espines i y j . Cuando $|\omega_0^i - \omega_0^j|$ es mucho más grande que el acoplamiento J_{ij} ($|\omega_0^i - \omega_0^j| \gg 2\pi|J_{ij}|$), los acoplamiento transversos se pueden despreciar. De esta forma, el Hamiltoniano se simplifica

$$\mathcal{H}_j = \frac{1}{\hbar} \sum_{i < j}^n 2\pi J_{ij} S_z^i S_z^j$$

(9.24)

Esta condición ($|\omega_0^i - \omega_0^j| \gg 2\pi|J_{ij}|$) se cumple fácilmente para los espines heteronucleares y que también puede cumplirse para las moléculas homonucleares pequeñas.

La interpretación del término de acoplamiento escalar de la Ec. (9.24) es que un espín “siente” un campo magnético estático a lo largo de $\pm z$ producido por los espines vecinos, además del campo \vec{B}_0 aplicado externamente. Este campo adicional desplaza los niveles de energía como podemos ver en la Fig. 9.5. Como resultado, la frecuencia de Larmor del espín i se mueve en una cantidad $-J_{ij}/2$ si el espín j está en el estado $|0\rangle$ (líneas rojas) y una cantidad $+J_{ij}/2$ si está en el estado $|1\rangle$ (líneas azules).

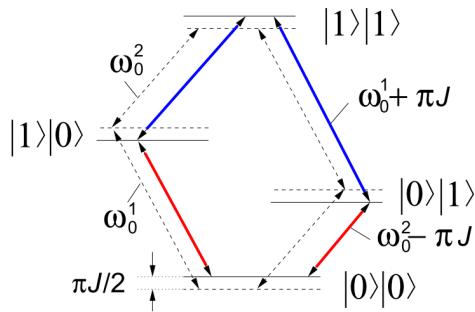


Figura 9.5: Diagrama de niveles de energía para dos espines desacoplados (líneas discontinuas) y dos espines acoplados (líneas continuas) por un Hamiltoniano de la forma de la Ec. (9.24) (en unidades de \hbar). Marcadas en rojo están las dos transiciones en las que el espín que no cambia está en el estado $|0\rangle$, mientras que en azul están en las que el espín que no cambia está en el estado $|1\rangle$. Vemos que la energía de transición de las primera disminuye, mientras que de las segundas aumenta.

9.3.1.3. Hamiltoniano completo.

En resumen, la forma más simple del Hamiltoniano para un sistema de n espines nucleares acoplados

es, pues (de las Ecs. (9.21) y (9.24))

$$\boxed{\mathcal{H}_{sys} = - \sum_i^n \omega_0^i S_z^i + \frac{1}{\hbar} \sum_{i < j} 2\pi J_{ij} S_z^i S_z^j} \quad (9.25)$$

9.3.2. Hamiltoniano de control

9.3.2.1. Campos de radiofrecuencia

Pasemos ahora a los mecanismos físicos para controlar el sistema de RMN. El estado de una partícula de espín 1/2 en un campo magnético estático \vec{B}_0 a lo largo del eje \hat{z} puede manipularse aplicando un campo electromagnético $\vec{B}_1(t)$ que gira en el plano $\hat{x} - \hat{y}$ a frecuencia ω_{rf} , en o cerca de la frecuencia de precesión del espín ω_0 . El Hamiltoniano de espín correspondiente al campo de radiofrecuencia (RF) es, análogo a la Ec. (9.12) para el campo estático B_0 ,

$$\mathcal{H}_{rf} = -\gamma B_1 [\cos(\omega_{rf} + \phi) S_x + \sin(\omega_{rf} + \phi) S_y] \quad (9.26)$$

donde ϕ es la fase del campo de RF, y B_1 su amplitud. Para n espines tenemos

$$\boxed{\mathcal{H}_{rf} = - \sum_i^n \gamma_i B_1 [\cos(\omega_{rf} t + \phi) S_x^i + \sin(\omega_{rf} t + \phi) S_y^i]}, \quad (9.27)$$

Nota: implementación de un campo rotante en el laboratorio

En la práctica, se aplica un campo magnético que oscila a lo largo de un eje fijo en el laboratorio, perpendicular al campo magnético estático. Este campo oscilante puede descomponerse en dos campos contrarrotatorios, uno de los cuales gira a ω_{rf} en la misma dirección que el espín y, por tanto, puede establecerse en resonancia con el espín o cerca de ella. La otra componente gira en la dirección opuesta y, por lo tanto, está muy lejos de la resonancia (en aproximadamente $2\omega_0$). Como veremos, su único efecto es un desplazamiento insignificante de la frecuencia de Larmor, llamado desplazamiento Bloch-Siegert.

9.3.2.2. Sistema de referencia rotante (rotating frame).

El movimiento de un espín nuclear individual sometido a un campo magnético estático y giratorio es bastante complejo cuando se describe en el sistema de coordenadas habitual del laboratorio (el marco del laboratorio). Sin embargo, se simplifica mucho describiendo el movimiento en un **sistema de coordenadas que gira** alrededor de \hat{z} a frecuencia ω_{rf} (el **rotating frame**):

$$|\psi\rangle^{rot} = \exp(-i\omega_{rf}tS_z/\hbar)|\psi\rangle \quad (9.28)$$

Para un solo espín libre, el Hamiltoniano será la suma (9.21) y (9.27) (con $n = 1$), es decir,

$$\mathcal{H} = -\omega_0 S_z - \omega_1 [\cos(\omega_{rf} t + \phi) S_x + \sin(\omega_{rf} t + \phi) S_y] \quad (9.29)$$

Este Hamiltoniano junto con el estado en el sistema de referencia estático cumplen la ecuación de Schrödinger

$$i\hbar \frac{d|\psi\rangle}{dt} = \mathcal{H} |\psi\rangle \quad (9.30)$$

Sustituyendo el cambio de variable de la Ec. (9.25) en la ecuación de Schrödinger, podemos calcular como tendría que ser el \mathcal{H}^{rot} que cumpla

$$i\hbar \frac{d|\psi\rangle^{rot}}{dt} = \mathcal{H}^{rot} |\psi\rangle^{rot} \quad (9.31)$$

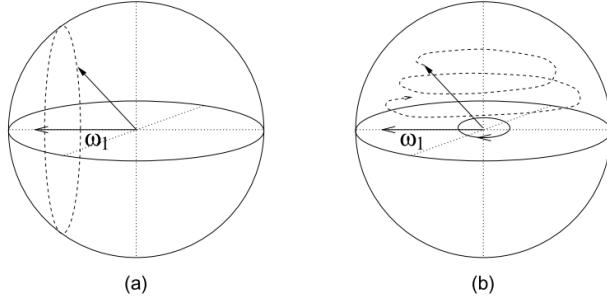


Figura 9.6: Nutación de un espín sometido a un campo de RF transversal observada en el marco de rotación (a) y observada en el marco de laboratorio (b).

Puede ver que el resultado es

$$\mathcal{H}^{rot} = -(\omega_0 - \omega_{rf})S_z - \omega_1 [\cos(\phi)S_x + \sin(\phi)S_y] \quad (9.32)$$

Naturalmente, el campo de RF se encuentra a lo largo de un eje fijo en el sistema de referencia que gira a ω_{rf} . El movimiento del espín visto desde un sistema de referencia o el otro es diferente.

- En el **sistema laboratorio (en reposo)**, al aplicar el campo magnético rotante \vec{B}_1 lo que sucede es que *el valor esperado del espín rota en espiral, bajando por la esfera*, como podemos ver en la Fig. 9.6b.
- En el **sistema rotante**, tenemos dos casos:
 - Caso **resonante** ($\omega_{rf} = \omega_0$): en este caso el primer término de la Ec. (9.32) desaparece. En este caso, un observador en el sistema rotante verá el espín simplemente *precesar* alrededor de \vec{B}_1 (Fig. 9.6a), un movimiento llamado nutación. La elección de ϕ controla el eje de nutación.
 - Caso **fuerza de resonancia**: Si el campo de RF está fuera de resonancia con respecto a la frecuencia de espín en $\Delta\omega = \omega_0 - \omega_{rf}$, el espín precesa en el marco de rotación alrededor de un eje inclinado con respecto al eje \vec{z} en un ángulo

$$\alpha = \arctan(\omega_1 / \Delta\omega) \quad (9.33)$$

como se ilustra en la Fig. 9.7.

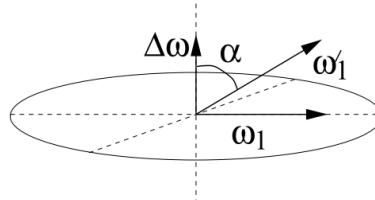


Figura 9.7: Eje de rotación (en el marco giratorio) durante un pulso de radiofrecuencia fuera de resonancia.

En este último caso se deduce que el **campo de RF no tiene prácticamente ningún efecto sobre espines que están lejos de la resonancia**, ya que α es muy pequeño cuando $|\Delta\omega| \gg \omega_1$ (ver Fig.

9.8). Si todos los espines tienen frecuencias de Larmor bien separadas, en principio podemos **rotar selectivamente** cualquier qubit sin rotar los otros espines.

Los pulsos moderadamente fuera de resonancia ($|\Delta\omega| \approx \omega_1$) hacen girar el espín, pero debido a la inclinación del eje de rotación, un solo pulso de este tipo no puede, por ejemplo, voltear un espín de $|0\rangle$ a $|1\rangle$ (véase de nuevo la Fig. 9.8). Por supuesto, los pulsos fuera de resonancia también pueden ser útiles, por ejemplo para la implementación directa de rotaciones sobre un eje fuera del plano $\hat{x} - \hat{y}$.

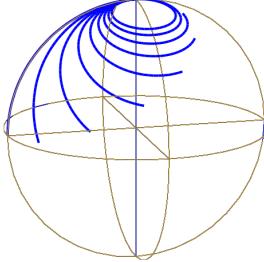


Figura 9.8: Trayectoria en la esfera de Bloch (en el sistema de referencia rotante) descrita por un qubit inicialmente en $|0\rangle$ (a lo largo de $+\hat{z}$), después de aplicar un pulso de $250 \mu s$ de intensidad $\omega_1 = 1 \text{ kHz}$ fuera de resonancia en $0, 0.5, 1, \dots, 4 \text{ kHz}$. En resonancia, el pulso produce una rotación 90. Lejos de la resonancia, el qubit apenas rota alejándose de $|0\rangle$.

Por otro lado, podemos elegir trabajar en **un sistema de referencia que rote a ω_0** (en vez de a ω_{rf} , donde

$$\boxed{\mathcal{H}^{rot} = -\omega_1 [\cos((\omega_{rf} - \omega_0)t + \phi)S_x + \sin((\omega_{rf} - \omega_0)t + \phi)S_y]} \quad (9.34)$$

Esta transformación no da un Hamiltoniano de RF independiente del tiempo (a menos que $\omega_{rf} = \omega_0$), como fue el caso para \mathcal{H}^{rot} en la Ec. 9.32). Sin embargo, es un punto de partida natural para la extensión al caso de **múltiples espines**, donde puede introducirse un marco de rotación separado para cada espín:

$$|\psi\rangle^{rot} = \left[\prod_i \exp(-i\omega_0^i t S_z^i / \hbar) \right] |\psi\rangle \quad (9.35)$$

En presencia de múltiples campos de RF indexados con r , el Hamiltoniano en este marco de referencia rotante e ω_0 nos queda

$$\boxed{\mathcal{H}^{rot} = \sum_{i,r} -\omega_1^r [\cos((\omega_{rf}^r - \omega_0^i)t + \phi^r)S_x^i + \sin((\omega_{rf}^r - \omega_0^i)t + \phi^r)S_y^i]} \quad (9.36)$$

donde las amplitudes ω_i^r y las fases ϕ^r están bajo control.

El Hamiltoniano del sistema de la Ec. (9.25) se simplifica en el sistema de referencia multi-rotante de la Ec. (9.35): el término S_z^i desaparece dejando solo el término de acoplamiento $J_{ij}S_z^i S_z^j$, que permanecen invariantes.

En resumen, en el sistema de referencia multi-rotante, el Hamiltoniano de *NMR* $\mathcal{H} = \mathcal{H}_{sys} + \mathcal{H}_{control}$ toma la forma

$$\boxed{\begin{aligned} \mathcal{H}_{sys} &= \frac{1}{\hbar} \sum_{i < j} 2\pi J_{ij} S_z^i S_z^j \\ \mathcal{H}_{control} &= \sum_{i,j} -\omega_1^r [\cos((\omega_{rf}^r - \omega_0^i)t + \phi^r)S_x^i + \sin((\omega_{rf}^r - \omega_0^i)t + \phi^r)S_y^i] \end{aligned}} \quad (9.37)$$

9.4. Técnicas de pulsos elementales.

Esta sección inicia nuestra discusión del tema principal de este artículo, una revisión de las **técnicas de control** desarrolladas en la computación cuántica de RMN para sistemas cuánticos acoplados de dos niveles. Comenzamos con una rápida visión general del lenguaje de los circuitos cuánticos y sus importantes teoremas de universalidad, luego lo conectamos con el lenguaje de las secuencias de pulsos tal y como se utiliza en la RMN, e indicamos cómo se pueden simplificar las secuencias de pulsos. Las principales aproximaciones empleadas en esta sección son que los pulsos pueden ser **fuertes** comparados con el Hamiltoniano del sistema mientras se dirigen selectivamente a **un solo qúbit** a la vez, y pueden ser perfectamente implementados.

9.4.1. Control cuántico, circuitos y pulsos

El objetivo del control cuántico, en el contexto de la computación cuántica, es la implementación de una **transformación unitaria** U , especificada en términos de una secuencia $U = U_k U_{k-1} \dots U_2 U_1$ de puertas cuánticas estándar U_i , que actúan localmente (normalmente sobre uno o dos qubits) y son sencillas de implementar. Como es habitual en las operaciones unitarias, las U_i se ordenan en el tiempo de derecha a izquierda.

9.4.1.1. Puertas cuánticas y circuitos

La rotación básica de un qúbit simple son las rotaciones de la forma (4.1), es decir

$$R_{\hat{n}}(\theta) = \exp \left[-\frac{i\theta \hat{n} \cdot \vec{\sigma}}{2} \right] \quad (9.38)$$

Como ya hemos comentado en la sección 4.1.2, usando la **parametrización de Euler** podemos general cualquier rotación sobre la esfera de Bloch usando tres rotaciones sobre dos ejes (Ec. (4.4)). En esa sección elegimos las rotaciones en \hat{z} y \hat{y} , pero en esta vamos a usar las siguientes:

$$U = e^{i\alpha} R_x(\beta) R_y(\gamma) R_x(\delta) \quad (9.39)$$

Como también comentamos, la puerta básica de dos qúbits es la CNOT (Ec. (6.24)).

Un teorema básico de la computación cuántica es que salvo una fase global irrelevante, cualquier U que actúe sobre n qubits puede componerse a partir de puertas U_{CNOT} y $R_{\hat{n}}(\theta)$ [9]. Así, el problema del control cuántico puede reducirse a la implementación de U_{CNOT} y rotaciones de qúbits simples, donde se requieren al menos dos rotaciones no triviales (veremos esto en detalle en la sección 11.5). Se conocen otros conjuntos de puertas universales de este tipo, pero éste es el que se ha empleado en la RMN.

9.4.1.2. Implementación de puertas de un qúbit.

Las rotaciones en qúbits individuales pueden implementarse directamente en el sistema de referencia rotante utilizando pulsos de RF. Del Hamiltoniano de control, Ec. (9.37), se deduce que cuando se aplica un campo de RF de amplitud ω_1 a un sistema de un único espín con $\omega_{rf} = \omega_0$, el espín evoluciona bajo la transformación

$$U = e^{i \frac{\mathcal{H}_{control}}{\hbar}} = \exp \left[i\omega_1 (\cos \phi S_x + \sin \phi S_y) \frac{t_{p\omega}}{\hbar} \right] \quad (9.40)$$

donde $t_{p\omega}$ es la **anchura del pulso** (o longitud), la duración temporal del pulso de RF. U describe una rotación en la esfera de Bloch de un ángulo θ proporcional al producto de $t_{p\omega}$ y $\omega_1 = \gamma B_1$, y al rededor de un eje en el plano $\hat{x} - \hat{y}$ determinado por una fase ϕ .

- Así, un pulso con fase $\phi = \pi$ y $\omega_1 t_{p\omega} = \pi/2$ realizará $R_x(90)$ (ver Ec. (4.1)), que es una rotación 90° sobre \hat{x} , denotada para abreviar como \sqrt{X} .
- Un pulso similar pero dos veces más largo realiza una rotación $R_x(180) = X$.
- Cambiando la fase del pulso RF a $\phi = -\pi/2$, pueden implementarse de forma similar \sqrt{Y} e Y
- Para $\phi = 0$ y $\omega_1 t_{p\omega} = \pi/2$, se obtiene una rotación negativa alrededor de \hat{x} : $R_x(-90) = \sqrt{X^\dagger}$.
- De forma similar $\phi = \pi/2$ y $\omega_1 t_{p\omega} = \pi/2$ da $\sqrt{Y^\dagger}$.

Para sistemas multiqubit, se utilizan subíndices para indicar sobre qué qúbit actúa la operación, por ejemplo, Z_3^\dagger es una rotación de 180° del qúbit 3 alrededor de $-\hat{z}$.

Por tanto, no es necesario aplicar el campo de RF a lo largo de diferentes ejes espaciales en el marco del laboratorio para realizar rotaciones \hat{x} y \hat{y} . Más bien, la fase del campo de RF determina el eje de nutación en el marco de rotación. Además, nótese que sólo importa la fase relativa entre pulsos aplicados al mismo espín. La fase absoluta del primer impulso en un espín determinado no tiene importancia en sí misma. Sólo establece una referencia de fase con la que deben compararse las fases de todos los pulsos posteriores en ese mismo espín, así como la lectura de ese espín.

Anteriormente señalamos que la capacidad de implementar rotaciones arbitrarias sobre $|x\rangle$ y $|y\rangle$ es suficiente para realizar rotaciones arbitrarias de un solo qúbit (Ec. 9.40). Dado que las rotaciones \hat{z} son muy comunes, existen dos descomposiciones explícitas útiles de $R_z(\theta)$ en términos de las rotaciones \hat{x} y \hat{y} :

$$R_z(\theta) = \sqrt{X} R_y(\theta) \sqrt{X^\dagger} = \sqrt{Y} R_x(-\theta) \sqrt{Y^\dagger} \quad (9.41)$$

9.4.1.3. Implementación de puertas a dos qubits

La puerta de dos qubits más natural es la generada directamente por el Hamiltoniano de acoplamiento espín-espín. Para espines nucleares en una molécula en solución líquida, el Hamiltoniano de acoplamiento viene dado por la Ec. (9.24) (tanto en el sistema laboratorio como en el sistema de rotación), a partir de la cual obtenemos el operador de evolución temporal $U_J(t) = \exp[-i2\pi JS_z^1 S_z^2 t/\hbar^2]$, o en forma matricial

$$U_J(t) = \begin{bmatrix} e^{-i\pi Jt/2} & 0 & 0 & 0 \\ 0 & e^{+i\pi Jt/2} & 0 & 0 \\ 0 & 0 & e^{+i\pi Jt/2} & 0 \\ 0 & 0 & 0 & e^{-i\pi Jt/2} \end{bmatrix} \quad (9.42)$$

Si se permite que esta evolución ocurra durante un tiempo $t = 1/2J$ se obtiene una transformación conocida como **puerta de fase controlada**, salvo un desplazamiento de fase de 90° en cada qubit y una fase global (y por tanto irrelevante):

$$U_{CPHASE} = \sqrt{-i} \sqrt{Z_1^\dagger} \sqrt{Z_2^\dagger} U_J(1/2J) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad (9.43)$$

Esta puerta es equivalente a la conocida puerta CNOT salvo un cambio de base del qubit objetivo y

un desplazamiento de fase en el qúbit de control:

$$\begin{aligned}
 U_{CNOT} &= iZ_1\sqrt{Y_2^\dagger}U_{CPHASE}\sqrt{Y_2} \\
 &= iZ_1\sqrt{Y_2^\dagger}\left[\sqrt{-i}\sqrt{Z_1^\dagger}\sqrt{Z_2^\dagger}U_J(1/2J)\right]\sqrt{Y_2} \\
 &= \sqrt{i}\sqrt{Z_1}\sqrt{Z_2^\dagger}\sqrt{X_2}U_j(1/2J)\sqrt{Y_2} \\
 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}
 \end{aligned}$$

El núcleo de esta secuencia, $\sqrt{X_2}U_j(1/2J)\sqrt{Y_2}$, puede entenderse gráficamente mediante la Fig. 9.9, suponiendo que los espines comienzan a lo largo de $\pm\hat{z}$. Primero, un pulso en el espín 2 que lo rota de \hat{z} a \hat{y} . A continuación, se deja que el sistema de espín evolucione libremente durante $1/2J_{12}$ segundos. Como la frecuencia de precesión del espín 2 se desplaza $\pm J_{12}/2$ dependiendo de si el espín 1 está en $|1\rangle$ o $|0\rangle$ (ver Fig. 9.5), el espín 2 llegará en $1/2J_{12}$ segundos a $+\hat{y}$ o $-\hat{y}$, dependiendo del estado del espín 1. Finalmente, un pulso de 90° sobre el espín 2 alrededor del eje \hat{x} hace girar el espín 2 de nuevo a $+\hat{z}$ si el espín 1 está en $|0\rangle$, o a $-\hat{z}$ si el espín 1 está en $|1\rangle$.

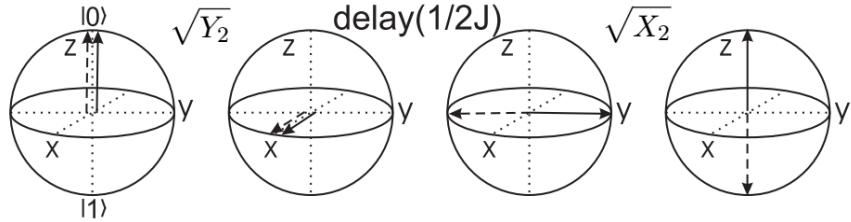


Figura 9.9: Representación en esfera de Bloch del funcionamiento de la puerta $CNOT_{12}$ entre dos qubits 1 y 2 acoplados por $2\pi JS_z^1S_z^2/\hbar$. Aquí se representa el estado del qubit 2 (el qubit objetivo de la CNOT), que comienza en $|0\rangle$ (a lo largo de \hat{z}) y se representa en un marco de referencia que gira alrededor de \hat{z} a $\omega_0^2/2\pi$. Las flechas continuas y discontinuas corresponden al caso en que el qubit 1 (de control) está en $|0\rangle$ y $|1\rangle$ respectivamente. Figura tomada de [16].

Ejercicio 39 Vamos a verificar el paso de la esfera 2 a la 3 de la Fig. 9.9. Multiplica la matriz de la Ec. (9.42) por los dos estados de la segunda esfera de la Fig. 9.9. Toma $t = 1/2J$ y escribe los estados resultantes de la forma 3.7 (recuerda que las fases globales no son importantes)

El resultado neto es que el espín 2 se invierte si y sólo si el espín 1 está en $|1\rangle$, lo que corresponde exactamente a la tabla de verdad clásica para la CNOT. Las rotaciones \hat{z} adicionales de la Ec. (9.44) son necesarias para dar a todos los elementos de U_{CNOT} la misma fase, por lo que la secuencia también funciona para estados de entrada en superposición.

Si el Hamiltoniano de interacción espín-espín no es de la forma $S_z^iS_z^j$ sino que contiene también componentes transversales (como en las Ec. (9.22)), se necesitan otras secuencias de pulsos más complicadas para realizar las puertas CPHASE y CNOT.

Si dos espines no están directamente acoplados entre sí, todavía es posible realizar una puerta CNOT entre ellos, siempre y cuando exista una red de acoplamientos que conecte los dos qubits. Por ejemplo, supongamos que queremos realizar una puerta CNOT con el qubit 1 como control y el qubit 3 como objetivo, $CNOT_{13}$, pero 1 y 3 no están acoplados entre sí. Si ambos están acoplados al qubit 2, como en la red de acoplamiento de la Fig. 9.10b, podemos primero intercambiar el estado de los qubits 1 y

2 (mediante la secuencia CNOT₁₂ CNOT₂₁ CNOT₁₂, es decir, una puerta SWAP como la de la Fig. 11.8), luego realizar un CNOT₂₃, y finalmente intercambiar de nuevo los qubits 1 y 2. El efecto neto es CNOT₁₃. Por extensión, se requieren como máximo $O(n)$ operaciones de intercambio para realizar una CNOT entre cualquier par de qubits en una cadena de n espines con sólo acoplamientos de vecino más cercano (Fig. 9.10b). Las operaciones SWAP también se pueden utilizar para realizar pueras de dos qubits entre dos qubits cualesquiera que estén acoplados a un qubit “bus” común (Fig. 9.10c).

Por el contrario, si un qubit está acoplado a muchos otros qubits (Fig. 9.10a) y queremos realizar una CNOT entre sólo dos de ellos, debemos **eliminar el efecto de los acoplamientos restantes**. Esto se puede lograr utilizando la técnica de **reenfoque**, que ha sido ampliamente adoptada en una variedad de experimentos de RMN.

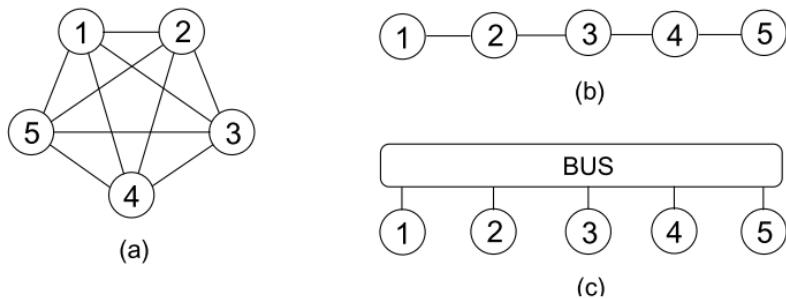


Figura 9.10: Tres posibles redes de acoplamiento entre cinco qubits. (a) Una red de acoplamiento completa. En la práctica, estas redes siempre tendrán un tamaño limitado, ya que las interacciones físicas tienden a disminuir con la distancia. (b) Una red de acoplamiento de vecino más próximo. Este tipo de cadenas lineales con acoplamientos de vecino más próximo, o variantes bidimensionales, se utilizan en muchas propuestas de estado sólido. (c) Acoplamiento a través de un “bus”. Es el caso, por ejemplo, de los esquemas de trampas de iones. Al igual que en el caso (a), el grado de libertad del bus sólo se acoplará bien a un número finito de qubits. Figura tomada de [16]

9.4.1.4. Reenfoque (refocusing): apagando las interacciones $S_z^i S_z^j$ indeseadas

Ya hemos visto que al estar los espines acoplados, estos evolucionan con el tiempo siguiendo la Ec. (9.42). Si queremos que esta evolución la sientan solo unos ciertos qubits, tenemos que eliminar el efecto de los términos de acoplamiento no deseados. Este es el caso, por ejemplo, de la CNOT. Como ya vimos, el paso intermedio de la CNOT es una evolución libre que solo deben de experimentar los dos qubits implicados en la CNOT.

El efecto de los términos de acoplamiento durante un intervalo de tiempo de evolución libre puede eliminarse mediante las denominadas técnicas de **reenfoque**. Para hamiltonianos de acoplamiento de la forma $S_z^i S_z^j$, como suele ocurrir en los experimentos de RMN de líquidos, el mecanismo de reenfoque puede entenderse a un nivel muy intuitivo.

Veamos primero dos formas de deshacer $S_z^i S_z^j$ en un sistema de dos qubits. En la Fig. 9.11a, la evolución del qubit 1 en el primer intervalo de tiempo τ se invierte en el segundo intervalo de tiempo, debido al pulso de 180º en el qubit 2. En la Fig. 9.11b, el qubit 1 continúa evolucionando en la misma dirección todo el tiempo, pero el primer pulso de 180º hace que los dos componentes del qubit 1 se reenfoquen al final del segundo intervalo de tiempo. El segundo pulso de 180º garantiza que ambos qubits vuelvan siempre a su estado inicial.

Matemáticamente, podemos ver cómo funciona el reenfoque de los acoplamientos J utilizando

$$X_1 U_J(\tau) X_1 = U_J(-\tau) = X_2 U_J X_2, \quad (9.44)$$

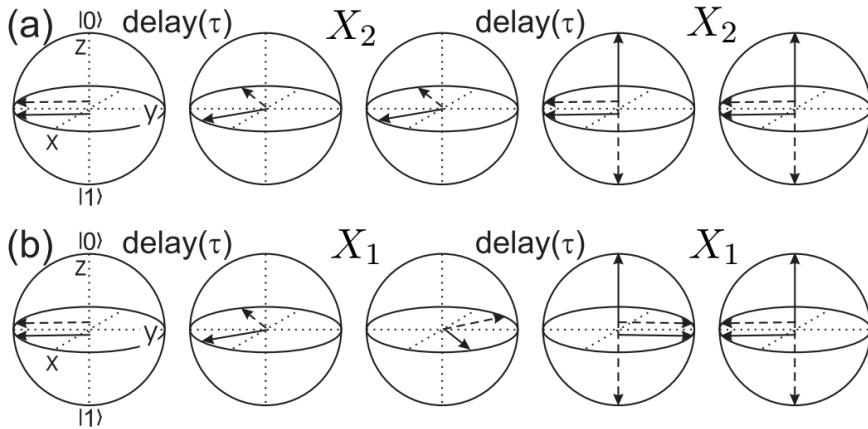


Figura 9.11: Representación en esfera de Bloch del funcionamiento de dos esquemas sencillos para reenfocar el acoplamiento entre dos qubits acoplados. El diagrama muestra la evolución del **qúbit 1** (en el sistema rotante) inicialmente a lo largo de $-\hat{y}$, cuando el qúbit 2 está en $|0\rangle$ (sólido) o en $|1\rangle$ (discontinuo). Los pulsos de reenfoque puede aplicarse tanto al qúbit 2 (a) como al qúbit 1 (b).

que nos lleva a

$$X_1 U_J(\tau) X_1 U_J(\tau) = I = X_2 U_J(\tau) X_2 U_J(\tau) \quad (9.45)$$

Ejercicio 40 Partiendo de los estados de las primeras esferas de Bloch de las figuras 9.11a y 9.11b, (sería el estado $|y-\rangle$ de la Ec. (3.4)), aplica una a una las puertas de las figuras, comprobando que estas son correctas. (Nota: no hace falta darle un valor a τ , simplemente dejarlo como parámetro libre)

Reemplazando todas las X_i por Y_i , las secuencias funcionan igual. Sin embargo, si utilizamos unas veces X_i y otras Y_i , obtendremos la matriz identidad salvo algunos desplazamientos de fase. Además, si aplicáramos pulsos en ambos qubits simultáneamente, por ejemplo $X_1 X_2 U_J(\tau) X_1 X_2 U_J(\tau)$, el acoplamiento no se eliminaría.

La Fig. 9.12 da una idea de las técnicas de reenfoque en un sistema multi-qubit. Específicamente, este esquema preserva el efecto de J_{12} , mientras que inactiva efectivamente todos los demás acoplamientos. La idea subyacente es que un acoplamiento entre los espines i y j actúa “hacia delante” durante los intervalos en los que ambos espines tienen el mismo signo en el diagrama, y actúa “a la inversa” siempre que los espines tienen signos opuestos. Cuando un acoplamiento actúa hacia delante y hacia atrás durante el mismo tiempo, no tiene ningún efecto neto.

Se han desarrollado métodos sistemáticos para diseñar esquemas de reenfoque para sistemas multi-qubit específicamente con el propósito de la computación cuántica. El esquema más compacto se basa en matrices de Hadamard [17]- [18]. Una matriz de Hadamard de orden n , denotada por $H(n)$, es una matriz $n \times n$ con entradas ± 1 , tal que

$$H(n) H(n)^T = nI \quad (9.46)$$

Las filas son, por tanto, ortogonales por pares, y dos filas cualesquiera coinciden exactamente en la mitad de las entradas. Identificando $+1$ y -1 con $+$ y $-$ como en el diagrama de la Fig. 9.12, vemos que $H(n)$ da un esquema de desacoplamiento válido para n espines utilizando sólo n intervalos de

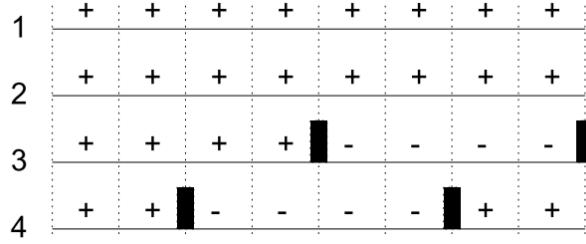


Figura 9.12: Esquema de reenfoque para un sistema de 4 espines, diseñado para preservar el efecto de la interacción J_{12} todo el tiempo pero neutralizando el efecto del resto de las J_{ij} . El intervalo está dividido en segmentos de igual duración, y los signos “+” y “-” indican si un espín está en suposición original o del revés. Los rectángulos negros representan pulsos de 180° , que dan la vuelta al correspondiente espín.

tiempo. Un ejemplo de $H(12)$ es

$$\left[\begin{array}{cccccccccccc} + & + & + & + & + & + & + & + & + & + & + & + \\ + & + & + & - & - & + & - & - & + & - & - & + \\ + & + & + & + & - & - & - & + & - & + & - & - \\ + & - & + & + & + & - & - & - & + & - & + & - \\ + & - & - & + & + & + & - & - & - & + & - & + \\ + & + & - & - & + & + & - & + & - & - & + & - \\ + & - & - & - & - & - & - & + & + & + & + & + \\ + & - & + & - & - & + & + & - & - & + & + & - \\ + & + & - & + & - & - & + & - & - & - & + & + \\ + & - & + & - & + & - & + & + & - & - & - & + \\ + & - & - & + & - & + & + & + & + & - & - & - \\ + & + & - & - & + & - & + & - & + & + & - & - \end{array} \right] \quad (9.47)$$

Si queremos que el acoplamiento entre un par de qubits permanezca activo mientras se elimina el efecto de todos los demás acoplamientos, podemos simplemente utilizar la misma fila de $H(n)$ para esos dos qubits.

$H(n)$ no existe para todos los n , pero siempre podemos encontrar una secuencia de desacoplamiento para n qubits tomando las primeras n filas de $H(\bar{n})$, siendo \bar{n} el menor número entero que satisface $\bar{n} \geq n$ con $H(\bar{n})$ conocida. A partir de las propiedades de las matrices de Hadamard, podemos demostrar que \bar{n}/n es siempre próximo a 1 (ver [17]). Así que los esquemas de desacoplamiento para n espines requieren \bar{n} intervalos de tiempo y no más de $\bar{n}n$ pulsos de 180° .

Terminamos esta subsección con tres observaciones adicionales. En primer lugar, cada qubit estará generalmente acoplado a no más de un número fijo de otros qubits, ya que las intensidades de acoplamiento tienden a disminuir con la distancia. En este caso, todos los esquemas de reenfoque pueden simplificarse enormemente.

En segundo lugar, si las evoluciones hacia delante y hacia atrás bajo J_{ij} no son iguales en duración, se produce una evolución neta acoplada correspondiente al exceso de evolución hacia delante o hacia atrás. En principio, por tanto, podemos organizar cualquier esquema de reenfoque de forma que incorpore cualquier cantidad deseada de evolución acoplada para cada par de qubits.

En tercer lugar, las secuencias de reenfoque también pueden utilizarse para eliminar el efecto de los términos S_z^i en el Hamiltoniano. Por supuesto, estos términos desaparecen en principio si trabajamos en el marco de rotación múltiple (véase la Ec. (9.37)). Sin embargo, puede haber cierta dispersión en

las frecuencias de Larmor, por ejemplo debido a inhomogeneidades del campo magnético. Este efecto puede invertirse utilizando pulsos de reenfoque.

Capítulo 10

Decoherencia y desfase

En esta sección vamos a seguir el artículo recopilatorio [19].

10.1. Introducción

Un **autoestado** cuántico de un Hamiltoniano particular es, por definición, un estado estacionario, en el que la función de onda puede variar espacialmente, pero no decae en el tiempo. Un sistema cuántico puede estar en una superposición de sus autoestados con relaciones definidas de fase y amplitud entre los estados base. Por ejemplo,

$$|\psi\rangle = \frac{1}{\sqrt{3}}|0\rangle + e^{i7\pi/8}\sqrt{\frac{2}{3}}|1\rangle \quad (10.1)$$

Esta superposición de autoestados con relaciones de fase definidas se denomina **coherencia cuántica**. La **decoherencia** se refiere vagamente a cómo un sistema pierde estas características de coherencia cuántica. Por ejemplo, puede referirse al **decaimiento de la amplitud** (a menudo exponencial) y la desaparición asociada de un estado propio cuántico en el tiempo (en virtud de su interacción con el entorno, por ejemplo). O puede referirse a la **pérdida de coherencia de fase**, porque la relación de fase definida entre los estados superpuestos desaparece con el tiempo, lo que da lugar al desfase.

En un sistema cuántico aislado, la decoherencia sólo podría surgir de los **grados de libertad dinámicos despreciados** en el Hamiltoniano original utilizado para definir el estado cuántico. Es decir, de aquellos términos que despreciamos o no tenemos en cuenta. En un sistema no aislado, la decoherencia podría surgir de forma natural del acoplamiento entre el sistema y el entorno, debido al intercambio de energía entre el sistema y el entorno. Obsérvese que no es necesario que el entorno esté físicamente separados del “sistema”; es una práctica habitual en física dividir un sistema grande en subsistemas que estén “razonablemente aislados” (en algún sentido operativo bien definido) entre sí, es decir, que el Hamiltoniano de interacción que acopla los distintos subsistemas sea “débil” de una manera definida con precisión.

La decoherencia en mecánica cuántica ha recibido mucha atención recientemente en el contexto del interés actual por la computación cuántica y el tratamiento de la información. En un ordenador cuántico, un algoritmo se realiza típicamente aplicando operaciones unitarias sobre un conjunto de sistemas de dos niveles (qubits) que transportan información cuántica. Para la computación cuántica, estos qubits deben estar aislados de los demás grados de libertad que podrían perturbar su evolución temporal unitaria. En otras palabras, la **decoherencia en un ordenador cuántico debe ser mucho más lenta que una operación de puerta cuántica típica para que la computación cuántica tenga éxito**. La relación entre el tiempo de puerta y el tiempo de decoherencia debe ser inferior a $10^{-3} \approx 10^{-6}$. Así pues, el control de la decoherencia es un aspecto crucial del procesamiento

cuántico de la información

Los canales de decoherencia específicos dependen siempre del sistema, pero también existen muchas características comunes. Por ejemplo, la mayoría de las propuestas de computación cuántica implican **sistemas cuánticos de dos niveles (TLS)** que desempeñan el papel de los qubits. Estos TLS pueden ser niveles de espín electrónico, niveles orbitales electrónicos, niveles de espín nuclear, niveles de carga, direcciones de flujo magnético, etc. El algoritmo básico de computación cuántica en cada esquema implica manipulaciones dinámicas de estos TLS utilizando diversos medios externos para realizar operaciones de uno y dos qubits. Por lo tanto, es imperativo que el tiempo de decoherencia en estas dinámicas TLS sea mucho mayor que los tiempos de operación de los qubits.

Debido a la naturaleza de dos niveles de estos sistemas, es posible describir su decoherencia utilizando sólo dos escalas de tiempo de desfase denominadas T_1 y $T_2 (\leq T_1)$, que dan una descripción fenomenológica de la relajación de fase y de población en estos sistemas. Para un conjunto de TLS, también debe definirse otra escala de tiempo $T_2^* \leq T_2$, ya que algunos espines pueden rotar más rápido que otros, lo que conduce a una pérdida reversible de coherencia cuántica entre ellos. Estas dos escalas de tiempo (T_2^* y T_2) deben distinguirse cuidadosamente. En las mediciones macroscópicas, la cantidad observada suele ser T_2^* debido al promedio del conjunto sobre la respuesta de un gran número de TLS.

Debemos señalar que el uso de sólo dos tiempos de relajación casi nunca proporciona una descripción completa de la dinámica de un sistema realista de dos niveles acoplado a un entorno físico. Sin embargo, en varios TLS paradigmáticos, como los espines nucleares sondeados por **RMN (resonancia magnética nuclear)** o los espines electrónicos sondeados por **ESR (resonancia de espín electrónico)**, las constantes de relajación T_1 y T_2 proporcionan la descripción cualitativa adecuada de los anchos de línea de la señal, y son buenas representaciones operativas de los distintos canales de relajación. También observamos que en muchas situaciones de interés T_1 y T_2 podrían ser iguales (o bastante cercanos en valores).

10.2. Decoherencia y las ecuaciones de Bloch.

Dado que la decoherencia se produce en el dominio del tiempo, es natural utilizar escalas temporales para describir la fuerza de un canal de decoherencia y la intensidad con que la decoherencia afecta a una variable dinámica concreta. Si nos limitamos a un TLS simple, el problema de describir fenomenológicamente el efecto del acoplamiento débil a grados de libertad externos sobre la evolución temporal de este TLS contiene sólo unos pocos parámetros.

En particular, dos escalas de tiempo de relajación, T_1 y T_2 , se introdujeron y utilizaron ampliamente en el campo de la RMN, y después se utilizaron también de forma natural en la ESR y en la óptica cuántica. En estas disciplinas, o bien el campo magnético estático aplicado (que causa el desdoblamiento de Zeeman a lo largo de la dirección del campo) o bien los TLS naturales (por ejemplo, para fotones, donde la polarización longitudinal y transversal se produce de forma natural) definen una dirección longitudinal y otra transversal. T_1 y T_2 son entonces, respectivamente, los tiempos de relajación longitudinal y transversal para magnetizaciones en RMN y ESR, o la diferencia de población y polarización en óptica cuántica. Nótese que el uso de T_1 y T_2 para caracterizar la decoherencia sólo se aplica a la dinámica TLS.

La definición de T_1 y T_2 es bastante específica del sistema; de hecho, la definición estricta de T_1 y T_2 se aplica específicamente a las mediciones por resonancia magnética. Un fenómeno de decoherencia arbitrario puede requerir más o menos parámetros para describir el proceso de desfase. En general, en ausencia de campo magnético y en sistemas isótropos, $T_1 = T_2$. También hay que señalar que T_1 y T_2 son parámetros puramente fenomenológicos (que caracterizan la relajación longitudinal y transversal respectivamente), a los que podrían contribuir, en principio, muchos mecanismos de decoherencia diferentes. En general, dos parámetros pueden no ser adecuados para describir completamente la

decoherencia TLS, pero la experiencia (particularmente en RMN, ESR y experimentos de bombeo óptico) sugiere que T_1 y T_2 son a menudo suficientes para caracterizar la decoherencia TLS en muchas situaciones diversas y son por tanto parámetros TLS extremadamente importantes.

Primero discutimos cómo se introducen T_1 y T_2 para un TLS particular: un espín de electrón en un campo magnético externo. Ya hemos visto en la Ec. (9.29) como es el Hamiltoniano para un partícula de espín 1/2 (como es el electrón) en un campo externo $\vec{B}_0 = B_0\hat{z}$ a la que se aplica un campo $\vec{B}_1(t)$ de control que gira en el plano $\hat{x} - \hat{y}$ a una frecuencia ω_{rf} . Teniendo en cuenta las expresiones de S_z , S_y y S_z de (9.6), el Hamiltoniano en forma de matriz nos queda:

$$\mathcal{H} = -\omega_0 S_z - \omega_1 [\cos(\omega_{rf}t + \phi) S_x + \sin(\omega_{rf}t + \phi) S_y] = -\frac{\hbar}{2} \begin{bmatrix} \omega_0 & \omega_1 e^{-i(\omega_{rf}t + \phi)} \\ \omega_1 e^{i(\omega_{rf}t + \phi)} & -\omega_0 \end{bmatrix} \quad (10.2)$$

donde $\omega_0 = \gamma B_0$ y $\omega_1 = \gamma B_1$. La dinámica de espín bajo el Hamiltoniano (10.2) está gobernada por la **Ecuación de von Neumann** para la matriz densidad

$$i\hbar \frac{\partial \rho}{\partial t} = [\mathcal{H}, \rho] \quad (10.3)$$

Nota: Ecuación de von Neumann

Así como la ecuación de Schrödinger describe cómo evolucionan los estados puros en el tiempo, la ecuación de von Neumann (también conocida como ecuación de Liouville-von Neumann) describe cómo evoluciona un operador de densidad en el tiempo. Las dos ecuaciones son equivalentes, en el sentido de que cualquiera puede derivarse de la otra. La ecuación de von Neumann estable

$$i\hbar \frac{\partial \rho}{\partial t} = [\mathcal{H}, \rho] \quad (10.4)$$

donde los corchetes denotan el **comutador**, es decir,

$$[\mathcal{H}, \rho] = \mathcal{H}\rho - \rho\mathcal{H} \quad (10.5)$$

Como también vimos, podemos eliminar la dependencia temporal de \mathcal{H} yendo a un sistema de referencia rotante a una frecuencia ω_{rf} . En este sistema, el Hamiltoniano toma la forma de la Ec. (9.32). En forma matricial:

$$\mathcal{H}^{rot} = -(\omega_0 - \omega_{rf})S_z - \omega_1 [\cos(\phi)S_x + \sin(\phi)S_y] = -\frac{\hbar}{2} \begin{bmatrix} (\omega_0 - \omega_{rf}) & \omega_1 e^{-i\phi} \\ \omega_1 e^{i\phi} & -(\omega_0 - \omega_{rf}) \end{bmatrix} \quad (10.6)$$

Además, tenemos dos restricciones en la matriz densidad:

$$\rho = \begin{bmatrix} \rho_{\uparrow\uparrow} & \rho_{\uparrow\downarrow} \\ \rho_{\downarrow\uparrow} & \rho_{\downarrow\downarrow} \end{bmatrix} \Rightarrow \begin{aligned} \rho_{\uparrow\uparrow} + \rho_{\downarrow\downarrow} &= 1 \\ \rho_{\uparrow\downarrow} &= \rho_{\downarrow\uparrow}^* \end{aligned}$$

Con lo cual, solo nos interesan dos de las cuatro ecuaciones, una para $\rho_{\uparrow\uparrow}$ y otra para $\rho_{\uparrow\downarrow}$

$$\begin{aligned} i\hbar \frac{\partial \rho_{\uparrow\uparrow}}{\partial t} &= \frac{\hbar}{2} (\rho_{\uparrow\downarrow} e^{-i\phi} - \rho_{\downarrow\uparrow} e^{i\phi}) \\ i\hbar \frac{\partial \rho_{\uparrow\downarrow}}{\partial t} &= -\frac{\hbar}{2} [2\Delta\omega\rho_{\uparrow\downarrow} + \omega_1 e^{-i\phi}(\rho_{\uparrow\uparrow} - \rho_{\downarrow\downarrow})] \end{aligned} \quad (10.7)$$

donde $\Delta\omega = (\omega_0 - \omega_{rf})$.

Nota: Derivación de (10.7)

$$\begin{aligned}
\frac{2}{\hbar} [\mathcal{H}, \rho] &= - \begin{bmatrix} \Delta\omega & \omega_1 e^{-i\phi} \\ \omega_1 e^{i\phi} & -\Delta\omega \end{bmatrix} \begin{bmatrix} \rho_{\uparrow\uparrow} & \rho_{\uparrow\downarrow} \\ \rho_{\downarrow\uparrow} & \rho_{\downarrow\downarrow} \end{bmatrix} + \begin{bmatrix} \rho_{\uparrow\uparrow} & \rho_{\uparrow\downarrow} \\ \rho_{\downarrow\uparrow} & \rho_{\downarrow\downarrow} \end{bmatrix} \begin{bmatrix} \Delta\omega & \omega_1 e^{-i\phi} \\ \omega_1 e^{i\phi} & -\Delta\omega \end{bmatrix} \\
&= - \begin{bmatrix} \Delta\omega\rho_{\uparrow\uparrow} + \omega_1\rho_{\downarrow\uparrow}e^{-i\phi} & \Delta\omega\rho_{\uparrow\downarrow} + \omega_1\rho_{\downarrow\downarrow}e^{-i\phi} \\ \vdots & \vdots \end{bmatrix} + \begin{bmatrix} \Delta\omega\rho_{\uparrow\uparrow} + \omega_1\rho_{\uparrow\downarrow}e^{i\phi} & \omega_1\rho_{\downarrow\downarrow}e^{-i\omega} - \Delta\omega\rho_{\downarrow\downarrow} \\ \vdots & \vdots \end{bmatrix} \\
&= - \begin{bmatrix} \omega_1(\rho_{\downarrow\uparrow}e^{i\phi} - \rho_{\uparrow\downarrow}^*e^{-i\phi}) & 2\Delta\omega\rho_{\uparrow\downarrow} + \omega_1e^{-i\phi}(\rho_{\uparrow\uparrow} - \rho_{\downarrow\downarrow}) \\ \vdots & \vdots \end{bmatrix}
\end{aligned}$$

La evolución de este espín de electrón giratorio o en precesión es unitaria ya que hasta ahora estamos considerando un único espín aislado sin desfase ni decoherencia. La coherencia cuántica se mantiene siempre. Sin embargo, **el espín de un electrón nunca está aislado**. Se acopla a los grados de libertad orbitales del electrón a través del acoplamiento espín-órbita, a los espines nucleares a través de la interacción hiperfina y la interacción dipolar, a la red cristalina (y por tanto a los fonones) a través del acoplamiento espín-órbita, a cualquier impureza magnética del entorno a través del acoplamiento directo espín-dipolo, y a otros espines de electrones a través del acoplamiento dipolar y de intercambio.

Con todos estos grados de libertad “ambientales” presentes en principio, las simples ecuaciones libres de decoherencia para la matriz de densidad de espín (10.7) son obviamente una idealización. Para determinar cómo de fuertes son estas influencias “externas”, necesitamos incluirlas en el Hamiltoniano de partida, y luego emplear aproximaciones para lograr una comprensión cuantitativa de cómo evoluciona el sistema y cómo el espín que nos ocupa pierde su coherencia debido a su acoplamiento al “entorno”. Obviamente, se trata de un problema muy complicado (de hecho, un problema irresoluble, ya que nunca podremos estar seguros de conocer con precisión todos los grados de libertad posibles del entorno).

Un enfoque sencillo para abordar este problema consiste en añadir términos de decaimiento exponencial a los lados derechos de las dos Ecs. (10.7) anteriores:

$$\begin{aligned}
i\hbar \frac{\partial \rho_{\uparrow\uparrow}}{\partial t} &= \frac{\hbar}{2} (\rho_{\uparrow\downarrow}e^{-i\phi} - \rho_{\downarrow\uparrow}e^{i\phi}) - \frac{i\hbar}{T_1} \rho_{\uparrow\uparrow}, \\
i\hbar \frac{\partial \rho_{\uparrow\downarrow}}{\partial t} &= -\frac{\hbar}{2} [2\Delta\omega\rho_{\uparrow\downarrow} + \omega_1e^{-i\phi}(\rho_{\uparrow\uparrow} - \rho_{\downarrow\downarrow})] - \frac{i\hbar}{T_2} \rho_{\uparrow\downarrow},
\end{aligned} \tag{10.8}$$

para imitar el comportamiento observado fenomenológicamente de la decoherencia. Esto es similar a añadir un término de fricción proporcional a la velocidad en la ecuación de newton clásica. Las dos constantes pueden calcularse en determinadas condiciones si se dispone de suficiente información sobre el entorno. Este sencillo enfoque fenomenológico resulta ser bastante exitoso en la descripción de muchos experimentos, que van desde la RMN y la ESR hasta la óptica cuántica, aunque los cálculos explícitos reales de T_1 y T_2 suelen ser bastante difíciles. Quizás sea más fructífero considerar T_1 y T_2 como parámetros puramente fenomenológicos (que caracterizan las relajaciones longitudinal y transversal respectivamente en la dinámica TLS) que deben obtenerse a partir de medidas experimentales.

Recordemos que el vector unitario de magnetización está relacionado con el espín mediante

$$\vec{M} = \text{Tr}(\rho\vec{\sigma}) \tag{10.9}$$

con lo que

$$M_x = 2\text{Re}(\rho_{\uparrow\downarrow}), \quad M_y = -2\text{Im}(\rho_{\uparrow\downarrow}), \quad M_z = (\rho_{\uparrow\uparrow} - \rho_{\downarrow\downarrow}). \tag{10.10}$$

Podemos ahora reescribir las Eqs. (10.8) con respecto a las componentes del vector real \vec{M} , obteniendo las **ecuaciones de Bloch** en el sistema de referencia rotante:

$$\begin{aligned}\frac{\partial M_x}{\partial t} &= -\frac{1}{T_2} M_x - \Delta\omega M_y, \\ \frac{\partial M_y}{\partial t} &= \Delta\omega - \frac{1}{T_2} M_y - \omega_1 M_z, \\ \frac{\partial M_z}{\partial t} &= \omega_1 M_y - \frac{1}{T_1} (M_z + 1).\end{aligned}\quad (10.11)$$

Vemos fácilmente que T_2 es el **tiempo de relajación para la magnetización xy (transversal) de los electrones**, mientras que T_1 es el **tiempo de decaimiento para la magnetización en dirección z (longitudinal)**.

Para describir un conjunto de espines, que en general pueden poseer diferentes desdoblamientos Zeeman $\hbar\omega$ (por ejemplo, en virtud de inhomogeneidades en el campo magnético aplicado y/o en el factor g del electrón) y por lo tanto tener diferentes valor de $\Delta\omega$, es necesario realizar un promedio adicional del conjunto. Este promediado conduce a una constante de tiempo T_2^* ($\leq T_2$) diferente para describir la anchura de la señal de resonancia magnética (el ensanchamiento no homogéneo), pero no afecta a la dirección longitudinal. Obsérvese que T_2 (o T_2^*) describe el proceso de desfase (T_2 suele denominarse **tiempo de desfase**), y T_1 ($\geq T_2$) es el **tiempo de relajación inelástica de inversión de espín (spin-flip)**. A menudo T_2 también se denomina **tiempo de relajación espín-espín** por razones que se discutirán más adelante.

Las ecuaciones de Bloch describen con éxito fenómenos de desfase y relajación en átomos, óptica cuántica, espines nucleares y espines de electrones en semiconductores, aunque microscópicamente bien puede darse el caso de que sólo dos escalas de tiempo no sean suficientes para describir la dinámica.

En cuanto a la decoherencia de un solo espín, observamos que los procesos de inversión de espín (spin-flip) causan tanto la relajación de la población como el desfase, contribuyendo a ambas tasas $1/T_1$ y $1/T_2$. Sin embargo, en un sistema físico real las direcciones longitudinal y transversal suelen verse afectadas de forma diferente por el entorno. De hecho, existen procesos de desfase puros que afectan sólo a T_2 pero no a T_1 . Un ejemplo son las moléculas que colisionan en un medio gaseoso ópticamente activo, donde las moléculas sufren constantemente colisiones entre sí, algunas de ellas inelásticas, pero la mayoría elásticas.

Otro ejemplo bien conocido de desfase puro es la interacción dipolar espín-espín en RMN, que produce fluctuaciones efectivas del campo magnético local y, por tanto, contribuye esencialmente sólo a T_2 (el efecto correspondiente sobre T_1 es extremadamente pequeño). Lo que es importante para el desfase es que debe producirse algún cambio en el estado del entorno debido a su interacción con el sistema—el desfase no requiere necesariamente un proceso de dispersión inelástica explícito para el sistema, aunque todas las dispersiones inelásticas producen necesariamente desfase. De hecho, como se ha mencionado antes, T_2 en el contexto de la ESR y la RMN se denomina a menudo tiempo de relajación espín-espín porque el efecto intrínseco más importante que contribuye a $1/T_2$ es la interacción dipolar entre varios espines del sistema, que, aunque transfiere energía entre los propios espines, no conduce a una relajación energética global del sistema de espín total. Por el contrario, las interacciones espín-red conducen a la relajación de la energía (a través de procesos de spin-flip) desde el sistema de espín a la red, y por lo tanto contribuyen a $1/T_1$, la tasa de relajación espín-red. En este contexto, observamos que T_2 establece la escala de tiempo para que el sistema de espín alcance el equilibrio dentro de sí mismo, mientras que T_1 establece la escala de tiempo para el equilibrio termodinámico global entre el sistema de espín y la red.

En resumen: **todos los procesos inelásticos que contribuyen a T_1 también conducen automáticamente al desfase, pero en muchas circunstancias puede haber procesos de des-**

fase adicionales (por ejemplo, el acoplamiento dipolar espín-espín en RMN y ESR) que contribuyen sólo a T_2 (y no a T_1), y por lo tanto $\mathbf{T}_1 \geq \mathbf{T}_2$ en general.

10.3. Resumen

En resumen, la decoherencia puede ser de dos tipos:

- Perdida de la fase relativa entre los estados ($\rightarrow T_2$)
- Decaimiento de las amplitudes ($\rightarrow T_1$)

Las decoherencia proviene de todas aquellas interacciones que no tenemos en cuenta el Hamiltoniano del sistema, debido a su complejidad (o imposibilidad) de tratamiento. Estas interacciones pueden ser entre los propios elementos del sistema o interacciones con el entorno. Para que la computación cuántica tenga éxito, los tiempos de decoherencia deben de ser mucho mayores que los tiempos de aplicación de las puertas cuánticas.

Sorprendentemente, en muchos TLS solo necesitamos dos parámetros para describir la decoherencia: T_1 y T_2 . El primero da cuenta del tiempo de relajación longitudinal y el segundo del transversal. Es decir, el primero nos dice el tiempo que se mantiene la amplitud relativa entre estados y el segundo el tiempo que se mantiene la fase relativa entre estados. Estos parámetros son puramente fenomenológicos.

Parte III

Algoritmos Cuánticos

Capítulo 11

Elementos básicos de los algoritmos cuánticos

En este capítulo vamos a ver una serie de conceptos importantes a la hora de construir los circuitos cuánticos.

11.1. Circuitos

Hasta ahora hemos visto varias puestas mono-qúbit y multi-qúbit, así como algunos circuitos simples. Antes de implementar algoritmos cuánticos en ordenadores cuánticos reales, es importante destacar la definición concreta de **circuito cuántico**, ya que construiremos circuitos cuánticos para implementar estos algoritmos.

11.1.1. Qué es un circuito cuántico?

Un **circuito cuántico** es una rutina computacional consistente en *operaciones cuánticas coherentes sobre datos cuánticos, como qúbits, y computación clásica concurrente en tiempo real. Se trata de una secuencia ordenada de puertas cuánticas, mediciones y reinicios, todos los cuales pueden estar condicionados y utilizar datos del cálculo clásico en tiempo real.*

Se dice que un conjunto de puertas cuánticas es **universal** si cualquier transformación unitaria de los datos cuánticos puede aproximarse de forma eficiente y arbitraria como una secuencia de puertas del conjunto. Cualquier programa cuántico puede representarse mediante una secuencia de circuitos cuánticos y computación clásica no concurrente.

Nota: ordenación de qúbits

Recordemos que la ordenación de los qúbits en el circuito tiene un convenio estándar que casi todo el mundo sigue. Sin embargo, uno de los principales agentes en este medio, IBM, usa un convenio distinto en su software Qiskit

Convenio estandar	Qiskit
$ a_{n-1}\rangle$	$ a_0\rangle$
$ a_{n-2}\rangle$	$ a_1\rangle$
\vdots	\vdots
$ a_0\rangle$	$ a_{n-1}\rangle$

Figura 11.1: Convenios de ordenación de los qubits en la forma estándar, resaltando que Qiskit decide usar el convenio al revés.

11.1.2. Ejemplo: Circuito de teleportación.

Vamos a ver en esta sección a modo de ejemplo un circuito más completo que el de la Fig. 8.3 para implementar la teleportación. Con lo de “más completo” me refiero a que el circuito de la Fig. 8.3 no presenta la inicialización de los estados, además de que simplifica el hecho de usar bit clásicos poniéndolos en medio del circuito. El circuito sería el de la Fig. 11.2.

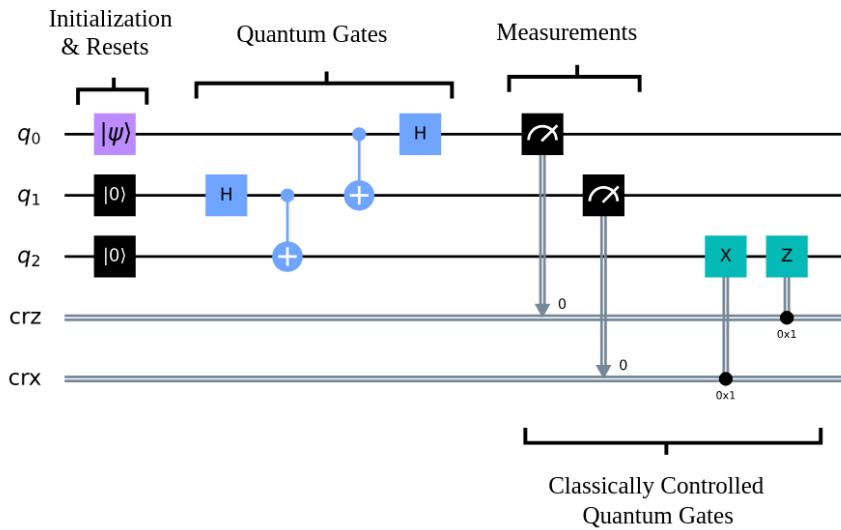


Figura 11.2: Circuito de teleportación completo.

El circuito cuántico utiliza tres qubits y dos bits clásicos. Hay cuatro componentes principales en este circuito cuántico.

- **Inicialización y reinicio:** En primer lugar, necesitamos comenzar nuestro cálculo cuántico con un estado cuántico bien definido. Esto se consigue mediante las operaciones de inicialización y reinicio. Los reinicios pueden realizarse mediante una combinación de puertas de un solo qubit y computación clásica concurrente en tiempo real que controla si hemos creado con éxito el estado deseado mediante mediciones.
- **Puertas cuánticas:** En segundo lugar, aplicamos una secuencia de puertas cuánticas que manipulan los tres qubits tal y como requiere el algoritmo de teleportación. En este caso, sólo necesitamos aplicar puertas Hadamard (H) de un qubit y CNOTs (dos qubits).
- **Mediciones:** En tercer lugar, medimos dos de los tres qubits. Un ordenador clásico interpreta

las medidas de cada qúbit como resultados clásicos (0 y 1) y los almacena en los dos bits clásicos.

- **Compuertas cuánticas controladas clásicamente:** En cuarto lugar, aplicamos puertas cuánticas y de un solo qúbit al tercer qubit. Estas puertas están condicionadas a los resultados de las medidas que se almacenan en los dos bits clásicos. En este caso, utilizamos los resultados del cálculo clásico simultáneamente en tiempo real dentro del mismo circuito cuántico.

Porque partes clásicas en los circuitos?

Aunque un ordenador cuántico universal puede hacer cualquier cosa que haga un ordenador clásico, a menudo añadimos partes clásicas a nuestros circuitos cuánticos porque los estados cuánticos son frágiles.

Cuando medimos el qúbit, colapsamos su estado y destruimos gran parte de la información. Como lo único que hace la medición es destruir información, en teoría podemos medir siempre en último lugar y no perder ninguna ventaja computacional. En realidad, medir antes ofrece muchas ventajas prácticas.

Por ejemplo, en el circuito de teleportación, medimos los qúbits para poder enviar la información por canales clásicos en lugar de por canales cuánticos. La ventaja es que los canales clásicos son muy estables, mientras que en realidad no tenemos forma de enviar información cuántica a otras personas, ya que los canales son muy difíciles de crear.

Un ejemplo de mezcla de computación clásica y cuántica son, como ya veremos, los famosos algoritmo **VQE (variational quantum eigensolvers)**. En estos algoritmos se itera en bucle, donde se hace un cálculo en ordenador cuántico con un circuito paramétrico, después se optimizan estos parámetro con un ordenador clásico, para volver a hacer el calculo en el ordenador cuántico, y así sucesivamente. Dividir el cálculo en cálculos cuánticos más pequeños nos hace perder cierta ventaja computacional, pero lo compensa el hecho de que tenemos hardware ruidoso y al hacer esto reducimos el tiempo que nuestros qúbits están en superposición. Esto significa que hay menos posibilidades de que las interferencias introduzcan imprecisiones en nuestros resultados.

Por último, para utilizar los resultados de nuestro cálculo cuántico en el mundo clásico cotidiano, tenemos que medir e interpretar esos estados al final del cálculo.

11.2. Retroceso de fase (Phase kickback)

Hemos estudiado ya el operador controlado CU . Es un error frecuente pensar que el qúbit controlador no se modifica. Un caso importante ocurre cuando el operador U actúa sobre uno de sus autoestados (recuerda que los autovalores de un operador unitario son fases puras)

$$U |u\rangle = e^{i\lambda} |u\rangle \quad (11.1)$$

Supongamos que por el **qúbit controlador** circula una superposición $(a|0\rangle + b|1\rangle)$ y por el **qúbit controlado** un autoestado $|u\rangle$ de U . La acción de CU es

$$CU : (a|0\rangle + b|1\rangle) \otimes |u\rangle \rightarrow a|0\rangle |u\rangle + b|1\rangle e^{i\lambda} |u\rangle = (a|0\rangle |u\rangle + b e^{i\lambda} |1\rangle) \otimes |u\rangle \quad (11.2)$$

En resultado final es que la fase $e^{i\lambda}$ ha **modificado** el estado del qúbit controlador, mientras que el segundo qúbit no ha cambiado. Podemos ver esto en la Fig. 11.3

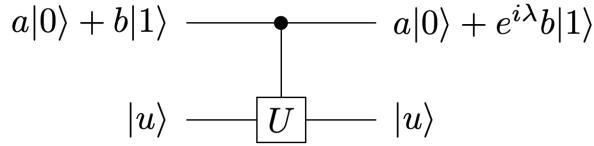


Figura 11.3: Ejemplo de *phase kickback*.

El punto es que, en el segundo paso, la fase generada por la acción de U **no pertenece** realmente a ninguno de los dos espacios sino al producto. De modo que puede adscribirse al primer espacio, como hemos hecho en el último paso. De ahí el nombre de **retroceso de fase**, en inglés **phase kickback**.

Ejercicio 41 Programa un circuito en el que $U = P(\phi)$ es el operador de fase y el estado en el primer cúbít es $|0\rangle$ y en el segundo es $|1\rangle$.

- Usando Qiskit representa el estado de salida para distintos de valores de $\phi \in [0, 2\pi)$
- ¿En qué plano rota el vector del primer qúbit? ¿Cómo podemos cambiar dicho plano de rotación?

11.3. Circuitos equivalentes

Es posible encontrar distintos circuitos que producen acciones idénticas sobre un estado arbitrario. Se denominan **circuitos equivalentes**. Matemáticamente, representan distintas descomposiciones del mismo operador unitario en unitarios. Esto es muy importante, pues nos permite construir unas puertas a partir de otras, con lo cual, como ya veremos, implica que *no es necesario tener implementadas todas las puertas*.

11.3.1. Conjugación

Una caso muy frecuente es la **conjugación** de una puerta V por un **unitario** U

$$UVU^\dagger = V' \quad (11.3)$$

El operador conjugado V' tiene la **misma acción** sobre la base **base rotada** $\{|e'\rangle = U|e\rangle\}$ que la que tiene V sobre la original $\{|e\rangle\}$. Por ello

- Los **autovalores** de V y de V' son los **mismos**
- Los **autovectores** son los **rotados**

esto es

$$\lambda' = \lambda \quad \text{y} \quad |\lambda'\rangle = U|\lambda\rangle \quad (11.4)$$

Nota: Comprobémoslo

$$\begin{aligned} V|\lambda\rangle = \lambda|\lambda\rangle &\Rightarrow V'|\lambda'\rangle = (UVU^\dagger)U|\lambda\rangle = UV|\lambda\rangle = U(\lambda|\lambda\rangle) = \lambda U|\lambda\rangle = \lambda|\lambda'\rangle \\ &\Rightarrow V'|\lambda'\rangle = \lambda|\lambda'\rangle \end{aligned}$$

Por ejemplo, en la Fig. 11.4 podemos ver la equivalencia entre Z y X conjugando con H .



Figura 11.4: Relación de conjugación entre Z y X .

Para entenderla, tenemos que recordar que la puerta H lo que hace es girar 180° entorno a un eje que está a 45° grados entre el eje z y el eje x . Es decir, lleva el eje z al eje x (lleva $|0\rangle$ a $|+\rangle$ y $|1\rangle$ a $|-\rangle$). Aquí vemos claramente que la acción de Z sobre la base sin rotar ($\{|0\rangle, |1\rangle\}$) es la misma que la acción de X sobre la base rotada ($\{|+\rangle, |-\rangle\}$). Este tipo de equivalencia se sigue de las identidades algebraicas

$$HXH = Z \quad , \quad HZH = X$$

Análogamente

$$SXS^\dagger = Y \quad , \quad SYS^\dagger = -X \quad , \quad SZS^\dagger = Z.$$

también son fáciles de visualizar recordando que $S = \sqrt{Z}$ es una rotación de 90° en torno al eje Z .

11.3.2. Conjugación de una exponencial

Muchas puertas unitarias son de la forma $V = e^{\alpha A}$. Por ejemplo, si $A = \hat{\mathbf{n}} \cdot \boldsymbol{\sigma}$ entonces $\Rightarrow V = R_{\hat{\mathbf{n}}}(\theta)$ es el operador que rota un ángulo θ en torno al vector $\hat{\mathbf{n}}$ en la esfera de Bloch de la Ec. (4.1).

Lemma 5 *La conjugación de una exponencial se exponencia*

Demostración:

$$Ue^{\alpha A}U^\dagger = U \left(1 + \alpha A + \frac{1}{2}\alpha^2 A^2 + \dots \right) U^\dagger \quad (11.5)$$

$$= 1 + \alpha UAU^\dagger + \frac{1}{2}\alpha^2 UAU^\dagger UAU^\dagger + \dots \quad (11.6)$$

$$= e^{\alpha UAU^\dagger} \quad (11.7)$$

■

Para el caso $A = \hat{\mathbf{n}} \cdot \boldsymbol{\sigma}$ la conjugación es

$$UAU^\dagger \rightarrow U(\hat{\mathbf{n}} \cdot \boldsymbol{\sigma})U^\dagger = (U\hat{\mathbf{n}}) \cdot \boldsymbol{\sigma} \quad (11.8)$$

Ejercicio 42 Comprueba la Ec. 11.8.

Coloralio 1 La conjugación de una rotación en torno a un eje produce una rotación en torno al eje conjugado

$$UR_{\hat{\mathbf{n}}}(\theta)U^\dagger = R_{U\hat{\mathbf{n}}}(\theta) \quad (11.9)$$

Ejemplo

Por ejemplo:

$$\begin{aligned} HR_z(\theta)H &= e^{-i\theta/2HZH} = e^{-i(\theta/2)X} \\ &= R_x(\theta) \end{aligned}$$



Figura 11.5: La conjugación con H de una rotación en Z nos da una rotación en X .

11.3.3. Varios qubits

11.3.3.1. Cambiar controlador y controlado en CZ y CP

La puerta controlada $CZ = \text{diag}(1, 1, 1, -1)$ es simétrica ya que lo único que hace, es cambiar de signo al estado $|11\rangle$. Podemos ver esto en la Fig. 11.6.

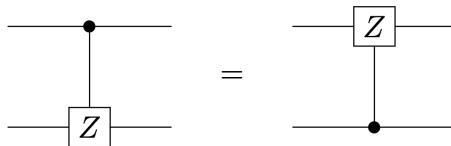


Figura 11.6: Equivalencia entre CZs cambiando el qúbit controlador y el controlado.

En realidad, CZ es un caso particular de $CP(\phi) = \text{diag}(1, 1, 1, e^{i\phi})$, para la cual la equivalencia es la misma.

11.3.3.2. Cambiar controlador y controlado en la CNOT

Otra equivalencia importante es la de la Fig. 11.7. Para probarla, observemos que las tres puertas del segundo qúbit se pueden componer para dar $H X H = Z$. Por el contrario, las dos puertas de Hadamard en el primer qúbit no se pueden multiplicar al haber un control entre ellas. Sin embargo, usando la equivalencia de la Fig. 11.6 podemos invertir la puerta CZ y, finalmente, conjugar en el primer qúbit $H Z H = X$.

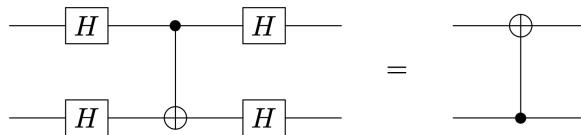


Figura 11.7: Equivalencia entre CNOTs cambiando el qúbit controlador y el controlado

11.3.3.3. SWAP a partir de CNOTs

Otra equivalencia nada intuitiva pero muy importante relaciona tres operaciones CNOT con la permutación U_{SWAP} de la Fig. 11.8. No hay una forma sencilla de probar esta identidad, así que lo recomendable es escribir las matrices asociadas a cada miembro y comprobar que son iguales. Vemos que gracias a esta equivalencia, la SWAP no es una puerta fundamental.

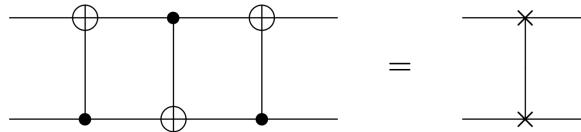


Figura 11.8: Construcción de la puerta SWAP mediante CNOTs.

Jupyter Notebook: 7. Elementos básicos de los algoritmos cuánticos

Ver la sección 7.1. SWAP a partir de CNOTs del notebook 7. Elementos básicos de los algoritmos cuánticos.

El Notebook puede descargarse de [Github](#).

11.3.3.4. CK (puerta de phase global controlada) y P_ϕ

La puerta de phase global controlada, $CK_\phi = \text{diag}(1, 1, e^{i\phi}, e^{i\phi})$, secretamente, no es una puerta controlada

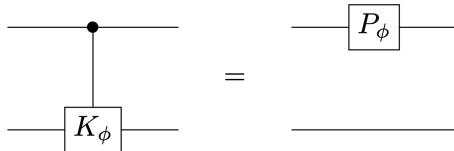


Figura 11.9: La puerta controlada de fase global no es una puerta controlada.

Ejercicio 43 Demuestra las equivalencias de circuitos anteriores de dos formas:

- sobre el papel, multiplicando las matrices asociadas
- en qiskit, componiendo los circuitos y extrayendo el operador unitario asociado.

Ejercicio 44 Comprueba la equivalencia de los dos circuitos siguientes, siempre que se verifique que $V^2 = U$

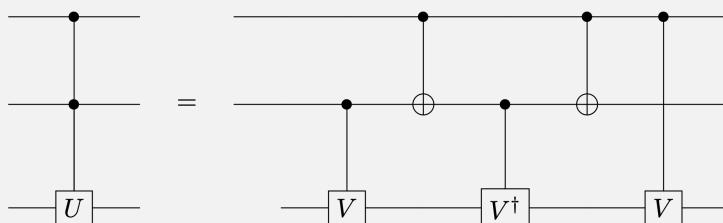


Figura 11.10: Descomposición de una puerta con dos controles

Pista: puedes demostrarlo viendo que operaciones hacen ambos circuitos sobre el tercer qubit cuando por los dos primeros entran los 4 estados posibles ($|00\rangle$, $|10\rangle$, $|01\rangle$ y $|11\rangle$)

11.4. Operadores de Clifford

Definición 26 Se define un **operador de Clifford**, U , como aquel que conjuga un operador de Pauli a otro operador de Pauli.

Los propios operadores de Pauli son operadores de Clifford. La conjugación correspondiente simplemente refleja el operador de Pauli. Por ejemplo con $U = Z$

$$ZXZ = -X \quad ZYZ = -Y \quad ZZZ = Z \quad (11.10)$$

Pero vemos que también H y S son de Clifford. Por el contrario T no es un operador de Clifford.

Nota: simulaciones de circuitos con operadores de Clifford

Los circuitos cuánticos compuestos únicamente por operadores de Clifford pueden simularse de forma eficiente en ordenadores clásicos gracias al **teorema de Gottesman-Knill** (ver [20]).

Esta definición se extiende a puertas multi-cúbits. Un operador de Clifford será aquel que conjuga una cadena de Pauli para dar otra cadena de Pauli. Por ejemplo

$$\begin{aligned} (XXH)(YZZ)(XXH)^\dagger &= XXH \otimes YZZ \otimes XXH \\ &= XYX \otimes XZX \otimes HZH \\ &= (-Y) \otimes (-Z) \otimes Z \\ &= YZZ \end{aligned}$$

También podemos conjugar operadores obtenidos por exponentiación

$$(XXH)e^{aYZZ}(XXH)^\dagger = e^{aXXH \otimes XZX \otimes HZH} = e^{aYZZ} \quad (11.11)$$

Para 2 qubits la **clase de Clifford** admite puertas controladas.

$$CX(X \otimes I)CX = X \otimes X \quad (11.12)$$

que copia el operador X en el segundo qubit. Esta identidad se puede demostrar gráficamente como vemos en la Fig. 11.11.

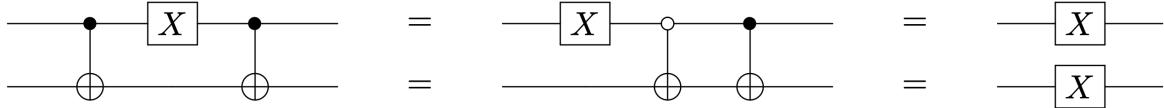


Figura 11.11: Clonación de la puerta X .

11.5. Universalidad de la computación cuántica con puertas.

El objetivo de un **computador cuántico universal** es el de ser capaz de implementar el operador unitario más general

$$U = \sum_x |f(x)\rangle \langle x| \quad (11.13)$$

donde $f : x \rightarrow f(x)$ es una función arbitraria invertible.

11.5.1. Teorema

Teorema 25 (Barenco et. al. 1995) *Cualquier operador unitario U_n sobre n qubits puede expresarse como el producto de*

- *puertas continuas de un qubit*
- *puertas CNOT*

Podemos ver un ejemplo de este teorema en la Fig. 11.12.

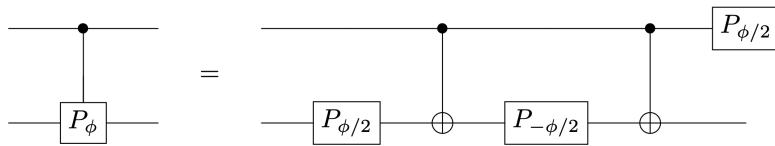


Figura 11.12: Descomposición de la puerta CP_ϕ en puertas de un qubit y CNOTs

Demostración: No vamos a ver en detalle la demostración pero su vamos a ilustrar cuales son los pasos a seguir:

1. Cualquier operador U_n sobre n cúbites se puede descomponer como **producto de operadores C^kU** controlados por k cúbites.
2. Los operadores C^kU se pueden descomponer como productos de un operador CU y puertas de Toffoli (CCNOT). En general para C^kU necesitamos $k - 1$ ancillas, como podemos ver en la Fig. 11.13.

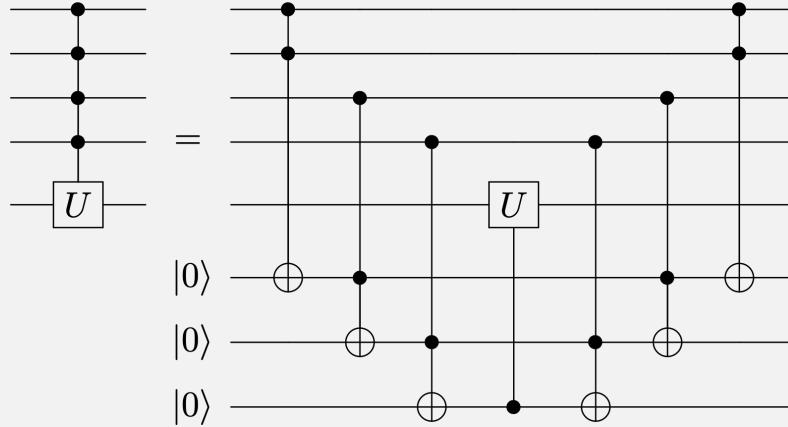


Figura 11.13: Descomposición de una puerta U multicontrolada.

3. Las puertas de Toffoli puede descomponerse como productos de H , CX y CS , como podemos ver en la Fig. 11.14. Este no es más que el caso particular de la descomposición general de C^2U de la Fig. 11.10 usando $U = X = HZH = HSSH = (HSH)(HSH) = V^2 \Rightarrow V = HSH$.

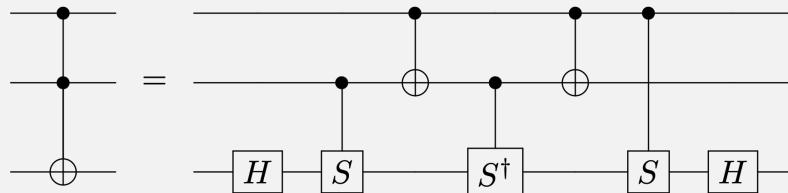


Figura 11.14: Descomposición de la puerta Toffoli.

4. Una puerta CU puede descomponerse de forma única usando tres rotaciones A, B y C que verifiquen

$$ABC = I \quad , \quad e^{i\delta} AXBXC = U, \quad (11.14)$$

siguiendo el circuito de la Fig. 11.15

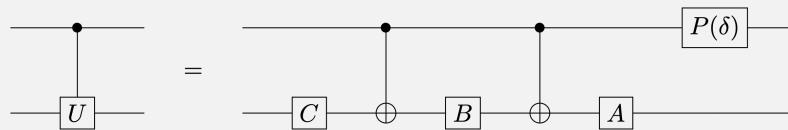


Figura 11.15: Descomposición de CU .

En efecto, si el qubit de control es $|0\rangle$ la fase $P(\delta)$ no le afecta y el operador efectivo en el segundo qubit es $ABC = I$. Por el contrario, si el primer qubits es $|1\rangle$, entonces se aplica $AXBXC$ al segundo qubit, y el operador $P(\delta)$ añade la fase global, que al ser global podemos pasársela al segundo qubit, lo que hace $e^{i\delta} AXBXC = U$.

Nota: detalles sobre la descomposición de CU

Las dos condiciones algebraicas admiten una solución única para un operador unitario genérico

$$U = e^{i\delta} \begin{bmatrix} e^{-i(\alpha+\beta)/2} \cos \frac{\theta}{2} & -e^{i(-\alpha+\beta)/2} \sin \frac{\theta}{2} \\ e^{i(\alpha-\beta)/2} \sin \frac{\theta}{2} & e^{i(\alpha+\beta)/2} \cos \frac{\theta}{2} \end{bmatrix} = e^{i\delta} AXBXC \implies$$

$$\implies \begin{cases} A = R_z(\alpha)R_y\left(\frac{\theta}{2}\right) \\ B = R_y\left(-\frac{\theta}{2}\right)R_z\left(-\frac{\alpha+\beta}{2}\right) \\ C = R_z\left(\frac{\beta-\alpha}{2}\right) \end{cases}$$

Como $U \in U(2)$ es un operador unitario y su determinante es una fase $\det U = e^{i\delta}$. Por su parte $\det(AXBXC) = 1$, y por tanto es un elemento de $SU(2)$. Por esta razón es necesario añadir la fase $e^{i\delta}$ para obtener un operador unitario general.

■

11.5.2. Conjuntos de puertas universales

La descomposición del teorema de Barenco es una identidad exacta capaz de descomponer un conjunto infinito y continuo de operadores U_n en puertas CNOT y puertas *continuas* U . Por otro lado, esto coincide con la demanda de la computación cuántica **resistente a errores (fault tolerant)** de una discretización del proceso de computación. De esta forma, lo que se busca es un *conjunto discreto de puertas universales* susceptibles de ser implementadas de manera resistente a errores.

Este conjunto de puertas universales puede ser diferente dependiendo de la plataforma utilizada. Por ejemplo, circuitos superconductores se usa, entre otras, la base

- Las puertas de Clifford: $\{H, S, \text{CNOT}\} + T$

Es importante relacionar las puertas universales con **puertas nativas** (las que de verdad se implementan y se usan para construir las demás). Por ejemplo, en ordenadores de iones atrapados hay las siguientes puertas nativas [21]:

$$\{\text{GPI}, \text{VirtualZ}, \text{MS}\} \quad (11.15)$$

donde

$$\text{GPI} = \begin{bmatrix} 0 & e^{-i\phi} \\ e^{i\phi} & 0 \end{bmatrix}, \quad \text{VirtualZ} = \begin{bmatrix} e^{-i\phi} & 0 \\ 0 & e^{i\phi} \end{bmatrix}, \quad (11.16)$$

y la puerta de Mølmer-Sørensen

$$\text{MS}(\phi_1, \phi_2) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & e^{-i(\phi_1+\phi_2)} \\ 0 & 1 & -ie^{-i(\phi_1-\phi_2)} & 0 \\ 0 & -ie^{i(\phi_1-\phi_2)} & 1 & 0 \\ -ie^{i(\phi_1+\phi_2)} & 0 & 0 & 1 \end{bmatrix} \quad (11.17)$$

11.6. Medidas de calidad de un Circuitos.

Algunas medidas cuantitativas permiten comparar la calidad de distintos circuitos que efectúan la misma tarea.

- **Anchura:** es el número total de qubits que necesita. El uso de ancillas incrementa la anchura de un circuito, y por tanto, reduce su calidad en comparación con otro circuito que tenga menor anchura.

- **Coste:** Número de puertas presente
- **Complejidad:** es una medida estandarizada asociada al número de **puertas elementales** en las que puede descomponerse un circuito. Es un número a reducir. Sin embargo no es una medida inambigua ya que depende de la librería utilizada. Por ejemplo, si ésta es la NCT, entonces el coste del circuito de la Fig. 11.14 es 1. Sin embargo, si la librería es la tomada por $\{H, S, T, \text{CNOT}\}$ entonces el coste sube hasta 7. Por ello, a la hora de comparar circuitos es importante definirlos en la misma base.
- **Profundidad:** para evaluar la **profundidad** es necesario agrupar todas las puertas que se puedan realizar en paralelo en cortes temporales de duración Δ (pulso). En particular puertas que actúen sobre registros diferentes no interferirán y se podrán paralelizar. Por ejemplo, el circuito de la Fig. 11.13 tiene un coste igual a 6, pero una profundidad igual a 5.

Jupyter Notebook: [7. Elementos básicos de los algoritmos cuánticos](#)

Ver la sección [7.2. Medidas de calidad de un circuito](#) del notebook [7. Elementos básicos de los algoritmos cuánticos](#).

El Notebook puede descargarse de [Github](#).

Ejercicio 45 *Competa los siguientes apartados:*

1. Programa y ejecuta un circuito de 16 cùbits que prepare el estado $|00\dots0\rangle \rightarrow \frac{1}{\sqrt{2}}(|00\dots0\rangle + |11\dots1\rangle)$
2. Modifica la posición de los controladores hasta reducir la profundidad a 5.

Capítulo 12

Estado inicial y oráculos

12.1. Preparación de un estado inicial genérico

En general, la preparación de un estado inicial arbitrario (así como el borrado de un estado) no es una operación, en general, eficiente. Esto es porque, en general, esta preparación implica fijar 2^n números complejos

$$U : |0\rangle \rightarrow \sum_{i=0}^{2^n-1} c_i |i\rangle \quad (12.1)$$

Como podemos ver, la complejidad no es polinómica, sino exponencial.

Veamos uno de los algoritmos más simples (los hay mas refinados) para preparar un estado inicial arbitrario. Separemos las amplitudes complejas en módulo y fase

$$c_i = a_i e^{\gamma_i}, \quad \text{donde } a_i = |c_i| \quad (12.2)$$

Veamos el caso $n = 2$. El circuito que nos permite preparar un estado genérico es el de la Fig. 12.1, donde

$$R_y(\theta) = \begin{bmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix}, \quad D(\gamma_i, \gamma_j) = \begin{bmatrix} e^{i\gamma_i} & 0 \\ 0 & e^{i\gamma_j} \end{bmatrix} = K(\gamma_i)P(\gamma_j - \gamma_i) \quad (12.3)$$

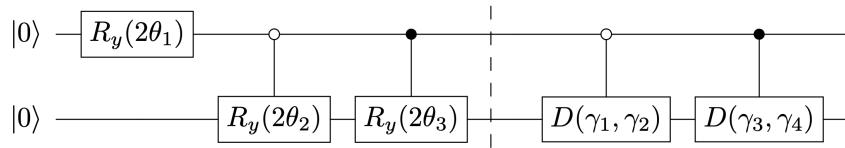


Figura 12.1: Preparación de un estado inicial genérico de 2 qubits.

Analicémoslo parte a parte. La primera parte fijará los módulos y la segunda las fases. El estado antes de la barrera será

$$\begin{aligned} |\psi_0\rangle &= \cos \theta_1 |0\rangle \otimes (\cos \theta_2 |0\rangle + \sin \theta_2 |1\rangle) + \sin \theta_1 |1\rangle \otimes (\cos \theta_3 |0\rangle + \sin \theta_3 |1\rangle) \\ &= \cos \theta_1 \cos \theta_2 |00\rangle + \cos \theta_1 \sin \theta_2 |01\rangle + \sin \theta_1 \cos \theta_3 |10\rangle + \sin \theta_1 \sin \theta_3 |11\rangle \end{aligned}$$

donde obtenemos 4 ecuaciones para 4 incógnitas

$$\begin{aligned} a_1 &= \cos \theta_1 \cos \theta_2 \\ a_2 &= \cos \theta_1 \sin \theta_2 \\ a_3 &= \sin \theta_1 \cos \theta_3 \\ a_4 &= \sqrt{1 - a_1^2 - a_2^2 - a_3^2} \end{aligned}$$

sólo necesitamos 3 ángulos para representar 4 amplitudes debido a la ligadura $\sum_i a_i^2 = 1$. Una vez fijadas las amplitudes, la última parte del circuito es equivalente al operador unitario

$$U = \begin{bmatrix} e^{i\gamma_1} & 0 & 0 & 0 \\ 0 & e^{i\gamma_2} & 0 & 0 \\ 0 & 0 & e^{i\gamma_3} & 0 \\ 0 & 0 & 0 & e^{i\gamma_4} \end{bmatrix} = \begin{bmatrix} K(\gamma_1)P(\gamma_2 - \gamma_1) & 0 \\ 0 & K(\gamma_3)P(\gamma_4 - \gamma_3) \end{bmatrix} = |0\rangle\langle 0| D(\gamma_1, \gamma_2) + |1\rangle\langle 1| D(\gamma_3, \gamma_4) \quad (12.4)$$

Evidentemente este circuito no puede ser eficiente puesto que es necesario ajustar un número $2 \cdot 2^n - 1$ de parámetros (el número de puertas crece exponencialmente)

Sin embargo, no todo está perdido. Como ya veremos, muchos algoritmos aprovechan que ciertos estados son más fáciles de preparar que otros. Por ejemplo el estado inicial que es una superposición homogénea de elementos de la base

$$|\psi\rangle = W|0\rangle = |+\rangle^{\otimes n} = \frac{1}{\sqrt{n}} \sum_{i=1}^{2^n-1} |i\rangle \quad (12.5)$$

se obtiene con un circuito de **coste** = n y **profundidad**=1 (aplicando una puerta de Hadamard a cada qubit). Este caso es el ejemplo de superposición perfecta. De una tacada estamos fijando los 2^n números complejos.

Muchas veces, como en química computacional, ya no se trata de preparar un estado arbitrario, sino que el estado inicial debe de cumplir ciertas condiciones que hacen que sea más fácil construir estos estados que construir estados arbitrarios.

12.2. Oráculos (funciones digitales)

Una clase de problemas en los que la computación cuántica promete alcanzar una ventaja con respecto a la clásica se denominan **algoritmos de consulta del oráculo (oracle query)**. En estos problemas tenemos un **oráculo**, que podemos verlo como una función a la cual nosotros le damos inputs y el oráculo nos da outputs dependiendo del input. El oráculo es como una **caja negra**. El objetivo de estos algoritmos es extraer información del oráculo haciendo consultas (dándole inputs). Como el espacio de funciones sobre el espacio de Hilbert es mucho más grande que el propio espacio de Hilbert, el hecho de adivinar una propiedad de una función es un problema de complejidad NP.

Nota: Origen del nombre “oráculo”

Precisamente, el nombre de oráculo viene de los oráculos como personas a las que se les iba a preguntar por el futuro y ellos te daban respuestas, sin saber muy bien de donde salían las respuestas.

Aun que consideremos los oráculos como cajas negras, lo cierto es que tenemos que construirlo, así que sabemos como están construidos. Sin embargo, el razonamiento es el mismo. Esto se entenderá mucho mejor cuando veamos, por ejemplo, el algoritmo de Grover.

12.2.1. Construcción de funciones binarias. Los min-términos

Una **función binaria (o digitales)** no es más que una función que lleva cadenas de n bits de entrada a cadenas de m bits como salida

$$f : \{0, 1\}^n \rightarrow \{0, 1\}^m \quad (12.6)$$

La construcción de f es equivalente a la especificación de m funciones f_1, f_2, \dots, f_m **binarias**

$$f_i : \{0, 1\}^n \rightarrow \{0, 1\} \quad (12.7)$$

Es evidente que ninguna función binaria es invertible para $n \geq 2$. Como la computación cuántica es reversible, tenemos que buscar otra forma de implementar estas funciones. La manera más simple de fabricar, a partir de un mapa no invertible f , otro invertible U_f , implica *conservar* los valores de la variables iniciales. Es decir, para $f : \{0, 1\}^n \rightarrow \{0, 1\}$ necesitamos un total de $n + 1$ qubits:

- n cúbites que contienen el argumento de la función, $|x\rangle_n \in \mathbb{C}^n$,
- 1 cúbite que guardará el resultado, $|y\rangle \in \mathbb{C}$.

Sea U_f el siguiente operador

$$U_f : |x\rangle |y\rangle \longrightarrow |x\rangle |y \oplus f(x)\rangle \quad (12.8)$$

Donde \oplus indica suma módulo 2. Es evidente de la definición que $U_f \cdot U_f = I$, así que es invertible.

Es muy sencillo establecer un método general para construir funciones binarias de la forma $f : \{0, 1\}^n \rightarrow \{0, 1\}$. Consideremos la siguiente **tabla de verdad** para una función $f : \{0, 1\}^3 \rightarrow \{0, 1\}$ concreta.

x_2	x_1	x_0	$f(x)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Tabla 12.1: Ejemplo de función binaria $f : \{0, 1\}^3 \rightarrow \{0, 1\}$

La idea es considerar exclusivamente los términos que tienen como salida la variable 1, que denominaremos **min-términos**. Por ejemplo hay un min-término de la forma $101 \rightarrow 1$ que se puede obtener mediante una puerta de la Fig. 12.2.

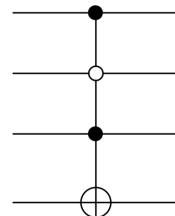


Figura 12.2: Uno de los min-términos de la función de la tabla 12.1

Ejercicio 46 Completa el código del notebook para implementar la función $f : \{0, 1\}^4 \rightarrow \{0, 1\}^4$ dada su tabla de verdad. Recuerda que en qiskit el bit menos significativo es el de arriba.

Ejercicio 47 Escribe una función $f : S^n \rightarrow S$ que produzca aleatoriamente $f(x) = \pm 1$ de forma equilibrada (es decir, tantos $f(x) = +1$ como $f(x) = -1$). Puedes ver la solución en la sección 4.4 de algoritmo de [Deutsch-Jozsa del notebook de Qiskit](#).

12.2.1.1. Función binaria lineal.

Dados dos n-tuplas binarias $x = (x_{n-1}, \dots, x_0)$ y $a = (a_{n-1}, \dots, a_0)$ definimos la **función lineal**

$$f(x; a) = a \cdot x = a_{n-1}x_{n-1} \oplus a_{n-2}x_{n-2} \oplus \dots \oplus a_0x_0, \quad (12.9)$$

donde \oplus es la suma módulo 2. Por ejemplo, el circuito que implementa esta función cuando $a = 11010$ es el de la Fig. 12.3

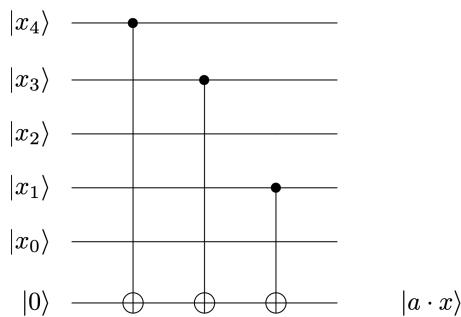


Figura 12.3: Ejemplo de función binaria lineal para $a = 11010$.

Ejercicio 48 Completa el código del notebook que genera el circuito asociado a la función binaria lineal $f(x; a)$.

Ejercicio 49 Sea sobre el conjunto de valores $x \in \{0, 1, 2, 3\}$ la función $f(x) = x^2$. Halla la tabla de verdad en binario y construye el oráculo que implementa esta función.

Jupyter Notebook: 8. Oráculos (funciones digitales)

Ver la sección 8.1. Construcción de funciones binarias. Los min-términos del notebook [8. Oráculos \(funciones digitales\)](#).

El Notebook puede descargarse de [Github](#).

12.2.2. Oráculos booleanos y de fase

En la sección anterior hemos definido el operador U_f usando un qubit ancilla

$$U_f |x\rangle \otimes |y\rangle = |x\rangle |y + f(x)\rangle \quad (12.10)$$

Dependiendo del valor que le demos a $|y\rangle$, tenemos dos clases de oráculos.

12.2.2.1. Oráculos booleanos.

Es la cláse de oráculos que vimos hasta ahora. Son aquellos en los que la salida se codifica como un 0 o un 1 en el bit $n + 1$, es decir

$$U_f |x\rangle \otimes |0\rangle = |x\rangle |f(x)\rangle \quad (12.11)$$

Es decir, tomamos $|y\rangle = 0$.

12.2.2.2. Oráculos de fase.

Nada nos impide inicializar la ancilla en un **autovector** de U_f .

Teorema 26 Los autovectores de U_f son los estados $|x\rangle \otimes |\pm\rangle$.

Demostración: Por un lado sabemos que los autovalores deben ser ± 1 dado que $U_f^2 = I$. Veamos cada caso

$$\begin{aligned} U_f |x\rangle \otimes |+\rangle &= |x\rangle \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + |1\rangle \right) = |x\rangle \otimes |+\rangle \\ U_f |x\rangle \otimes |-\rangle &= |x\rangle \otimes \frac{1}{\sqrt{2}} \left(|0\rangle - |1\rangle \right) = (-1)^{f(x)} |x\rangle \otimes |-\rangle \end{aligned}$$

donde vemos que se produce un típico efecto de *retroceso de fase*. ■

Si especificamos $|y\rangle = |-\rangle$ codificamos $f(x)$ en **la fase** $\rightarrow (-1)^{f(x)} = e^{i\pi f(x)}$.

Capítulo 13

Primeros algoritmos: algoritmos del oráculo.

En esta capítulo veremos una clase de algoritmos denominados **algoritmos del oráculo**. Estos son una colección de tres algoritmos didácticos, en el sentido de que no tienen aplicación práctica más allá de ser suficientemente simples como para que sea buena idea empezar el estudio por ellos. Aún así, el tercero de estos algoritmos, el algoritmo de Simon, sirvió como inspiración para que Peter Shor construyera el famoso algoritmo que lleva su nombre, el algoritmo de factorización de Shor.

Ya hemos visto que la computación cuántica destaca por su paralelismo. El ejemplo más fácil inicializar estado aplicando una puerta de Walsh-Hadammar sobre n qúbits. Si los n qúbits están en el estado $|0\rangle$ ya hemos visto que lo que obtenemos es la superposición uniforme de todos los estados de la base:

$$H^{\otimes n} |0\rangle = |00\cdots 0\rangle + |00\cdots 1\rangle + \cdots + |11\cdots 1\rangle \quad (13.1)$$

Esto es una ventaja respecto a un cálculo clásico, pues en el caso clásico tendríamos que ir inicializando uno a uno los estados. Sin embargo, este paralelismo es, en cierta medida, una ilusión. Si quisieramos medir las amplitudes de todos los estados de este tipo de superposiciones, tendríamos que repetir el cálculo (es este caso, aplicar $H^{\otimes n}$) un número exponencial de veces con el tamaño del registro n . Esto es debido a que, como ya se comentó, al medir **destruimos** la superposición, así que si queremos volver a medir sobre el estado del principio, tenemos que volver a construirlo.

La potencial de la computación cuántica reside en que podamos fabricar estados que concentran la solución a un problema en una o varias amplitudes. El foco se desplaza entonces a encontrar los *problemas adecuados*, en los que la solución al problema esté en un número pequeño de amplitudes.

Los problemas de **búsqueda de oráculo** consisten en desvelar alguna propiedad de la función que implementa dicho oráculo mediante el menor número de llamadas al oráculo.

Ejemplo

Denotaremos el conjunto $S_n = \{0, 1, \dots, 2^n - 1\} \sim \{0, 1\}^n$ indistintamente. Vamos a considerar una función binaria $f : S_n \rightarrow S_1$ implementada a través de un oráculo O que podemos consultar a placer. La implementación unitaria de f se realiza en la forma de un operador controlado

$$U_f : |x\rangle_n \otimes |y\rangle \rightarrow |x\rangle_n \otimes |y + f(x)\rangle \quad (13.2)$$

En particular, si $|y\rangle = |- \rangle$ tendremos el oráculo $f(x)$ **codificado en la fase**. Vamos a estudiar el circuito de la Fig. 13.1

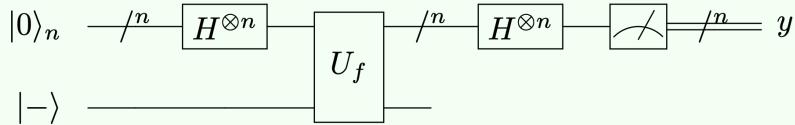


Figura 13.1: Circuito ejemplo de la implementación de un oráculo de fase sobre un qúbit ancila.

Los ingredientes de este circuito son:

- paralelismo, pues evaluamos el oráculo en **todos** los elementos de la base simultáneamente,
- codificación del oráculo en **la fase** (retroceso de fase)
- interferencia para **concentrar** la información en algunas amplitudes.

Veamos esto último viendo el estado de salida:

$$\begin{aligned}
 |\psi_0\rangle &= |0\rangle_n \otimes |- \rangle \\
 &\xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |0\rangle \otimes |- \rangle \quad (\text{paralelismo}) \\
 &\xrightarrow{U_f} \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |0\rangle \otimes (-1)^{f(x)} |- \rangle \quad (\text{codificación del oráculo en la fase}) \\
 &\xrightarrow{W_n} \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} (H^{\otimes n} |x\rangle) \otimes |-1\rangle = \frac{1}{2^n} \sum_{x,y=0}^{2^n-1} (-1)^{f(x)+x \cdot y} |y\rangle \otimes |- \rangle \quad (\text{interferencia}),
 \end{aligned}$$

donde

$$x \cdot y = x_{n-1}y_{n-1} \oplus x_{n-2}y_{n-2} \oplus \dots \oplus x_0y_0 \quad (13.3)$$

Vemos que el estado final está **factorizado**, por un lado tenemos el estado del primer registro

$$|\Phi\rangle = \frac{1}{2^n} \sum_{x,y=0}^{2^n-1} (-1)^{f(x)+x \cdot y} |y\rangle. \quad (13.4)$$

que se ha visto modificado debido al **retroceso de fase**, y por otro el estado del segundo registro, $|-\rangle$. Precisamente, como el estado de este último qúbit es independiente del primer registro, podemos ignorarlo y tratarlo como una **ancila**, es decir, una especie de qúbit auxiliar que usamos para un cálculo intermedio y después desecharmos, pues no lo medimos. Con respecto al primer registro, dependiendo de cómo sea $f(x)$ podremos conseguir *interferencias* que concentren la probabilidad en algún estado.

Nota: Recordatorio de la Walsh-Hadamard.

Ya que la vamos a usar mucho, recordemos la acción de la puerta de Walsh-Hadamard que vimos en la Ec. (6.17):

$$H^{\otimes n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} (-1)^{x \cdot y} |y\rangle \quad \text{donde } x \cdot y = x_{n-1}y_{n-1} \oplus x_{n-2}y_{n-2} \oplus \dots \oplus x_0y_0 \quad (13.5)$$

donde \oplus se refiere a **suma binaria**, es decir, $x \cdot y$ acaba valiendo 0 o 1.

Nota: ordenación de qúbits

Recordemos que la ordenación de los qúbits en el circuito tiene un convenio estándar que casi todo el mundo sigue. Sin embargo, uno de los principales agentes en este medio, IBM, usa un convenio distinto en su software Qiskit

Convenio estandar	Qiskit
$ a_{n-1}\rangle$ —	$ a_0\rangle$ —
$ a_{n-2}\rangle$ —	$ a_1\rangle$ —
\vdots	\vdots
$ a_0\rangle$ —	$ a_{n-1}\rangle$ —

Figura 13.2: Convenios de ordenación de los qúbits en la forma estándar, resaltando que Qiskit decide usar el convenio al revés.

Vamos a plantear los siguientes problemas del oráculo de la siguiente forma: tenemos una **promesa** y un **problema**. Es decir, sabemos algo del oráculo (la promesa) y queremos usar esta información junto con el propio oráculo para obtener más información sobre el mismo (el problema)

13.1. Algoritmo de Deutsch-Jozsa.

- **Promesa:** la función f que implementa el oráculo es de una de las dos siguientes clases:
 - *constante* ($C \Rightarrow f(x)$ igual para todo x)
 - *equilibrada* ($E \Rightarrow f(x)$ nos da tantos unos como ceros.)
- **Problema:** descubrir si f es de clase C o E . (Clásicamente, si nos toca un oráculo de clase C , deberíamos hacer $2^n/2 + 1$ consultas al oráculo para asegurarnos de que no es de clase E . Los de clase E con suerte podríamos diferenciarlos antes).
- **Solución:** corremos el circuito una vez y medimos sobre el estado $|\Phi\rangle$ (ver Ec. (13.4)):
 - Si $f \in C \Rightarrow$ la probabilidad de medir $|0^n\rangle$ es 1.
 - Si $f \in E \Rightarrow$ la probabilidad de medir $|0^n\rangle$ es 0.

El circuito es de la forma de la Fig. 13.3

Demostración: Veámoslo. Para ellos usamos la expresión de $|\Psi\rangle$ de la Ec. (13.4):

$$\begin{aligned}
 p_0 &= |\langle 0^n | \Phi \rangle|^2 \\
 &= \left| \langle 0^n | \sum_{y=0}^{2^n-1} \left(\frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)+x \cdot y} \right) |y\rangle \right|^2 \\
 &= \left| \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)x \cdot 0} \right|^2 = \left| \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \right|^2.
 \end{aligned}$$

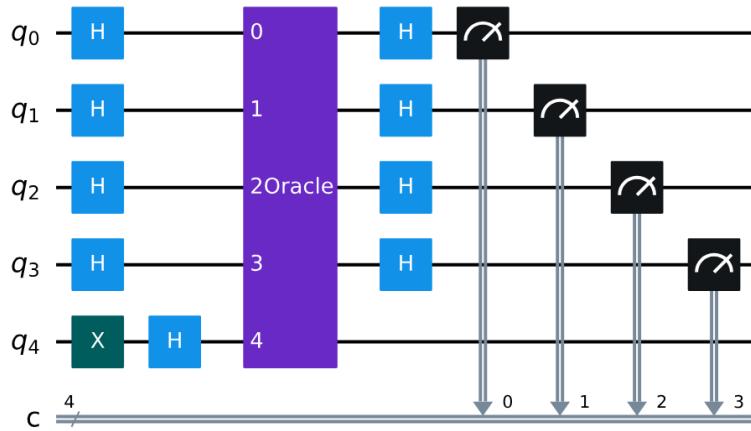


Figura 13.3: Forma genérica de los circuitos de Deutsch-Jozsa y Bernstein-Vazirani, donde el oráculo toma como entradas todos los quíbits menos el último, el quíbit ancila. Sobre este último quíbit es sobre el que se aplica el oráculo. Como el quíbit ancila se inicializa en el estado $|-\rangle$, estamos ante oráculos de fase.

Para el caso de $f \in C$ tenemos que $f(x) = f_0$ con lo que:

$$p_0 = \left| \sum_{x=0}^{2^n-1} (-1)^{f(x)} \right|^2 = \left| (-1)^{f_0} \frac{1}{2^n} \sum_{x=0}^{2^n-1} \right|^2 = \left| (-1)^{f_0} \right|^2 = 1.$$

Para el caso de $f \in E$ tenemos igual número de $f(x) = 0$ que de $f(x) = 1$, con lo que en el sumatorio tenemos tantos 1 como -1 :

$$p_0 = \left| \sum_{x=0}^{2^n-1} (-1)^{f(x)} \right|^2 = 0.$$

■

Ejercicio 50 Oráculos constantes sólo hay dos, $f(x) = 0$ ó $f(x) = 1$ para todo x . Oráculos equilibrados hay muchos. Construye un oráculo constante y uno equilibrado. Construye el circuito de Deutsch-Josza y ponlo a prueba con estos dos oráculos. Para ello, sigue el tutorial del Notebook de Qiskit (<https://learn.qiskit.org/course/ch-algorithms/deutsch-jozsa-algorithm>) .

Nota

Sobre el registro x que controla el oráculo, la acción del mismo tiene que ser la identidad. Es decir, el oráculo solo actúa sobre el registro objetivo. Este es el motivo por el cual en el tutorial anterior de qiskit se ponen tantas puertas X al principio como al final, pues hay que deshacer los cambios hechos sobre el registro de control.

13.2. Algoritmo de Bernstein-Vazirani

- **Promesa:** la función f es una *lineal*, definida por una cadena de bits $a \in \{0, 1\}^n$

$$f(x) = a \cdot x = a_{n-1}x_{n-1} \oplus \dots \oplus a_0x_0$$

- **Problema:** hallar $a = a_{n-1} \dots a_0$. (Clásicamente necesitaríamos invocar el oráculo n veces. Por

ejemplo $f(0 \cdots 01) = 0, 1$ revela $a_0 = 0, 1$ respectivamente. Iterativamente $f(0 \cdots 010) \rightarrow a_1, f(0 \cdots 100) \rightarrow a_2 \cdots$, etc.)

- **Solución:** correr el circuito una sola vez y medir el estado final. El circuito es de la forma de la Fig. 13.3.

Demostración: Veámoslo:

$$\begin{aligned}
|\Phi\rangle &= \frac{1}{2^n} \sum_{y=0}^{2^n-1} \left(\sum_{x=0}^{2^n-1} (-1)^{f(x)+y \cdot x} \right) |y\rangle = \frac{1}{2^n} \sum_{y=0}^{2^n-1} \left(\sum_{x=0}^{2^n-1} (-1)^{(a+y) \cdot x} \right) |y\rangle \\
&= \frac{1}{2^n} \sum_{y=0}^{2^n-1} \left(\sum_{x=0}^{2^n-1} (-1)^{(-a+y) \cdot x} \right) |y\rangle \\
&= \frac{1}{2^n} \delta_{y,a} \left(\sum_{x=0}^{2^n-1} 1 \right) |y\rangle = \delta_{y,a} |y\rangle \\
&= |a_0 a_1 \cdots a_{n-1}\rangle
\end{aligned}$$

En el primer paso (de la primera linea a la segunda) usamos la igualdad trivial $(-1)^a = (-1)^{-a}$. Entender el segundo paso y la aparición de la delta es un poco más complicado. Para ello, tenemos que darnos cuenta de que en el sumatorio en x vamos a tener tantos 1 como -1 , con lo cual va a ser 0 excepto para el valor de y para el cual el término $(-a + y) = 0$. Para este último, independientemente del valor de x , todos los exponentes serán cero, así que tendremos una suma de unos. ■

Jupyter Notebook: 9. Algoritmos del Oráculo)

Ver la sección 9.1. El problema de Bernstein-Vazirani del notebook 9. Algoritmos del Oráculo). El Notebook puede descargarse de [Github](#).

Nota: Bernstein-Vazirani en Qiskit

Se puede ver otra explicación del algoritmo de Bernstein-Vazirani junto con su implementación en Qiskit en el [Notebook de Qiskit](#) (<https://learn.qiskit.org/course/ch-algorithms/bernstein-vazirani-algorithm>) .

Nota: Porque funciona el circuito del algoritmo de Bernstein-Vazirani?

Supongamos que tomamos, por ejemplo, el caso $a = 010111$. Podemos ver como sería el circuito para este caso en la Fig. 13.4.

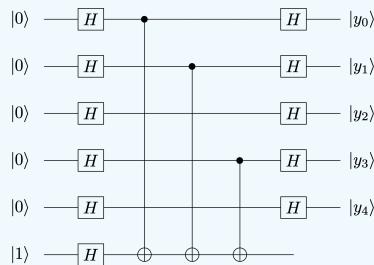


Figura 13.4: Circuito de Bernstein-Vazirani para $a = 010111$ (en la notación de Qisqit).

Usando el hecho de que $HH = I$, podemos añadir pares de puertas H entre las CNOTs en el qubit ancilla. Ahora solo tenemos que usar la identidad para invertir las CNOTs de la Fig. 11.7 y llegaremos al circuito de la Fig. 13.5

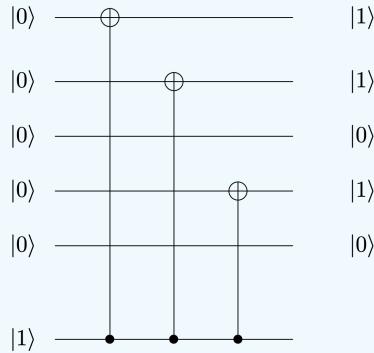


Figura 13.5: Circuito multiplica binariamente qubit a qubit $x \oplus a$. Para obtener el resultado de la multiplicación hay que hacer la multiplicación binaria de los qubit del final.

13.3. Algoritmo de Simon

Los algoritmos de Deutsch-Jozsa y Bernstein-Vazirani son deterministas, pues con un 100 % de probabilidad nos dan la solución. Si relajamos esta condición y nos quedamos con que nuestro algoritmo nos da la solución de forma probabilística, podemos ver muchos más algoritmos, como por ejemplo, el de Simon.

El algoritmo de Simon fue el primer algoritmo cuántico que mostró una aceleración exponencial frente al mejor algoritmo clásico en la resolución de un problema específico. Esto inspiró los algoritmos cuánticos basados en la transformada cuántica de Fourier (QFT), que se utiliza en el algoritmo cuántico más famoso: El algoritmo de factorización de Shor.

- **Promesa:** Consideremos ahora una función $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ con la siguiente propiedad: la función f puede ser de dos tipos:
 - **Uno-a-uno:** asigna una salida única para cada entrada. Un ejemplo sería el siguiente:

$$f(00) \rightarrow 01 \quad f(01) \rightarrow 11 \quad f(10) \rightarrow 00 \quad f(11) \rightarrow 10 \quad (13.6)$$

- **dos-a-uno:** asigna exactamente dos entradas a cada salida única. Este mapeo dos-a-dos es de acuerdo con una cadena de bits oculta a , donde

$$\text{dados } x_1, x_2 \text{ tal que } f(x_1) = f(x_2), \text{ es seguro que } x_1 \oplus x_2 = a \quad (13.7)$$

Equivalentemente, podemos escribir:

$$f(x_1 \oplus a) = f(x_2). \quad (13.8)$$

Un ejemplo con una función que toma 4 entradas es

$$f(00) \rightarrow 01 \quad f(01) \rightarrow 11 \quad f(10) \rightarrow 01 \quad f(11) \rightarrow 11 \quad (13.9)$$

Donde $00 \oplus 10 = 10$ y $01 \oplus 11 = 10$, con lo cual $s = 10$

Nota: Suma bit a bit

Obsérvese que $x_1 \oplus x_2 = a$ se refiere a suma módulo 2 bit a bit, es decir, sumamos el primer bit con el primero y, independientemente de cual sea el resultado de la suma, no nos llevamos nada. El ejemplo más claro es este: $00111 \oplus 00111 = 00000$.

- **Problema:** Dada esta caja negra f , como de rápido podemos determinar si f es uno-a-uno o dos-a-uno? Entonces, si f resulta ser dos-a-uno, como de rápido podemos determinar a ? En realidad los dos casos consisten en encontrar a , pues el caso uno-a-uno corresponde con $a = 00\dots$. (Clásicamente, si queremos conocer s con 100 % de certeza, tenemos que verificar hasta $2(n-1)+1$ entradas, donde n es el número de bits de la entrada. Es decir, necesitamos verificar la mitad de los casos.)

- **Solución:** El circuito será el de la Fig. 13.6

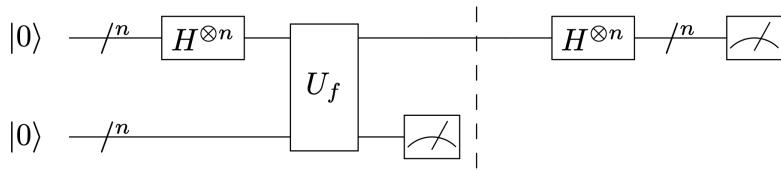


Figura 13.6: Circuito para el algoritmo de Simon.

A diferencia de los casos anteriores, ahora tenemos varias ancillas y están en el estado $|0\rangle_n$, es decir, tenemos un oráculo booleano, no de fase:

$$U_f \sum_{x=0}^{2^n-1} |x\rangle \otimes |0\rangle = \sum_{x=0}^{2^n-1} |x\rangle \otimes |f(x)\rangle . \quad (13.10)$$

Después de aplicar el oráculo el estado está entrelazado. Al hacer la medida del segundo registro el estado del segundo registro colapsará a un cierto estado $|f(x_0)\rangle$. En virtud de la *promesa* $|f(x_0)\rangle = |f(x_0 \oplus s)\rangle$, el primer registro colapsará a una *superposición de dos estados*:

$$|\psi\rangle = \frac{1}{\sqrt{2}} (|x_0\rangle + |x_0 \oplus s\rangle) . \quad (13.11)$$

justo antes de la barrera. Es decir, **la medida intermedia es parte importante del algoritmo**, pues queremos el colapso que provoca.

Siguiendo el circuito, aplicamos de nuevo la puerta de Walsh-Hadamard al primer registro

$$\begin{aligned} H^{\otimes n} \frac{1}{\sqrt{2}} (|x_0\rangle + |x_0 \oplus s\rangle) &= \frac{1}{\sqrt{2^{n+1}}} \sum_{y=0}^{2^n-1} [(-1)^{x_0 \cdot y} + (-1)^{(x_0 \oplus s) \cdot y}] |y\rangle \\ &= \frac{1}{\sqrt{2^{n+1}}} \sum_{y=0}^{2^n-1} [(-1)^{x_0 \cdot y} + (-1)^{x_0 \cdot y} (-1)^{s \cdot y}] |y\rangle \\ &= \frac{1}{\sqrt{2^{n+1}}} \sum_{y=0}^{2^n-1} (-1)^{x_0 \cdot y} \left(1 + (-1)^{s \cdot y} \right) |y\rangle \end{aligned} \quad (13.12)$$

Observar el factor

$$\frac{1}{2} (1 + (-1)^{s \cdot y}) = \begin{cases} 0 & \text{si } s \cdot y \pmod{2} = 1 \\ 1 & \text{si } s \cdot y \pmod{2} = 0 \end{cases} . \quad (13.13)$$

Este hace que **sólo tengan amplitud no nula** aquellos $|y\rangle$ con $s \cdot y \pmod{2} = 0$. Midiendo de forma repetida, el primer registro obtendremos una serie de n-bits $y^{(a)} = y^{(1)}, y^{(2)}, \dots, y^{(n)}$ todos los cuales verifican un sistema homogéneo de n ecuaciones lineales

$$\begin{aligned} s \cdot y^{(1)} \pmod{2} &= s_{n-1}y_{n-1}^{(1)} \oplus s_{n-2}y_{n-2}^{(1)} \oplus \dots \oplus s_0y_0^{(1)} = 0 \\ s \cdot y^{(2)} \pmod{2} &= s_{n-1}y_{n-1}^{(2)} \oplus s_{n-2}y_{n-2}^{(2)} \oplus \dots \oplus s_0y_0^{(2)} = 0 \\ &\vdots \\ s \cdot y^{(n)} \pmod{2} &= s_{n-1}y_{n-1}^{(n)} \oplus s_{n-2}y_{n-2}^{(n)} \oplus \dots \oplus s_0y_0^{(n)} = 0 \end{aligned} \tag{13.14}$$

donde todas las sumas se entienden módulo dos. Por un lado $s = s_{n-1} \dots s_0$ son nuestras incógnitas y, por otro, $y^{(a)} = y_{n-1}^{(a)} \dots y_0^{(a)}$ los coeficientes conocidos como resultado de las medidas. Dado que tenemos que averiguar los n bits que conforman la solución s necesitaremos, como mínimo, n ecuaciones linealmente independientes. Es decir, necesitamos medir mínimo n estados diferentes.

Una vez que hemos medido n estados diferentes, podemos hacer un post-procesado clásico para calcular s . Por ejemplo, eliminación gaussiana. Otra opción (poco eficiente) es hacerlo “*a lo bruto*”, cogiendo los n resultados y probando uno a uno con los 2^n valores posibles de s a ver cual es el que consigue que las n multiplicaciones modulo 2 $s \cdot y^{(i)}$ sean iguales a 0.

Jupyter Notebook: 9. Algoritmos del Oráculo)

Ver la sección 9.2. El problema de Simon del notebook 9. Algoritmos del Oráculo).
El Notebook puede descargarse de [Github](#).

Ejercicio 51 Completa el código del algoritmo de Simon en el notebook 08-Busqueda_Oracula

Nota: un algoritmo probabilístico (no determinista).

No hay garantía de que las cadenas de bits y obtenidos en las distintas evaluaciones del circuito sean siempre diferentes entre sí. Por tanto en general, para obtener un sistema lineal resoluble será necesario correr el circuito un número mayor de veces que n . Es por esta razón que el algoritmo de Simon es *probabilístico*.

Nota: Bernstein-Vazirani en Qiskit

Se puede ver otra explicación del algoritmo de Simon junto con su implementación en Qiskit en el [Notebook de Qiskit](#) (<https://learn.qiskit.org/course/ch-algorithms/simons-algorithm>).

Nota: primer algoritmo con ventaja exponencial

En este problema concreto el algoritmo cuántico realiza exponencialmente menos pasos que el clásico. Puede resultar difícil imaginar una aplicación de este algoritmo (aunque inspiró el algoritmo más famoso creado por Shor), pero representa la primera prueba de que puede haber una aceleración exponencial en la resolución de un problema específico utilizando un ordenador cuántico en lugar de uno clásico.

Capítulo 14

QFT: Quantum Fourier Transform

14.1. La QFT en computación cuántica

La transformada de Fourier aparece en muchas versiones diferentes a lo largo de la computación clásica, en ámbitos que van desde el procesamiento de señales a la compresión de datos, pasando por la teoría de la complejidad. La transformada cuántica de Fourier (QFT) es la implementación cuántica de la transformada discreta de Fourier sobre las amplitudes de una función de onda. Forma parte de muchos algoritmos cuánticos, entre los que destacan el algoritmo de factorización de Shor y la estimación cuántica de fase.

En su versión clásica, la transformada actúa sobre un vector $(x_0, x_1, \dots, x_{N-1})$ y lo mapea a otro vector $(\tilde{x}_0, \tilde{x}_1, \dots, \tilde{x}_{N-1})$ de acuerdo con la fórmula

$$\tilde{x}_k = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \omega^{ky} x_y, \quad \text{donde } \omega^{ky} = e^{2\pi i \frac{ky}{N}} \quad (14.1)$$

En concreto, la transformada de Fourier aplicada sobre cada uno de los vectores de la base nos lleva esta base a otra denominada **base de Fourier**. Esta transformación es biyectiva, es decir, mapea cada vector a un único vector, con lo cual, es invertible.

De forma análoga, la transformada de Fourier cuántica es un operador unitario que actúa sobre un estado cuántico $|\Psi\rangle = \sum_{j=0}^{N-1} \psi_j |j\rangle$ y lo mapea a otro estado cuántico $|\widetilde{\Psi}\rangle = \sum_{y=0}^{N-1} \tilde{\psi}_y |y\rangle$ de acuerdo con la fórmula

$$\tilde{\psi}_y = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega^{jy} \psi_j, \quad \text{donde } \omega^{jy} = e^{2\pi i \frac{jy}{N}} \quad (14.2)$$

Es decir, la transformación de un estado genérico $|\psi\rangle$ sería:

$$|\Psi\rangle \rightarrow |\widetilde{\Psi}\rangle = \sum_{y=0}^{N-1} \tilde{\psi}_y |y\rangle = \sum_{y=0}^{N-1} \left(\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega^{jy} \psi_j \right) |y\rangle \quad (14.3)$$

Véase que aquí tanto $|\psi\rangle$ como $|\widetilde{\Psi}\rangle$ están expresados en la misma base, por ejemplo, la base computacional.

Nota: base computacional y base de fourier

Independientemente de la “etiqueta” que haya dentro del ket, estaremos tratando con elementos base computacional **cuando el elemento no lleve tilde**, es decir, $|x\rangle$, $|j\rangle$, Estaremos tratando con la transformada de Fourier de ese elemento de la base (es decir, el correspondiente elemento de la base pero en la base de Fourier) cuando les pongamos una tilde: $|x\rangle \rightarrow |\tilde{x}\rangle$, $|j\rangle \rightarrow |\tilde{j}\rangle$,

Por otro lado, reservamos las letras griegas para referirnos a **estados**, es decir, combinaciones lineales de elementos de la base: $|\Psi\rangle = \sum_{j=0}^{N-1} \psi_j |j\rangle$

Veamos como actúa la QFT sobre los elementos de la base computacional. Para ello, solo tenemos que tomar $|\Psi\rangle$ como un elemento de la base en la Ec. (14.3). Es decir

$$|\Psi\rangle = \sum_{j=0}^{N-1} \psi_j |j\rangle \xrightarrow[\psi_x=1]{\psi_j=0 \forall j \neq x} |\Psi\rangle = |x\rangle \quad (14.4)$$

Haciendo esta misma sustitución en la Ec. (14.3) llegamos a

$$|x\rangle \rightarrow |\tilde{x}\rangle = \sum_{y=0}^{N-1} \left(\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega^{jy} \psi_j \right) |y\rangle \xrightarrow[\psi_x=1]{\psi_j=0 \forall j \neq x} QFT|x\rangle = |\tilde{x}\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \omega^{xy} |y\rangle \quad (14.5)$$

donde

$$\boxed{\omega = e^{2\pi i/N}}, \quad x = 0, \dots, N-1, \quad \tilde{x} = \widetilde{0, \dots, N-1}, \quad \text{y} \quad N = 2^n. \quad (14.6)$$

La Ec. (14.5) es la transformada de Fourier de un elemento de la base computacional $|x\rangle$. Esta nos da el elemento transformado en la base de Fourier $|\tilde{x}\rangle$. Como podemos ver, la transformada de Fourier es un operador unitario. Como también sabemos, un operador está completamente definido si especificamos como actúa sobre todos los elementos de una base, como es el caso.

Nota: QFT y Walsh-Hadamard

Comparar la Ec. (14.5) con la de la transformada de Walsh-Hadamard (13.5)

$$W : |x\rangle \rightarrow |\tilde{x}\rangle = \frac{1}{\sqrt{N}} \sum_y (-1)^{xy} |y\rangle = \frac{1}{\sqrt{N}} \sum_y e^{i\pi xy} |y\rangle \quad (14.7)$$

que es formalmente igual, pero con $\omega = e^{2\pi i/2} = -1$. Ya hemos visto cómo este factor produce *interferencias* interesantes que conducen a soluciones como la del problema de Simon.

El nuevo factor $\omega = e^{2\pi i/N}$ también sirve para producir interferencias destructivas y constructivas interesantes. Esto se debe esencialmente a la importante fórmula de **suma nula**

$$\frac{1}{N} \sum_{y=0}^N e^{2\pi ixy/N} = \delta_{x,0 \bmod N} \quad (14.8)$$

Como vemos, esta suma en y es cero para todos los valores de x excepto para $x = 0, N, 2N, \dots$

Ejercicio 52 Utiliza la Ec. (14.8) para demostrar que, invirtiendo los signos de las fases ob-

tenemos la QFT inversa. Es decir, demuestra que, si

$$U_{TFC}^{-1} |x\rangle = \frac{1}{\sqrt{N}} \sum_y e^{-2\pi i xy/N} |y\rangle \quad (14.9)$$

se sigue que

$$U_{TFC}^{-1}(U_{TFC} |x\rangle) |x\rangle \quad (14.10)$$

Esto confirma que es un operador unitario $U_{TFC}^{-1} = U_{TFC}^\dagger$.

En general, los elementos de matriz serán evidentemente, las fases

$$\langle x | U_{QFT} | y \rangle = \frac{1}{\sqrt{N}} e^{2\pi i xy/N} = \frac{1}{\sqrt{N}} \omega^{xy} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_N & \omega_N^2 & \omega_N^3 & \cdots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \omega_N^6 & \cdots & \omega_N^{2(N-1)} \\ 1 & \omega_N^3 & \omega_N^6 & \omega_N^9 & \cdots & \omega_N^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \omega_N^{3(N-1)} & \cdots & \omega_N^{(N-1)(N-1)} \end{bmatrix}$$

Ejemplos

- Para $n = 1 \rightarrow \omega = e^{2\pi i/2^1} = -1$ y la QFT no es otra que la puerta de Hadamard

$$U_{QFT} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = H \quad (14.11)$$

Su acción es

$$U_{QFT} |0\rangle = |+\rangle \quad , \quad U_{QFT} |1\rangle = |-\rangle \quad (14.12)$$

Observamos que los vectores imagen están situados en el plano ecuatorial de la esfera de Bloch

- Para $n = 2 \rightarrow \omega = e^{2\pi i/2^2} = i$ y entonces

$$U_{QFT} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix} \quad (14.13)$$

Observar que la suma de cualquier columna o fila que no sean las primeras da cero

En el primer ejemplo es fácil de ver que los vectores de la base de Fourier se sitúan sobre el plano xy . En el segundo caso también es cierto, pero es más difícil de ver a primera vista. Esta es una regla de la QFT:

Teorema 27 Los vectores de la **base de Fourier** son estados **factorizables** en productos de estados de un qubit que se sitúan **sobre el ecuador** de la esfera de Bloch (sobre el plano xy)

Uno de los puntos importantes del anterior teorema es que nos dice que la base de Fourier es factorizable, no entrelazada. Esto puede parecer sorprendente, pues la QFT es una suma de muchos elementos con fases distintas. Como la base de Fourier es factorizable, podemos dibujarla como productos de esferas de Bloch, como se puede ver en la Fig. 14.1. En esta figura podemos ver varios ejemplos de transformadas de Fourier de elementos de la base computacional para 4 qubits. Veamos ahora la demostración del teorema:

Nota: Notación binaria con decimales

Antes de ver la demostración del teorema, veamos la notación binaria con decimales: Un número $x \in \mathbb{Q}$ puede expresarse en base 2 como

$$\begin{aligned} x &= x_{n-1}2^{n-1} + \cdots + x_12^1 + x_02^0 + x_{-1}2^{-1} + \cdots + x_{-m}2^{-m} \\ &= x_{n-1} \dots x_1 x_0 . x_{-1} \dots x_{-m} \\ &= x_{n-1} \dots x_1 x_0 + 0.x_{-1} \dots x_{-m} \end{aligned}$$

en particular, si $y \in S_n = [0, \dots, 2^n - 1]$, en base 2 tendríamos

$$\begin{aligned} y &= y_{n-1} \dots y_2 y_1 y_0 \\ y/2 &= y_{n-1} \dots y_2 y_1 + 0.y_0 \\ y/2^2 &= y_{n-1} \dots y_2 + 0.y_1 y_0 \\ &\vdots \\ y/2^n &= 0.y_{n-1} y_{n-2} \dots y_0 \end{aligned} \tag{14.14}$$

Demostración: Vamos a estudiar la acción de U_{TFC} sobre un elemento $|x\rangle = |x_{n-1}\rangle |x_{n-2}\rangle \dots |x_0\rangle$ de la base computacional

$$\begin{aligned} |\tilde{x}\rangle \equiv U_{\text{TFC}}|x\rangle &= \frac{1}{\sqrt{N}} \sum_{y=0}^{2^n-1} e^{2\pi i xy/2^n} |y\rangle \\ &= \frac{1}{\sqrt{N}} \sum_{y_1, \dots, y_n=\{0,1\}} e^{2\pi i x(y_{n-1}2^{n-1} + y_{n-2}2^{n-2} + \dots + y_0)/2^n} |y_{n-1}\rangle |y_{n-2}\rangle \dots |y_0\rangle \\ &= \frac{1}{\sqrt{N}} \sum_{y_1, \dots, y_n=\{0,1\}} e^{2\pi i x(\frac{y_{n-1}}{2} + \frac{y_{n-2}}{2^2} + \dots + \frac{y_0}{2^n})} |y_{n-1}\rangle |y_{n-2}\rangle \dots |y_0\rangle \\ &= \frac{1}{\sqrt{N}} (|0\rangle + e^{2\pi i \frac{x}{2}} |1\rangle) (|0\rangle + e^{2\pi i \frac{x}{2^2}} |1\rangle) \dots (|0\rangle + e^{2\pi i \frac{x}{2^n}} |1\rangle) \\ &= \frac{1}{\sqrt{N}} (|0\rangle + e^{2\pi i 0.x_0} |1\rangle) (|0\rangle + e^{2\pi i 0.x_1 x_0} |1\rangle) \dots (|0\rangle + e^{2\pi i 0.x_{n-1} \dots x_0} |1\rangle) \\ &\equiv |\tilde{x}_{n-1}\rangle |\tilde{x}_{n-2}\rangle \dots |\tilde{x}_0\rangle \end{aligned}$$

Se ha usado que las partes enteras de x no contribuyen a las fases (es un número entero multiplicado por 2π)

$$e^{2\pi i x/2} = e^{2\pi i (x_{n-1} \dots x_1 + 0.x_0)} = e^{2\pi i (x_{n-1} \dots x_1)} e^{2\pi i 0.x_0} = e^{2\pi i 0.x_0} \tag{14.15}$$

Vemos que la transformada de Fourier es factorizable y cada elemento de la base vive en el ecuador de la esfera de Bloch. ■

Como acabamos de ver en la desmostración anterior, tenemos que:

$$|\tilde{x}\rangle \equiv U_{\text{TFC}}|x\rangle = \frac{1}{\sqrt{N}} (|0\rangle + e^{2\pi i \frac{x}{2}} |1\rangle) (|0\rangle + e^{2\pi i \frac{x}{2^2}} |1\rangle) \dots (|0\rangle + e^{2\pi i \frac{x}{2^n}} |1\rangle) \tag{14.16}$$

En realidad, esta expresión es tan útil e importante que podríamos haberla tomado esta como la definición de la transformada de Fourier, pues es equivalente a la Ec. (14.5). Otra forma de escribirla

es con la notación decimal anterior y teniendo en cuenta el resultado de la Ec. (14.15) es

$$|\tilde{x}\rangle \equiv U_{\text{TFC}}|x\rangle = \frac{1}{\sqrt{N}} (|0\rangle + e^{2\pi i 0.x_0}|1\rangle) (|0\rangle + e^{2\pi i 0.x_1 x_0}|1\rangle) \dots (|0\rangle + e^{2\pi i 0.x_{n-1} \dots x_0}|1\rangle) \quad (14.17)$$

Nota: QFT sobre vectores

Por supuesto, la operación U_{TFC} no está restringida a elementos de la base computacional. Ya hemos visto en la Ec. (14.3) que puede actuar sobre vectores.

14.2. Intuición

La transformada cuántica de Fourier (QFT) transforma entre dos bases, la base computacional (Z) y la base de Fourier. La puerta H es la transformada de Fourier para un solo qúbit, y esta transforma entre la base Z ($|0\rangle$ y $|1\rangle$) en la base X ($|+\rangle$ y $|-\rangle$). Del mismo modo, todos los estados multi-qúbit en la base computacional tienen estados correspondientes en la base de Fourier. La QFT es simplemente la función que transforma entre estas bases.

$$|\text{Estado en la Base Computacional}\rangle \xrightarrow{\text{QFT}} |\text{Estado en la Base de Fourier}\rangle$$

$$\text{QFT}|x\rangle = |\tilde{x}\rangle$$

Habitualmente denotamos los estados en la base de Fourier usando la tilde (\sim)

14.2.1. Contando en la base de Fourier.

En la base computacional, almacenamos los números en binario usando los estados $|0\rangle$ y $|1\rangle$. En la base de Fourier, almacenamos números utilizando diferentes rotaciones alrededor del eje Z. En este segundo caso, el número que queremos almacenar dicta los ángulos que se rotan cada qúbit.

Como podemos observar en la Fig. 14.1, la frecuencia con la que cambian los distintos qúbits es diferente:

- En la base computacional, el qúbit situado más a la izquierda en las imágenes (menos significativo) cambia con cada incremento del número, el siguiente con cada 2 incrementos, el tercero con cada 4 incrementos, y así sucesivamente.
- En la base de Fourier, cada qúbit se rota el doble del anterior, empezando por el de la izquierda en las imágenes (el menos significativo). En el estado $|\tilde{0}\rangle$ todos los qúbit están en el estado $|+\rangle$. Como también podemos ver en las imágenes, para codificar, por ejemplo, $|\tilde{5}\rangle$ en 4 qúbits, rotamos el qúbit de la izquierda en las imágenes (el menos significativo) un ángulo $\frac{5}{2^n} \times 2\pi = \frac{5}{16} \times 2\pi$. El siguiente qúbit se rota el doble ($2 \frac{5}{16} \times 2\pi$), el siguiente el doble de este y así sucesivamente.

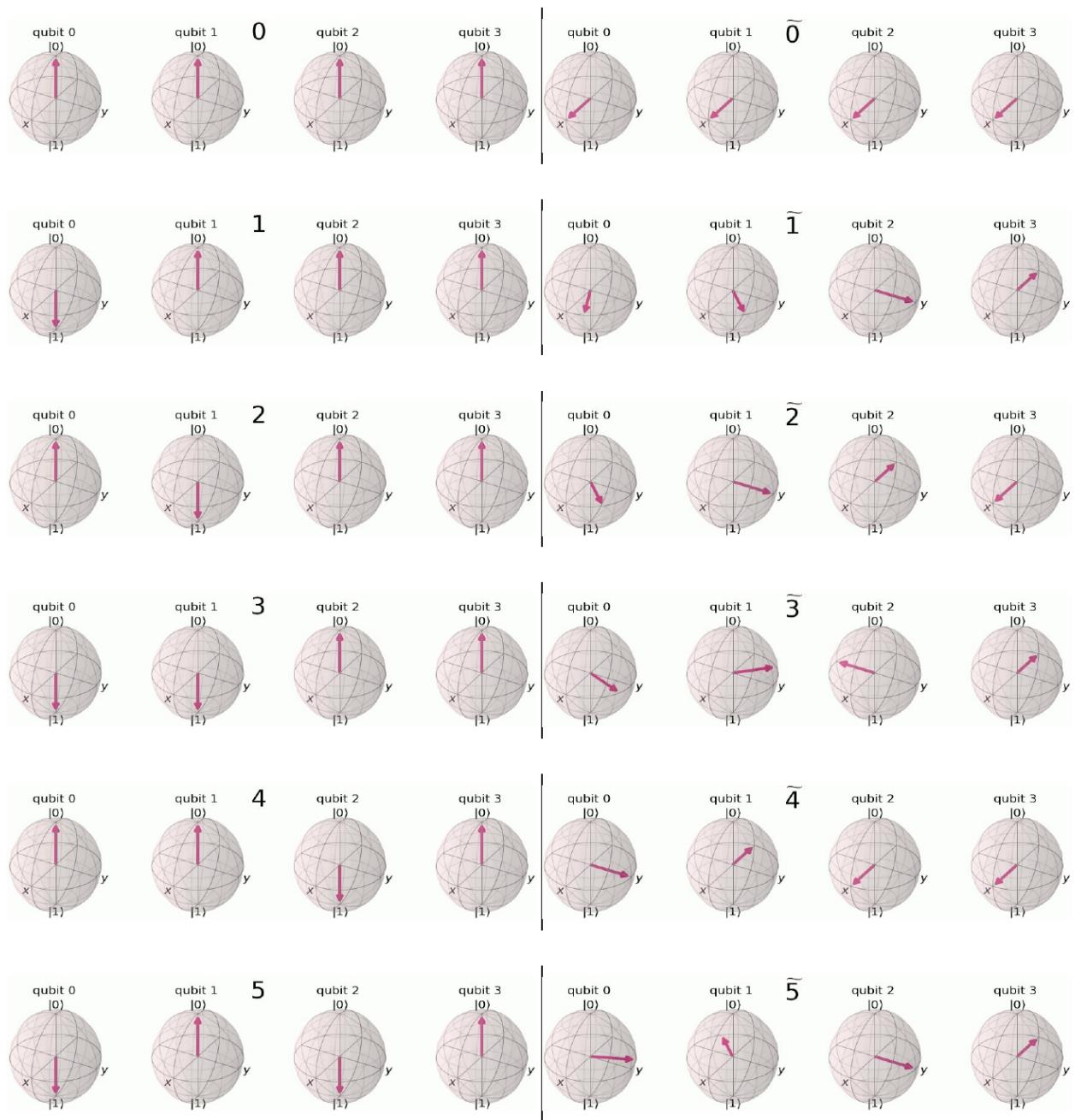


Figura 14.1: Base computacional y base de Fourier

14.3. Circuito que implementa la QFT.

Sin más preámbulo, vamos a ver como es el circuito que implementa la QFT. Podemos ver el mismo en la Fig. 14.2.

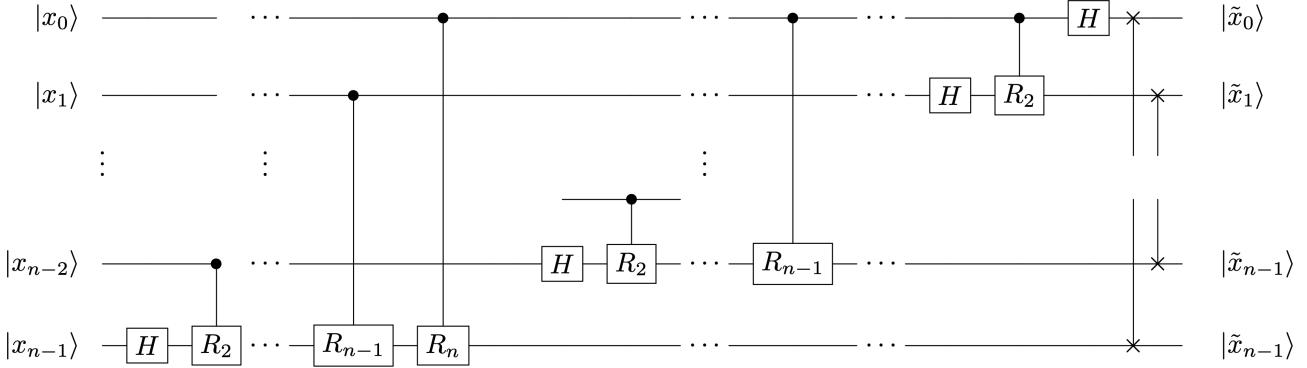


Figura 14.2: Implementación de la QFT (circuito en el convenio de Qiskit)

Vemos que este circuito solo incluye tres elementos:

- Puertas de Hadamard H ,
- Puertas de fase discreta $R_k \equiv P(\phi = \pi/2^{k-1})$

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i \frac{1}{2^k}} \end{bmatrix} \quad \Leftrightarrow \quad R_k |y\rangle = e^{2\pi i \frac{y}{2^k}} |y\rangle \quad (14.18)$$

pero en su forma controlada CR_k :

$$CR_k |x\rangle |y\rangle = |x\rangle R_k^x |y\rangle = |x\rangle e^{2\pi i \frac{y}{2^k} x} |y\rangle \quad (14.19)$$

donde $|x\rangle$ es el qúbit de control.

- Puertas SWAP.

Demostración: Vamos a analizar la acción del circuito de la Fig. 14.2 para ver que efectivamente nos da la expresión de la Ec. (14.16). Comencemos viendo la acción del primer bloque:

$$(H |x_{n-1}\rangle) |x_{n-2}...x_0\rangle = \left(|0\rangle + e^{2\pi i (\frac{x_{n-1}}{2})} |1\rangle \right) |x_{n-2}...x_0\rangle$$

$$\begin{aligned} (R_2^{x_{n-2}} H |x_{n-1}\rangle) |x_{n-2}...x_0\rangle &= \left(|0\rangle + e^{2\pi i (\frac{x_{n-1}}{2} + \frac{x_{n-2}}{2^2})} |1\rangle \right) |x_{n-2}...x_0\rangle \\ &\vdots \\ \left(R_{(n-1)}^{x_0} ... R_3^{x_{n-3}} R_2^{x_{n-2}} H |x_{n-1}\rangle \right) |x_{n-2}...x_0\rangle &= \left(|0\rangle + e^{2\pi i (\frac{x_{n-1}}{2} + \frac{x_{n-2}}{2^2} + \dots + \frac{x_0}{2^n})} |1\rangle \right) |x_{n-2}...x_0\rangle \\ &= \left(|0\rangle + e^{2\pi i 0.x_{n-1}...x_0} |1\rangle \right) |x_{n-2}...x_0\rangle \\ &\equiv |\tilde{x}_0\rangle |x_{n-2}...x_1x_0\rangle \end{aligned}$$

El primer bloque ha generado el estado ecuatorial $|\tilde{x}_0\rangle$ pero *en la posición equivocada*. Si repetimos el mismo procedimiento con los siguientes qúbits $|x_{n-2}\rangle$, obtendremos finalmente

$$|\tilde{x}_0\rangle |\tilde{x}_1\rangle ... |\tilde{x}_{n-2}\rangle |\tilde{x}_{n-1}\rangle \quad (14.20)$$

La parte final del circuito introduce los operadores de SWAP que rectifican el orden de los qubits

$$\text{SWAP}^{\otimes n} (|\tilde{x}_0\rangle \dots |\tilde{x}_{n-1}\rangle) = |\tilde{x}_{n-1}\rangle \dots |\tilde{x}_0\rangle \equiv |\tilde{x}\rangle \quad (14.21)$$

■

14.3.1. Algunos comentarios sobre la implementación.

Obsérvese que sólo el último qubit depende de los valores de todos los demás qubits de entrada y que cada qubit posterior depende cada vez menos de los qubits de entrada. Esto es importante en las implementaciones físicas de la QFT, donde los acoplamientos entre vecinos más cercanos son más fáciles de conseguir que los acoplamientos entre qubits distantes.

Además, a medida que el circuito QFT se hace grande, se emplea una cantidad de tiempo cada vez mayor en realizar rotaciones cada vez más ligeras (hay rotaciones de hasta $2\pi/2^n$). Resulta que podemos ignorar las rotaciones por debajo de un cierto umbral y seguir obteniendo resultados decentes, lo que se conoce como **QFT aproximada**. Esto también es importante en las implementaciones físicas, ya que reducir el número de operaciones puede reducir en gran medida la decoherencia y los posibles errores de puerta.

Jupyter Notebook: 10. Quantum Fourier Transform (QFT)

Ver el notebook [10. Quantum Fourier Transform \(QFT\)](#).

El Notebook puede descargarse de [Github](#).

14.4. Complejidad y QFT aproximada

14.4.1. Complejidad y ventaja exponencial

El número de puertas que hemos necesitado es n puertas de Hadamard y $n(n - 1)/2$ fases controladas CR . En total esto es un número de orden $\mathcal{O}(n^2)$. Clásicamente, el algoritmo más eficiente para calcular la Transformada de Fourier Discreta precisa de $\mathcal{O}(n2^n)$ por tanto la QFT transforma un problema de tipo NP en uno de tipo P .

En realidad no hemos calculado la QFT, ya que del estado final no podemos deducir las fases de los elementos de la base separadamente, las cuales constituye la transformada de Fourier del qubit de entrada. Por tanto, el punto estará en ser capaces de encontrar problemas en los que la QFT sea un ingrediente que aporte una ventaja exponencial. Este es el caso del algoritmo de Shor.

14.4.2. QFT aproximada

Como acabamos de comentar, cuando aplicamos al QFT exacta sobre n qubits necesitamos del orden de $\mathcal{O}(n^2)$ operaciones/puertas (en concreto, son $\frac{1}{2}n(n + 1)$ operaciones). Si nos fijamos en la puerta CR_k que aparece en la QFT vemos que es de la forma

$$CR_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \exp\left(\frac{2\pi i}{2^k}\right) \end{bmatrix} \quad (14.22)$$

Se puede apreciar que cuando k crece mucho, las puertas CR_k se aproximan a la Identidad. Esto hace que podamos prescindir de las puertas con un valor k mayor que un cierto umbral k_{max} y aplicar así una versión **aproximada** (con menos operaciones) de la QFT. Puede demostrarse (ver [22]) que

el error introducido por ignorar todas las puertas con un valor $k > k_{max}$ es proporcional a $n2^{-k_{max}}$. Podemos pues tomar k_{max} del orden de $\mathcal{O}(\log_2 n)$, pasando de tener del orden de $\mathcal{O}(n^2)$ operaciones (puertas) a $\mathcal{O}(n \log_2(n))$ operaciones (en concreto $\frac{1}{2}(2n - \log_2 n)(\log_2 n - 1)$ operaciones).

Capítulo 15

QPE: Estimación de Fase Cuántica

15.1. Introducción

La estimación de fase cuántica (Quantum Phase Estimation - **QPE**) es una pieza fundamental de algoritmos más complejos, como por ejemplo el de Shor.

Este algoritmo sirve para calcular los autovalores de un **operador unitario**. Como sabemos, los operadores unitarios son los únicos que podemos aplicar a un estado cuántico (son las puertas en los circuitos), pues son los únicos que preservan la normalización de los estados cuánticos, es decir, que las probabilidades sumen la unidad. Esto se manifiesta en que los autovalores de los operadores unitarios tiene módulo 1, es decir, son **fases**. Dado un operador unitario U y un autovector $|\psi\rangle$ del mismo, tenemos:

$$U |\psi\rangle = e^{2\pi i \theta} |\psi\rangle \quad (15.1)$$

El algoritmo lo que hace es estimar el valor de θ . Es decir:

- **Promesa:** Podemos preparar $|\psi\rangle$ y aplicar el operador U tantas veces como queramos.
- **Problema:** Calcular la mejor aproximación posible a θ .

Nota: Periodicidad de la fase

Las fases $e^{i2\pi\theta} = e^{i2\pi(\theta+n)}$ son iguales, para cualquier $n \in \mathbb{Z}$ entero. Por tanto será suficiente considerar $\theta \in [0, 1)$ para generar *todos* los posibles autovalores distintos.

15.2. Circuito

En la Fig. 15.1 podemos ver el circuito que implementa el algoritmo de estimación de fases. Podemos ver que el circuito consta de tres partes. El primer registro tiene dimensión t , mientras que el segundo registro tiene dimensión n . La entrada es el estado $|0\rangle_t \otimes |\psi\rangle$:

- La dimensión t del primer registro controlará la *precisión de nuestra aproximación a θ* .
- La dimensión n del espacio al que pertenece $|\varphi\rangle$ es la necesaria para servir de espacio de representación para U .

Vemos que aparece también la **inversa de la transformada de Fourier**, esto es, QFT^\dagger

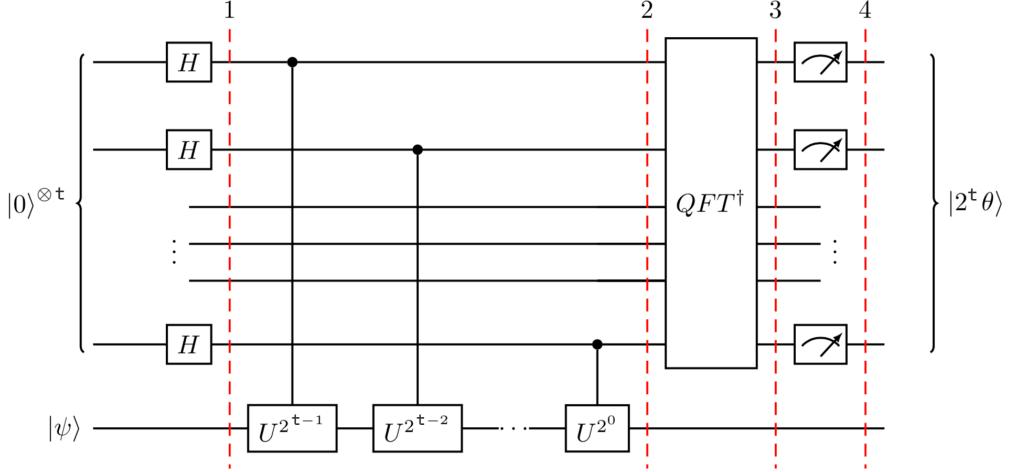


Figura 15.1: Implementación del algoritmo de estimación de fase cuántica (en el convenio estándar)

15.3. Formulación matemática

Veamos paso a paso que hace el circuito anterior para estimar la fase θ .

1. **Inicialización:** Por un lado tenemos un registro de qubits que forman un autoestado $|\psi\rangle$ del operador U . Por otro lado tenemos un conjunto de t qubits que forman un **registro de conteo** donde al final del circuito tendremos almacenado el valor $2^t\theta$:

$$|\Psi_0\rangle = |0\rangle^{\otimes t} |\psi\rangle \quad (15.2)$$

2. **Superposición:** Aplicamos una operación de puerta de Hadamard t-bit al registro de conteo:

$$|\Psi_1\rangle = \frac{1}{2^{t/2}} (|0\rangle + |1\rangle)^{\otimes t} |\psi\rangle \quad (15.3)$$

3. **Operaciones unitarias controladas:** Aplicamos sucesivas veces el operador controlado CU , es decir, aplicar U en el registro objetivo solo si el qubit controlador está en el estado $|1\rangle$. En concreto, aplicamos 2^j veces U en el registro $|\psi\rangle$ controlado por un qubits del registro de conteo. El número de veces 2^j que aplicamos U depende de que qubit es el controlador (si es el bit más significativo tenemos $j = t - 1$, para el siguiente $j = t - 2$, ..., hasta llegar al bit menos significativo en el cual $j = 0$).

En la notación del convenio estandar, el qubit más significativo es el primero. En Qiskit es al revés.

Como U es unitaria y $|\psi\rangle$ es autovector de U , aplicar 2^j veces U se traduce en:

$$U^{2^j} |\psi\rangle = U^{2^j-1} U |\psi\rangle = U^{2^j-1} e^{2\pi i \theta} |\psi\rangle = \dots = e^{2\pi i 2^j \theta} |\psi\rangle \quad (15.4)$$

Usando la relación

$$\begin{aligned} CU \left[(|0\rangle + |1\rangle) \otimes |\psi\rangle \right] &= CU \left[|0\rangle \otimes |\psi\rangle + |1\rangle \otimes |\psi\rangle \right] \\ &= |0\rangle \otimes |\psi\rangle + |1\rangle \otimes e^{2\pi i \theta} |\psi\rangle \\ &= (|0\rangle + e^{2\pi i \theta} |1\rangle) \otimes |\psi\rangle , \end{aligned}$$

llegamos a

$$\begin{aligned} |\Psi_2\rangle &= \frac{1}{2^{t/2}} \left(|0\rangle + e^{2\pi i \theta 2^{t-1}} |1\rangle \right) \otimes \cdots \otimes \left(|0\rangle + e^{2\pi i \theta 2^1} |1\rangle \right) \otimes \left(|0\rangle + e^{2\pi i \theta 2^0} |1\rangle \right) \otimes |\psi\rangle \\ &= \frac{1}{2^{n/2}} \sum_{y=0}^{2^t-1} e^{2\pi i \theta y} |y\rangle \otimes |\psi\rangle \end{aligned}$$

Nota

Podemos ver que si por el registro de conteo entra un estado $|z\rangle = |z_{n-1}, z_{n-2}, \dots, z_0\rangle$, la salida antes de aplicar la QFT^{-1} es de la forma

$$|z\rangle |\psi\rangle \rightarrow |z\rangle U^{z_{t-1} 2^{t-1}} U^{z_{t-2} 2^{t-2}} \cdots U^{z_0 2^0} |\psi\rangle = |z\rangle U^z |\psi\rangle \quad (15.5)$$

Es decir, lo que se hace es aplicar z veces el operador U sobre el estado $|\psi\rangle$.

4. **Transformada de Fourier inversa:** Si nos fijamos en detalle vemos que la expresión anterior es igual al resultado de aplicar la QFT un estado de t qubits $|x\rangle$ (ver Ec. (14.16))

$$(U_{QFT} |x\rangle) \otimes |\psi\rangle = \frac{1}{2^{t/2}} \sum_{y=0}^{2^t-1} e^{2\pi i xy / 2^t} |y\rangle \otimes |\psi\rangle \quad (15.6)$$

si tomamos $x = 2^t \theta$.

Lo que podemos hacer es aplicar la transformada de Fourier inversa en el registro de conteo para obtener el valor $2^t \theta$

$$\begin{aligned} |\Psi_3\rangle &= \left[U_{QFT}^\dagger \left(\frac{1}{2^{t/2}} \sum_{y=0}^{2^t-1} e^{2\pi i \theta y} |y\rangle \right) \right] \otimes |\psi\rangle = \left[\frac{1}{2^{t/2}} \sum_{y=0}^{2^t-1} e^{2\pi i \theta y} (U_{QFT}^\dagger |y\rangle) \right] \otimes |\psi\rangle \\ &= \left[\frac{1}{2^{t/2}} \sum_{y=0}^{2^t-1} e^{2\pi i \theta y} \left(\sum_{x=0}^{2^t-1} e^{-2\pi i y x / 2^t} |x\rangle \right) \right] \otimes |\psi\rangle \\ &= \left[\frac{1}{2^t} \sum_{x=0}^{2^t-1} \sum_{y=0}^{2^t-1} e^{-\frac{2\pi i y}{2^t} (x - 2^t \theta)} |x\rangle \right] \otimes |\psi\rangle \end{aligned}$$

En general, $2^t \theta$ no será un número entero, así que, en general $(x - 2^t \theta) \neq 0 \ \forall x \in \mathbb{Z}$.

5. **Medida.** Aunque no parezca obvio, la expresión anterior está picada entorno a $x = 2^t \theta$ incluso para el caso en el que $2^t \theta$ no sea un número entero. Para verlo, primero sepáremos el número $2^t \theta$ en su parte entera y decimal de la siguiente forma

$$2^t \theta = a + \delta$$

con $a \in \mathbb{Z}$ y $\delta \in [0, 1)$. Con esta separación, el resultado del apartado anterior nos queda

$$|\Psi_3\rangle = \left[\frac{1}{2^t} \sum_{x=0}^{2^t-1} \sum_{y=0}^{2^t-1} e^{\frac{2\pi i y}{2^t} (a + \delta - x)} |x\rangle \right] \otimes |\psi\rangle \quad (15.7)$$

- Si $\delta = 0$, es decir, si $2^t\theta$ fuese un entero $a \in S_t = \{0, \dots, 2^t - 1\}$, entonces el resultado sería **exactamente** $|\Psi_4\rangle = |a\rangle = |2^t\theta\rangle$

$$\begin{aligned} |\Psi_4\rangle &= \left[\sum_{x=0}^{2^t-1} \frac{1}{2^t} \left(\sum_{y=0}^{2^t-1} e^{\frac{2\pi i}{2^t} y(a-x)} \right) |x\rangle \right] \otimes |\psi\rangle \\ &= \left[\sum_{x=0}^{2^t-1} \frac{1}{2^t} (2^t \delta_{x,a}) |x\rangle \right] \otimes |\psi\rangle \\ &= |a\rangle \end{aligned}$$

En este caso, midiendo el primer registro obtendríamos un registro binario del número $a \in [0, 2^t - 1]$. Con a recuperaríamos la fase buscada *de forma exacta*:

$$\varphi = \frac{a}{2^t} \in [0, 1) \quad (15.8)$$

- Si $\delta \neq 0$ (si $2^t\theta$ no es entero), el estado en el primer registro $|\Phi\rangle = \sum_{x=0}^{2^t-1} f(x) |x\rangle$ será una superposición.

$$|\Psi_4\rangle = \left[\sum_{x=0}^{2^t-1} \left(\frac{1}{2^t} \sum_{y=0}^{2^t-1} e^{2\pi i(a+\delta-x)y/2^t} \right) |x\rangle \right] \otimes |\psi\rangle = \left[\sum_{x=0}^{2^t-1} f(x) |x\rangle \right] \otimes |\psi\rangle \quad (15.9)$$

Una medida del primer registro dará el registro binario de *un número entero* $x \rightarrow m \in [0, 2^t - 1]$ con distribución de probabilidad

$$p(m) = |f(m)|^2 \quad (15.10)$$

picada en $m = a = [2^t\varphi]$.

Jupyter Notebook: [11. Quantum Phase Estimation \(QPE\)](#)

Ver la sección [11.1. Ejemplo: Un solo autoestado en la ancilla](#) del notebook [11. Quantum Phase Estimation \(QPE\)](#).

El Notebook puede descargarse de [Github](#).

La **anchura t del circuito auxiliar** determina la anchura de la curva de probabilidad en torno al valor medio, es decir, como de precisa será nuestra estimación de la fase. Es fácil convencerse de que cuanto mayor sea t , mayor será nuestra precisión. Podemos acotar la aproximación al valor real de la fase buscada según el siguiente teorema (ver Nielsen [23] p. 224)

Teorema 28 *El algoritmo QPE (Quantum Phase Estimation) es capaz de producir una estimación m de orden k para la fase φ (en el sentido de que $|\varphi - m/2^t| < 2^{-k}$) con una probabilidad $1 - \epsilon$, tomando una dimensión del espacio de representación*

$$t \geq k + \left\lceil \log \left(1 + \frac{1}{2\epsilon} \right) \right\rceil \quad (15.11)$$

15.4. ¿Y si no conocemos el autoestado?

Hemos visto que el algoritmo de estimación de fase requiere de seamos capaces de preparar un autoestado. Esto en realidad es contradictorio, pues para conocer un autoestado de un operador genérico

(no diagonal), lo que tenemos que hacer es diagonizarlo. Diagonalizar un operador (una matriz) es una operación costosa y además, una vez diagonalizada la matriz, es trivial calcular los autovalores. Es decir, tal y como está planteado hasta ahora el problema de estimación de fase, no nos aporta nada debido a la restricción de conocer un autovector.

Sin embargo, no es necesario conocer un autovector en concreto. El secreto está en que los autovectores de un operador forman una **base**, de forma que cualquier estado se puede escribir como combinación lineal de estos autoestados. Supongamos por ejemplo que tenemos un operador U con autovectores $|\psi_i\rangle$ y autovalores $2\pi i \theta_i$:

$$U |\psi_i\rangle = e^{2\pi i \theta_i} |\psi_i\rangle \quad (15.12)$$

Si tenemos por ejemplo un estado genérico $|b\rangle$, podemos escribirlo en la base de autovectores de U :

$$|b\rangle = \sum_{i=1}^N c_i |\psi_i\rangle \quad (15.13)$$

Debido a linealidad del circuito QPE , a la salida del mismo (antes de la medida) entraremos una combinación lineal de estados de la forma

$$\begin{aligned} U_{QPE} : |0\rangle_t \left(\sum_{i=1}^N c_i |\psi_i\rangle \right) &\rightarrow |\Psi_3\rangle = \sum_{i=1}^N c_i U_{QPE} \left(|0\rangle_t |\psi_i\rangle \right) \\ &= \sum_{i=1}^N c_i |\Phi_i\rangle |\psi_i\rangle \\ &= \sum_{i=1}^N c_i |\Psi_{3,i}\rangle \end{aligned}$$

donde $|\Phi_i\rangle$ es el estado del registro de t qubits antes de la media en la Ec. (15.7), es decir

$$|\Psi_{3,i}\rangle = \left[\frac{1}{2^t} \sum_{x=0}^{2^t-1} \sum_{y=0}^{2^t-1} e^{\frac{2\pi i y}{2^t} (2^t \theta_i - x)} |x\rangle \right] \otimes |\psi_i\rangle = |\Phi_i\rangle \otimes |\psi_i\rangle = |\Phi_i\rangle |\psi_i\rangle \quad (15.14)$$

Nota: Aclaración

Para entender un poco este resultado, quedémonos primero con la expresión siguiente:

$$|\Psi_3\rangle = \sum_{i=1}^N c_i |\Phi_i\rangle |\psi_i\rangle \quad (15.15)$$

y comparemosla con la expresión de la Ec. (15.7). Vemos que ahora lo que tenemos es una combinación lineal de los resultados para cada autovector que compone nuestro estado $|b\rangle$. Para entenderlo mejor, podemos verlo al revés. El caso de la Ec. (15.7) es el caso en el que $|b\rangle$ es igual a un solo autovector, es decir, en el que **todos los c_i son cero excepto uno de ellos**.

Como ya comentamos, si $2^t \theta_i$ no es entero, el estado del primer registro será una superposición de la forma $|\Phi_i\rangle = \sum_{x=0}^{2^t-1} f_i(x) |x\rangle$, es decir

$$|\Psi_3\rangle = \sum_{i=1}^N c_i \left(\sum_{x=0}^{2^t-1} f_i(x) |x\rangle \right) |\psi_i\rangle \quad (15.16)$$

Una medida del primer registro dará el registro binario de *un número entero* $x \rightarrow m \in [0, 2^t - 1]$ con distribución de probabilidad

$$p(m) = \sum_i^N |c_i|^2 |f_i(m)|^2 \quad (15.17)$$

Comparando este resultado con el de la Ec. (15.10) vemos que lo que obtenemos es una **suma ponderada de las distribuciones de probabilidad para cada autovalor**. Es decir:

- Cuando nuestro $|b\rangle$ es un autovector, lo que obtenemos es la distribución de probabilidad para su autovalor (ver Ec. (15.10)). Como ya hemos comentado, esta está picada entorno al valor binario que más se parece al autovalor que buscamos.
- Cuando $|b\rangle$ es un estado genérico, podemos escribirlo en la base de autovectores de U (ver Ec. (15.13)). Es decir, entendemos $|b\rangle$ como una combinación lineal de autovectores con coeficientes c_i . En este caso obtenemos las distribuciones de probabilidades de cada uno de los autovectores de U (distribuciones picadas entorno al valor binario de los autovalores) pero ponderadas por los $|c_i|^2$.

Esto quiere decir que dependiendo de como sea $|b\rangle$, este podrá tener más componente de unos autovectores que de otros. Con lo cual, dependiendo de como sea $|b\rangle$ podremos estimar unos autovalores o de otros.

Nota

Esta última sección puede resultar un poco confusa, pero se entiende mejor si pensamos en un ejemplo concreto. Como vamos a ver a continuación, en el algoritmo de Shor se usa la QPE. Como no conocemos los autoestados del operador, lo que se hace es usar una propiedad del propio operador para, de una manera sencilla, darle al algoritmo de QPE en vez de un autoestado, **una superporsición homogénea de todos los autoestados**. Es decir, en el algoritmo de Shor estamos en el caso en el que todos los $|c_i|^2$ son iguales.

De esta forma, cada vez que ejecutamos el algoritmo estamos obteniendo una estimación de uno de los autovalores y tenemos la misma probabilidad de estimar un autovalor que otro. Esto es así porque, como ya veremos, sabemos que con alguno de estos autovalores podemos realizar un calculo que nos permite factorizar un número. Como no sabemos que autovalor nos sirve, vamos probando a hacer ejecuciones del circuito hasta que damos con uno que nos sirve. Como todos los autovalores tienen la misma probabilidad de salir, no tardaremos mucho en encontrar uno que nos sirva.

Jupyter Notebook: 11. Quantum Phase Estimation (QPE)

Ver la secciones 11.2. ¿y si no podemos preparar un autoestado? y 11.3. Precisión del notebook 11. Quantum Phase Estimation (QPE).

El Notebook puede descargarse de [Github](#).

Capítulo 16

Algoritmo de Shor (Periodicity Finding)

El algoritmo de Shor se basa en el algoritmo de **Estimación de Fase Cuántica (Quantum Phase Estimation)**, que a su vez usa la **Transformada de Fourier Cuántica (Quantum Fourier Transform)**. El algoritmo de Shor lo que hace es convertir el problema de la factorización de un número en un problema de **encontrar el periodo de una función**, el cual puede ser implementado en un tiempo polinómico.

Para factorizar un número N básicamente el algoritmo de Shor lo que hace calcular el periodo de una función (periódica) de la forma

$$f(x) = a^x \bmod N$$

donde a y N son enteros positivos mayores que 1, siendo además $a < N$ y no teniendo factores comunes. La operación ($z \bmod N$) a lo que se refiere es a quedarnos con el **resto** de dividir el número que z por N . El periodo r de esta función se calcula mediante el algoritmo de estimación de fase cuántica. Una vez se tiene el periodo r , si este es par (sino hay que probar con otro valor de a) se pueden calcular los factores de N ya que existe una alta probabilidad de que el máximo común divisor de N y $(a^{r/2} - 1)$ o $(a^{r/2} + 1)$ sea un factor propio de N .

16.1. Introducción.

16.1.1. Criptografía y factorización.

El algoritmo de Shor es uno de los algoritmos de computación cuántica más conocidos debido a que es mejor en su tarea que cualquier algoritmo de computación clásica conocido hasta la fecha. Además, resuelve un problema que tiene una aplicación práctica directa: factorizar un número.

Para entender porque este algoritmo es tan importante debemos hablar primero de criptografía, en concreto, del encriptado de transmisiones a través de Internet mediante el método de clave pública y clave privada (**criptografía asimétrica** usando el famoso **algoritmo RSA**). La explicación conceptual de esta forma de encriptar es simple. Cuando quieras, por ejemplo, acceder a la aplicación de tu banco tienes que ingresar las claves de acceso (DNI y pin), estas se mandan por Internet hasta la sede de tu banco, el cual verifica que son correctas y te da acceso. El problema radica precisamente en que la conexión se hace mediante Internet, con lo cual el mensaje con las claves de acceso puede ser interceptado. La solución a este problema es que el mensaje que el cliente envía esté encriptado y que solo tu banco pueda desencriptar el mensaje.

El método de encriptación más usado en Internet usando el algoritmo RSA. En este tipo de encriptado

el receptor del mensaje (en nuestro ejemplo, el banco) genera dos claves dependientes entre sí, una la publicará al exterior (clave pública) y otra solo la conocerá él (clave privada). Si un receptor quisiera recibir un mensaje encriptado, bastaría con que publicase su clave pública de forma que cualquiera que quiera mandarle un mensaje, pueda usarla para encriptar el mismo. Sin embargo, la clave privada solo es conocida por el receptor del mensaje, y se usa para desencriptar. Puede decirse que la clave pública es como un candado y la clave privada es la llave. Cualquiera puede cerrar el candado, pero solo el que tiene la llave puede abrirlo.

El punto importante aquí es que, la clave privada son dos números primos (de gran tamaño, decenas de cifras), y la clave pública es la multiplicación de estos dos números. La solidez de este método de cifrado (RSA) radica en el hecho de que si tenemos dos números primos multiplicarlos es muy fácil, pero si tenemos la multiplicación de los mismos (la clave pública) hallar cuales son los dos números con los que se construyó (factorizar el número en sus elementos primos) es extremadamente difícil. Como es esperable, cuanto más larga es la clave, más tiempo se tarda en factorizarla. El problema radica específicamente en que el tiempo que se requiere crece **exponencialmente** con el número de bits. Para las longitudes de clave que se manejan actualmente, incluso con los mejores superordenadores se tardarían cientos o miles de años en hallar los factores.

La tremenda potencia y aplicabilidad del algoritmo de Shor es que convierte este problema de complejidad exponencial en el número de bits para un computador clásico, en un problema de complejidad polinómica para un computador cuántico. Es decir, con el algoritmo de Shor el tiempo requerido para factorizar un número crece **polinómicamente** con el número de cifras del número. De esta forma, si se llega a tener un ordenador cuántico con suficientes qubits como para aplicar este algoritmo a números de la longitud de clave que se usa actualmente, se podrían factorizar y hallar la clave privada en un tiempo razonable para la escala humana. El algoritmo de Shor tiene el potencial de romper la criptografía asimétrica y hacer vulnerables las comunicaciones a través de la red, pero estamos muy lejos de tener un ordenador cuántico capaz de implementarlo a la escala requerida. Se estima que se necesitarían del orden del millón de qubits, mientras que actualmente (año 2023) los ordenadores cuánticos más grandes andan por el orden de cientos de qubits.

16.1.2. Algoritmo de factorización.

El algoritmo de Shor se basa en el hecho de poder reducir un problema de factorización a uno de **period finding (hallar el periodo -orden- de una función)**. Antes de hablar de nada cuántico, vamos a ver, tal y como se describe en el Nielsen-Chuang [23], como sería la estructura general de un algoritmo de factorización de esta forma comentando en que punto entra la computación cuántica para acelerarlo.

Lo primero es introducir la noción de **números coprimos**:

Definición 27 Dos números a y b son coprimos si su máximo común divisor es 1, esto es, $\gcd(a, b) = 1$. Es decir, dos números coprimos solo comparten como divisor común el 1.

Los pasos para reducir un problema de factorización en uno de period finding son los siguientes. Sea N el número que queremos factorizar:

1. Si N es par, devolver el factor 2.
2. Determinar si $N = p^b$ para los enteros $p \geq 1$ y $b \geq 2$, y si es así, devolver el factor p (puede hacerse en un tiempo polinómico).
3. Elegir un número entero aleatorio a tal que $1 < a \leq N - 1$. Usando el algoritmo de Euclides, determinar si $\gcd(a, N) > 1$. Si lo es, devolver el factor $\gcd(a, N)$.

4. Si $\gcd(a, N) = 1$ (a y N son coprimos) y $a^{r/2} \neq -1 \pmod{N}$, calculamos el periodo r de la función $f(x) = a^x \pmod{N}$.
5. Si r es impar, volvemos al paso 3. Sino, calculamos $\gcd(a^{r/2}-1, N)$ y $\gcd(a^{r/2}+1, N)$. Probamos a ver si uno de estos dos es un factor no-trivial de N , y devolvemos el mismo si lo es.

Todos los pasos de este algoritmo, excepto el **paso 4**, se pueden implementar en un ordenador clásico y resolverse en un tiempo polinómico. Esto es debido a que para calcular el máximo común divisor se puede usar el Algoritmo de Euclides [24], el cual resuelve el problema en un **tiempo polinómico** (se puede calcular en un tiempo razonable).

El paso complicado y que, por lo menos hasta la fecha, no hay ninguna forma de implementarlo en un tiempo polinómico en un ordenador clásico (se implementa en un **tiempo exponencial**) es el **paso 4**, hallar el periodo de la función. Sin embargo, este paso puede implementarse en un ordenador cuántico en un tiempo polinómico. Tenemos pues que la forma óptima de factorizar un número consiste en **implementar los pasos 1, 2, 3 y 5 en un ordenador clásico, y el paso 4 en un ordenador cuántico**.

16.1.3. Explicación cualitativa

Vamos a intentar entender de forma cualitativa porqué calculando el periodo de una función se pueden hallar los factores de un número. En la sección 16.1.4 veremos un poco más en detalle las afirmaciones que se hacen en esta sección.

La función que nos interesa es la siguiente

$$f(x) = a^x \pmod{N} \quad (16.1)$$

donde a y N son enteros positivos mayores que 1, siendo además $a < N$ y no teniendo factores comunes, es decir, que sean coprimos ($\gcd(a, N) = 1$). La operación $(z \pmod{N})$ a lo que se refiere es a quedarnos con el **resto** de dividir el número que z por N .

La función $f(x)$ se denomina **exponentiales moduladas**. Esta se encaja dentro de la **aritmética modular** y si se cumplen las condiciones anteriores esta función es periódica. Denominaremos r al **valor del periodo de la función** $f(x)$, es decir, r es el mínimo valor entero para que se cumple:

$$f(x+r) = f(x). \quad (16.2)$$

Este se puede calcular mediante un circuito cuántico.

Una vez se tiene el periodo r , si este es par (sino hay que probar con otro valor de a) se pueden calcular los factores de N . Esto es debido a que

$$a^r \pmod{N} = 1 \quad (16.3)$$

con lo cual

$$(a^r - 1) \pmod{N} = 0 \quad (16.4)$$

Con lo cual, N debe ser un divisor de $a^r - 1$. Si r es par (sino hay que probar con otro valor de a), podemos escribir:

$$a^r - 1 = (a^{r/2} - 1)(a^{r/2} + 1) \quad (16.5)$$

Entonces tenemos una alta probabilidad de que el **máximo común divisor** de N y $a^{r/2} - 1$ o $a^{r/2} + 1$ sea un factor propio de N .

16.1.4. Formalismo matemático

Veamos un poco el formalismo matemático detrás de las afirmaciones de la sección anterior.

16.1.4.1. Periodicidad de $f(x)$

Demostrar que, dada la condición $\gcd(a, N) = 1$, la función $f(x) = a^x \bmod N$ es periódica no es fácil, pues se necesitan plantear varios teoremas y la explicación se hace árida (puede verse el Appendix 4 de [23]). Aquí vamos a ver una explicación más simple partiendo del siguiente resultado (sin demostrarlos):

Lema 6 *Dada la función $f(x) = a^x \bmod N$, si se cumple que $\gcd(a, N) = 1$, tenemos que para algún valor entero $z > 0$ se cumple $f(z) = a^z \bmod N = 1$.*

Vemos a ver ahora que este valor $z > 0$ para el cual se cumple $f(z) = a^z \bmod N = 1$ será el periodo de la función. Denominaremos a este valor r , es decir, r será **el primer valor (mayor que cero) para el cual se cumple $f(r) = 1$** . Tenemos que

$$a^0 = 1 \rightarrow f(0) = a^0 \bmod N = 1 = a^r \bmod N = f(r). \quad (16.6)$$

En el momento en el que llegamos a un exponente r tal que $a^r \bmod N = 1$ podemos escribir

$$a^r = \alpha N + 1 \quad (16.7)$$

con lo cual

$$\begin{aligned} f(r+z) &= a^{r+z} \bmod N = a^r a^z \bmod N = (\alpha N + 1)a^z \bmod N = \\ &= \alpha N a^z \bmod N + a^z \bmod N = a^z \bmod N = f(z) \end{aligned} \quad (16.8)$$

Hemos visto pues que $f(x)$ es periódica.

16.1.4.2. Factores de N a partir del periodo r

Para entender como pasar del periodo r de nuestra función a tener los factores de N nos hace falta conocer un par de teoremas, ambos presentes en [23].

Teorema 29 *(Teorema 5.2 del Nielsen-Chuang) Supongamos que N es un número compuesto de L bits, y x es una solución no trivial de la ecuación $(x^2 = 1 \bmod N)$ en el rango $1 \leq a \leq N$, esto es, ni $x = (1 \bmod N)$ ni $x = N - 1 = (-1 \bmod N)$. Entonces, uno de $\gcd(x - 1, N)$ y $\gcd(x + 1, N)$ es un factor no trivial de N que se puede calcular usando $\mathcal{O}(L^3)$ operaciones.*

Demostración: Ya que $x^2 \bmod N = 1 \rightarrow (x^2 - 1) \bmod N = 0$, debe de cumplirse que N divide a $(x^2 - 1) = (x + 1)(x - 1)$, con lo cual N debe de tener un factor común con $(x + 1)$ o con $(x - 1)$. Como por suposición tenemos que $1 < x < N - 1$, con lo cual $x - 1 < x + 1 < N$, de lo cual podemos ver que el factor común no puede ser el propio N . Usando el Algoritmo de Euclides [24] podemos calcular $\gcd(x - 1)$ y $\gcd(x + 1)$, y con lo cual obtener un factor no trivial de N , usando \mathcal{O} operaciones. ■

Teorema 30 *(Teorema 5.3 del Nielsen-Chuang) Supongamos $N = p_1^{\alpha_1} \dots p_m^{\alpha_m}$ es la descomposición en factores primos de un entero impar positivo. Sea x un número entero elegido uniformemente al azar, sujeto a la restricción $1 \leq x \leq N - 1$ y x coprimo de N . Sea r el periodo de $x \bmod N$. Entonces*

$$p(r \text{ es impar y } x^{r/2} \bmod N \neq -1) \geq 1 - \frac{1}{2^m} \quad (16.9)$$

esto es, la probabilidad de hallar un r impar y que cumpla $x^{r/2} \bmod N \neq -1$ es mayor que $1 -$

$1/2^m$.

Nota

En nuestro caso el teorema 5.2 se aplica con $x = a^r$ y el teorema 5.3 con $x = a$.

El teorema 5.3 nos da la probabilidad de éxito del algoritmo para cada cálculo de r . Para el caso que comentamos de la criptografía tenemos $m = 2$ (dos claves privadas), así que la probabilidad de obtener un r válido es $1/2$.

Los teoremas 5.2 y 5.3 pueden combinarse para dar un algoritmo que, con alta probabilidad, devuelve un factor no trivial de cualquier numero no primo N . Todos los pasos del algoritmo pueden realizarse de forma eficiente en un ordenador clásico, excepto (por lo que se sabe hoy en día) una subrutina de búsqueda de periodo que utiliza el algoritmo. Repitiendo el procedimiento podemos encontrar una factorización prima completa de N .

16.2. Hallar del periodo de una función (Period Finding)

Como hemos comentado, el paso 4 (el de la búsqueda de periodo) descrito en la sección 16.1.2 se puede implementar en un ordenador cuántico. Para ello lo que se hace es reducir el problema a un problema de **QPE (Estimación de Fase Cuántica)** vista en el capítulo 15.

16.2.1. La función

Como comentamos en la introducción, lo que queremos es hallar el periodo de la función

$$f(x) = a^x \bmod N \quad (16.10)$$

donde a y N son enteros positivos mayores que 1, siendo además $a < N$ y no teniendo factores comunes. La operación $(z \bmod N)$ a lo que se refiere es a quedarnos con el **resto** de dividir el número que z por N . A este tipo de funciones se las denomina **exponenciales moduladas**.

Denominaremos r al valor del periodo de la función $f(x)$, es decir, r es el mínimo valor entero para que se cumple:

$$f(x + r) = f(x) \quad (16.11)$$

En la Fig. 16.1 vemos un ejemplo de este tipo de funciones con $a = 3$ y $N = 35$. Vemos que para este caso el periodo es $r = 12$.

16.2.2. Solución: Estimación de fase de un operador U

El algoritmo de Shor se basa en implementar el algoritmo de estimación de fases al operador unitario

$$U|y\rangle \equiv |ay \bmod N\rangle \quad (16.12)$$

Al aplicar sucesivas veces el operador U sobre el estado $|1\rangle$ vamos obteniendo los valores de $f(x)$ con $x \in \mathbb{N}$, esto es,

$$U^x|1\rangle = |f(x)\rangle \quad (16.13)$$

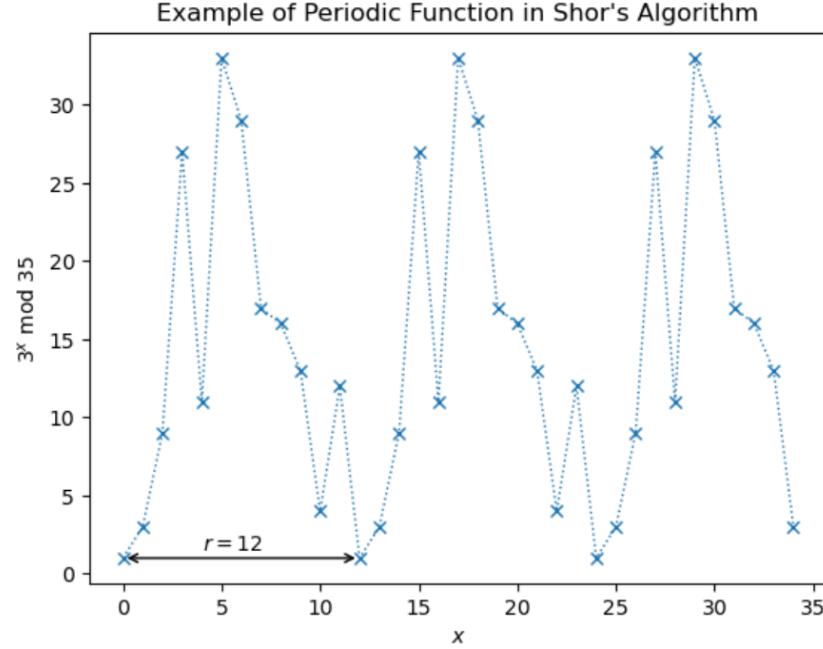


Figura 16.1: Gráfica de los primeros valores de la función periódica $f(x) = a^x \bmod N$ con $a = 3$ y $N = 15$. Véase que las líneas puntuadas que unen las cruces son solo por estética.

Por ejemplo, para el caso que vimos en la gráfica anterior ($a = 3$ y $N = 35$) tenemos

$$\begin{aligned} U^0|1\rangle &= |1\rangle \\ U|1\rangle &= |3\rangle \\ U^2|1\rangle &= |9\rangle \\ &\vdots \\ U^{r-1}|1\rangle &= |12\rangle \\ U^r|1\rangle &= |1\rangle \end{aligned}$$

(Recordemos que dado un estado de n qubits $|x\rangle$ tenemos que $|x\rangle = |x_1x_2\dots x_n\rangle$ donde x_1 es el bit más significativo.)

Como podemos ver, aplicar una vez más el operador U significa pasar de un número al siguiente de la lista periódica. Veámoslo explícitamente:

$$\begin{aligned} U(U^0|1\rangle) &= U(|1\rangle) = |3\rangle \\ U(U|1\rangle) &= U(|3\rangle) = |9\rangle \\ U(U^2|1\rangle) &= U(|9\rangle) = |27\rangle \\ &\vdots \\ U(U^{r-1}|1\rangle) &= U(|12\rangle) = |1\rangle \\ U(U^r|1\rangle) &= U(|1\rangle) = |3\rangle \end{aligned}$$

Con esto se entiende fácilmente que la superposición equiprobable de todos los estados es un autoestado del operador U con autovalor 1:

$$|u_0\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |a^k \bmod N\rangle, \quad \text{donde } U|u_0\rangle = |u_0\rangle \quad (16.14)$$

Ejemplo

Ejemplo: caso con $a = 3$ y $N = 35$

$$\begin{aligned} U|u_0\rangle &= U\left[\frac{1}{\sqrt{12}}(|1\rangle + |3\rangle + |9\rangle + \dots + |4\rangle + |12\rangle)\right] = \\ &= \frac{1}{\sqrt{12}}(U|1\rangle + U|3\rangle + U|9\rangle + \dots + U|4\rangle + U|12\rangle) = \\ &= \frac{1}{\sqrt{12}}(|3\rangle + |9\rangle + |27\rangle + \dots + |12\rangle + |1\rangle) = \\ &= |u_0\rangle \end{aligned}$$

Un autoestado de autovalor 1 no nos es muy interesante a la hora de aplicar el algoritmo de estimación fase. Otro conjunto de autoestados mucho más interesantes son aquellos de la forma:

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i k \frac{s}{r}} |a^k \bmod N\rangle, \quad \text{donde } U|u_s\rangle = e^{2\pi i \frac{s}{r}} |u_s\rangle \quad (16.15)$$

donde $0 \leq s \leq r - 1$. Si ahora aplicamos el algoritmo de estimación de fase cuántica a uno de estos autoestados $|u_s\rangle$, lo que obtendremos en el registro de conteo es $|2^n s/r\rangle$. De aquí podemos extraer el valor de r . Sin embargo, para preparar el estado $|u_s\rangle$ tenemos que conocer r , es decir, lo que queremos calcular.

Una solución elegante y fácil de implementar es darnos cuenta de que la suma de todos estos estados $|u_s\rangle$ nos da el estado $|1\rangle$, esto es,

$$\frac{1}{r} \sum_{s=0}^{r-1} |u_s\rangle = \frac{1}{r} (|u_0\rangle + |u_1\rangle + \dots + |u_{r-1}\rangle) = |1\rangle \quad (16.16)$$

Ejercicio 53 Comprueba la veracidad de la ecuación (16.16) para el caso $a = 7$ y $N = 15$.

Si ahora aplicamos el algoritmo de estimación de fase cuántico (QPS) al estado $|1\rangle$ (un estado fácilmente implementable) obtenemos una superposición equiprobable de estados de la forma $|2^n s/r\rangle$, es decir:

$$|0\rangle|1\rangle \xrightarrow{QPS} \frac{1}{\sqrt{r}} \left(\left| 2^t \frac{1}{r} \right\rangle + \left| 2^t \frac{2}{r} \right\rangle + \dots + \left| 2^t \frac{r-1}{r} \right\rangle \right) |1\rangle \quad (16.17)$$

donde t es el número de qubits del registro de conteo. Usando el algoritmo de las fracciones continuas [25] podemos calcular r a partir de los cocientes s/r . En Fig. 16.2 podemos ver el circuito (en el orden de Qiskit para los qubits) que implementa la estimación de fase cuántica

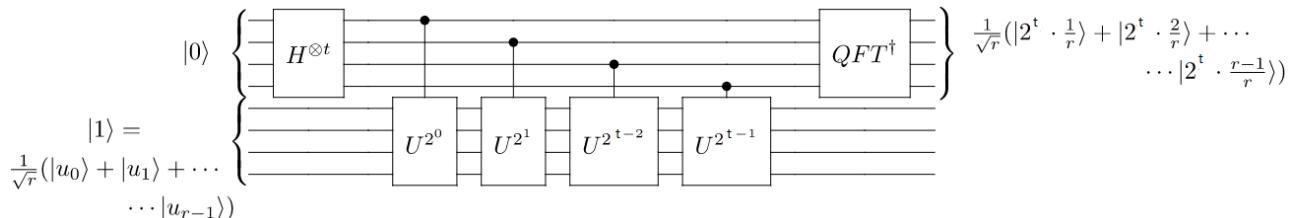


Figura 16.2: Implementación del algoritmo de estimación de fase cuántica (en el convenio de Qiskit).

Nota

Denominamos n al número de qubits que necesitamos para codificar el número N (que queremos factorizar) en un registro cuántico. Para aplicar el algoritmo de Shor se suelen usar $t = 2n$ qubits en el registro de conteo.

16.3. Implementación (ad hoc) en Qiskit para $N = 15$

En esta sección vamos a ver una implementación un poco *ad hoc* del algoritmo de Shor para factorizar el número 15. Con lo de *ad hoc* a lo que nos referimos es a que el oráculo que representa la exponencial modulada del operador U está construido específicamente para el caso de factorizar 15 y no se va a explicar como funciona. El objetivo de esta sección es ver como tratamos los resultados del circuito, aplicando el método de las fracciones continuas [25] para obtener r y no tanto ver como se construye el oráculo.

En concreto, el caso que vamos a resolver es el de $a = 7$ y $N = 15$. Es fácil ver que para este caso tenemos un periodo de $r = 4$, donde los cuatro posibles estados son $|1\rangle$, $|7\rangle$, $|4\rangle$ y $|13\rangle$.

16.3.1. Caso con muchos shots (ineficiente pero didáctico)

Jupyter Notebook: [12. Shor: Implementación \(ad hoc\) en Qiskit para \$N = 15\$](#)

Ver secciones [12.1 - Caso con muchos shots \(ineficiente pero didáctico\)](#) del notebook [12. Shor: Implementación \(ad hoc\) en Qiskit para \$N = 15\$](#) .

El Notebook puede descargarse de [Github](#).

16.3.2. Caso shot a shot (óptimo)

Jupyter Notebook: [12. Shor: Implementación \(ad hoc\) en Qiskit para \$N = 15\$](#)

Ver secciones [12.2. Caso shot a shot \(óptimo\)](#) del notebook [12. Shor: Implementación \(ad hoc\) en Qiskit para \$N = 15\$](#) .

El Notebook puede descargarse de [Github](#).

Capítulo 17

Algoritmo de Shor: Implementación con $2n+3$ qubit

En este capítulo vamos a ver una implementación eficiente del algoritmo de Shor, siguiendo el paper [26], donde lo que se intenta es minimizar el número de qubits lo máximo posible.

Denominamos n al número de qubits que necesitamos para codificar el número N (que queremos factorizar) en un registro cuántico. Para aplicar el algoritmo de Shor se suelen usar $t = 2n$ qubits en el registro de conteo.

17.1. La idea

Esta sección puede ser un poco árida si es la primera vez que se ve esta implementación, pues está enfocada para ser re-leída una vez se ha mirado más a fondo la implementación. Veamos paso a paso como es esta implementación:

1. Algoritmo cuántico de suma (ver sección 17.2.1):

Como vamos a ir viendo en las siguientes secciones, esta implementación se basa en **sumar**. En concreto, se parte de la implementación del algoritmo cuántico de suma de Draper [27], que podemos ver en la Fig. 17.1.

Este algoritmo suma dos registros cuánticos a y $\phi(b)$, donde $|\phi(x)\rangle$ hace referencia a la transformada de Fourier del registro x :

$$|\phi(x)\rangle = QFT|x\rangle. \quad (17.1)$$

Lo que vamos a ver es como, partiendo de esta implementación del algoritmo de suma, podemos construir la exponencial modulada.

2. Valor clásico + registro cuántico, puerta $\phi ADD(a)$ (ver sección 17.2.2):

Nosotros queremos llegar a calcular el periodo de la función $f(x) = a^x \text{ mod } N$, donde a es un valor que fijamos al principio del algoritmo de Shor. Como precisamente este valor es fijo, la primera simplificación para reducir el número de qubits es prescindir de los qubits que codifican a en el algoritmo de suma y tomar a como un valor clásico. Definimos así la puerta $\phi ADD(a)$ (ver Fig. 17.2), que **suma un valor clásico a a la un registro cuántico que codifica el valor b** :

$$\boxed{\phi ADD(a)|\phi(b)\rangle = |\phi(a+b)\rangle}, \quad (17.2)$$

Podemos además, definir la inversa de esta puerta, es decir, una puerta de resta. La acción de esta última se recoge en la Fig. 17.3 (**véase que la puerta de suma tiene la barra negra a la derecha y la de resta a la izquierda**)

3. Suma modulada, puerta $\phi ADD(a)MOD(N)$ (ver sección 17.2.3):

Un vez definida esta puerta, podemos usarla para construir una puerta doble controlada de **suma modulada $\phi ADD(a)MOD(N)$** (ver Fig. 17.4) tal que

$$\boxed{\phi ADD(a)MOD(N)|\phi(b)\rangle = |\phi((a+b) \bmod N)\rangle} = QFT|(a+b) \bmod N\rangle \quad (17.3)$$

4. Multiplicación modulada, puerta $CMULT(a)MOD(N)$ (ver sección 17.2.4):

Ahora, podemos usar esta puerta doble controlada de suma modulada para construir una puerta controlada de **multiplicación modulada $CMULT(a)MOD(N)$** (ver Fig. 17.5) tal que

$$\boxed{CMULT(a)MOD(N)|c\rangle|x\rangle|b\rangle = |c\rangle|x\rangle|(b+ax) \bmod N\rangle, \quad \text{si } c=1} \quad (17.4)$$

5. Puerta controlada $C-U_a$ (ver sección 17.2.5):

Juntando esta puerta con una puerta **SWAP** (pues la multiplicación modulada nos sale en el registro de b , no el de x) y tomando $b = 0$ (una ancilla), podemos finalmente construir una puerta controlada $C-U_a$ (ver Fig. 17.6) tal que :

$$\boxed{C-U_a|c\rangle|x\rangle = |c\rangle|(a \cdot x) \bmod N\rangle, \quad \text{si } c=1} \quad (17.5)$$

6. Exponencial modulada, puerta $C-U_{a^s}$ (ver sección 17.2.6):

Ahora, para implementar el algoritmo de Shor lo que se hace es tomar $|x\rangle = |1\rangle$ y implementar puertas $C-U_{a^s}$, donde $s = 2^0, 2^1, \dots, 2^{2n-1}$, pues puede verse que

$$\boxed{C-U_{a^s} = (C-U_a)^s} \quad (17.6)$$

7. Círculo final con $4n + 2$ qubits, sin la simplificación del registro de conteo (ver sección 17.2.7):

El circuito para la implementación del algoritmo de Shor sería el de la Fig. 17.7.

8. Círculo final con $2n + 3$. El truco de un qubit de control (ver sección 17.2.8):

Nos faltaría implementar la simplificación del registro de conteo, donde se pasa de $2n$ qubits en el mismo a 1, llegando a tener el circuito de la Fig. 17.11.

17.2. Explicación desgranada

En las siguientes subsecciones vamos ir explicando poco a poco los pasos mencionados en la sección 17.1.

17.2.1. Algoritmo cuántico de suma

Como vamos a ir viendo en las siguientes secciones, esta implementación se basa en **sumar**. En concreto, se parte de la implementación del **algoritmo cuántico de suma de Draper [27]**, que podemos ver en la Fig. 17.1

Conditional Phase Shift

$$\begin{array}{c} \text{Caja } k \\ \text{y} \\ \text{Caja } k \end{array} = \begin{array}{c} \bullet \\ \text{y} \\ \text{Caja } k \end{array} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{\frac{2\pi i}{2^k}} \end{bmatrix}$$

Addition Transform

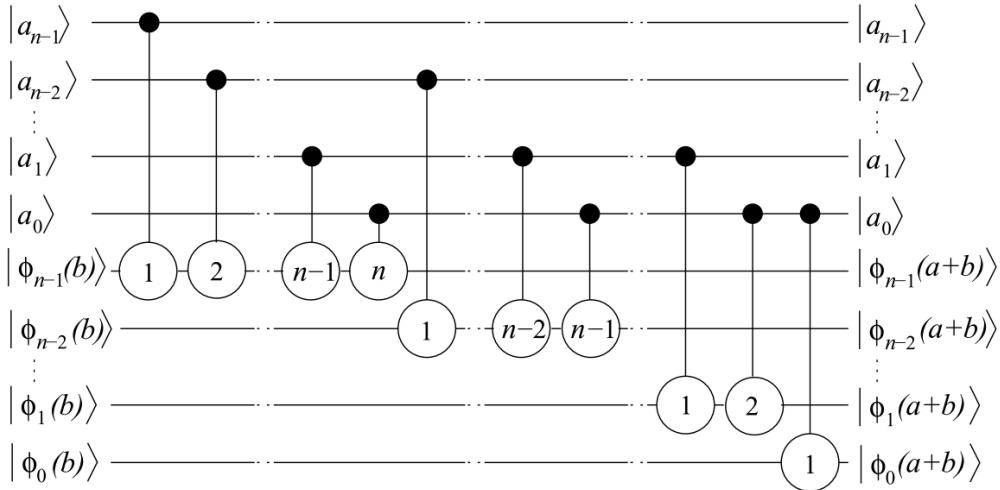


Figura 17.1: Algoritmo cuántico de suma de Draper (en el convenio **estándar**). Figura tomada de [26]

Nota importante

Véase que la puerta “Conditional Phase Shif” de la Fig. 17.1 es la puerta CR_k de la Ec. (14.19). Véase también que estas puertas no son más que puertas $P(\phi)$ (o P_ϕ) controladas con $\phi_k = 2\pi i/2^k$.

Este algoritmo suma los valores a y b . Las entradas del circuito de suma son el **n qúbits** representando el número a y **n qúbits** que contienen la transformada de Fourier de otro número b , denotada como $\phi(b)$, es decir,

$$|\phi(b)\rangle = QFT|b\rangle. \quad (17.7)$$

El registro que codifica el número a no cambia, mientras que registro que codifica $\phi(b)$ pasa a albergar la suma de $a+b$ en el espacio de Fourier, $\phi(a+b)$. Haciendo la transformada inversa se puede recuperar el valor $a+b$:

$$QFT^{-1}|\phi(a+b)\rangle = |a+b\rangle \quad (17.8)$$

Lo que vamos a ver es como, partiendo de esta implementación del algoritmo de suma podemos construir la exponencial modulada.

Nota

No perdamos el objetivo de vista. Nosotros queremos calcular el periodo de la función $f(x) = a^x \bmod N$, donde $a < N$ (tomamos además $b < N$). Al tener un $\bmod N$ sabemos que el esta función no puede devolver valores mayores que N . Tenemos pues que los **n qúbits** que dijimos que usamos para codificar a y b son el número de qúbits que nos hace falta para codificar N .

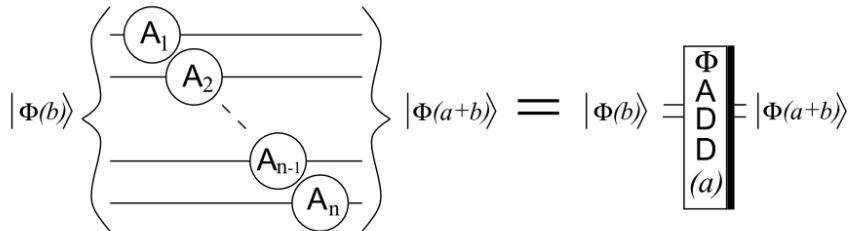


Figura 17.2: Puerta $\phi ADD(a)$. Figura tomada de [26]

17.2.2. Valor clásico + registro cuántico (puerta $\phi ADD(a)$)

Nosotros queremos llegar a calcular el periodo de la función $f(x) = a^x \bmod N$, donde a es un valor fijo menor que N .

Como precisamente este valor es fijo, no nos vemos en la necesidad de codificarlo usando un registro cuántico. Podemos pues sustituir los qubits que codifican a por bits clásicos. Las puertas controladas pasan entonces a ser puertas controladas clásicamente. Además, como sabemos de antemano el valor de a , podemos precalcular el producto de las puertas sobre cada qubit (sumando las fases), aplicando así solo una puerta por qubit (reducimos la profundidad del circuito).

Definimos así la puerta $\phi ADD(a)$ (ver Fig. 17.2), que **suma un valor clásico a a un registro cuántico que codifica el valor $\phi(b)$** . La entrada de esta puerta es la transformada de Fourier del registro b , es decir, $\phi(b)$, y la salida es la transformada de Fourier de la suma, $\phi(a + b)$:

$$\boxed{\phi ADD(a) |\phi(b)\rangle = |\phi(a + b)\rangle}, \quad (17.9)$$

Comentamos antes que el número de qubits n que usamos para codificar a y b es número de qubits que nos hacen falta para codificar N (ya que $a, b < N$). Puede darse el caso de que al hacer la suma tengamos un valor mayor que N , es decir, $a + b > N$. Podríamos tener entonces un número mayor que el número más grande que podemos codificar con los n qubits de los que partimos. Esto se denomina **overflow**. Para evitar esto, lo que podemos hacer es añadir un qubit extra al registro que contiene $\phi(b)$. Tenemos pues que **$\phi(b)$ es de forma efectiva la transformada de Fourier de un registro de $n+1$ qubits que contiene un número de n bits**. De esta forma, antes de la suma, el bit más significativo de la transformada de Fourier inversa del registro $\phi(b)$ es siempre $|0\rangle$:

$$\boxed{\text{el bit más significativo de } QFT^{-1}|\phi(b)\rangle = |b\rangle \text{ es siempre } |0\rangle} \quad (17.10)$$

Podemos además, definir la inversa de esta puerta, es decir, una puerta de resta. La acción de esta última se recoge en la Fig. 17.3 (**véase que la puerta de suma tiene la barra negra a la derecha y la de resta a la izquierda**). En esta figura p es un número n bit y g un número $n+1$ bit.

Véase que tenemos resultados diferentes si $g \geq p$ o $g < p$. Precisamente, usando (17.10) podemos usar esto para saber cual de los dos números es mayor:

- Si después de la resta el bit más significativo es $|0\rangle$, estamos en el caso de $g \geq p$ (solo si $g - p$ es n bit).
- Si después de la resta el bit más significativo es $|1\rangle$, estamos en el caso de $p > g$.

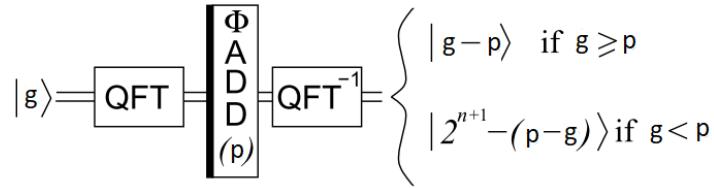


Figura 17.3: Puerta $\phi ADD^{-1}(a)$, es decir, la puerta inversa de $\phi ADD(a)$, una puerta de resta. Figura tomada de [26]

Nota

Aplicaremos este método con:

- $p = N$ y $g = b, a + b$ (parte 1 en la Fig. 17.4)
- $p = a$ y $g = b, a + b, a + b - N$ (parte 2 en la Fig. 17.4)

En ambos casos tenemos que p es un número n bit y si $g \geq p$ tenemos $g - p < N$, así que $g - p$ es n bit. Podemos pues aplicar el criterio del bit más significativo para saber qué número es más grande.

Nota: $\phi ADD(p)$ y $\phi ADD^{-1}(p)$ son una la inversa de la otra

Si estamos en el caso $g < p$ tenemos

$$\phi ADD^{-1}(p) |\phi(g)\rangle = |\phi(2^{n+1} - (p - g))\rangle \quad (17.11)$$

Si ahora sumamos otra vez p tenemos:

$$\phi ADD(p) |\phi(2^{n+1} - (p - g))\rangle = |\phi(2^{n+1} - (p - g)) + p\rangle = |\phi(2^{n+1} + g)\rangle \quad (17.12)$$

Como $2^{n+1} + g$ es mayor que el valor máximo que podemos almacenar con $n + 1$ qubits, tenemos overflow:

$$\phi ADD(p) |\phi(2^{n+1} - (p - g))\rangle = |\phi(2^{n+1} + g)\rangle = |\phi(g)\rangle \quad (17.13)$$

Con lo cual, efectivamente, $\phi ADD(p)$ y $\phi ADD^{-1}(p)$ son una la inversa de la otra.

17.2.3. Suma modulada (puerta $\phi ADD(a)MOD(N)$)

Un vez definida la puerta de suma $\phi ADD(a)$, podemos usarla para construir una puerta de **suma modulada $\phi ADD(a)MOD(N)$** (ver Fig. 17.4). Esta puerta suma $a + b$ y le resta N si $a + b \geq N$. Las entradas de la misma son $\phi(b)$ con $b < N$ y un valor clásico $a < N$.

$$\boxed{\phi ADD(a)MOD(N)|\phi(b)\rangle = |\phi((a + b) \bmod N)\rangle} = QFT |(a + b) \bmod N\rangle \quad (17.14)$$

En la Fig. 17.4 vemos que se han añadido dos qubits de control ($|c_1\rangle, |c_2\rangle$) para futuros usos. La puerta solo se activa si $c_1 = c_2 = 1$. Se han numerado las puertas (cuadros rojos) para facilitar la explicación.

Como podemos ver en la imagen, el circuito que implementa esta puerta tiene dos partes:

- Parte 1: Calcula $(a + b) \bmod N$.
- Parte 2: Vuelve a poner en el estado $|0\rangle$ el qubit ancila (el qubit de abajo del todo en la imagen).

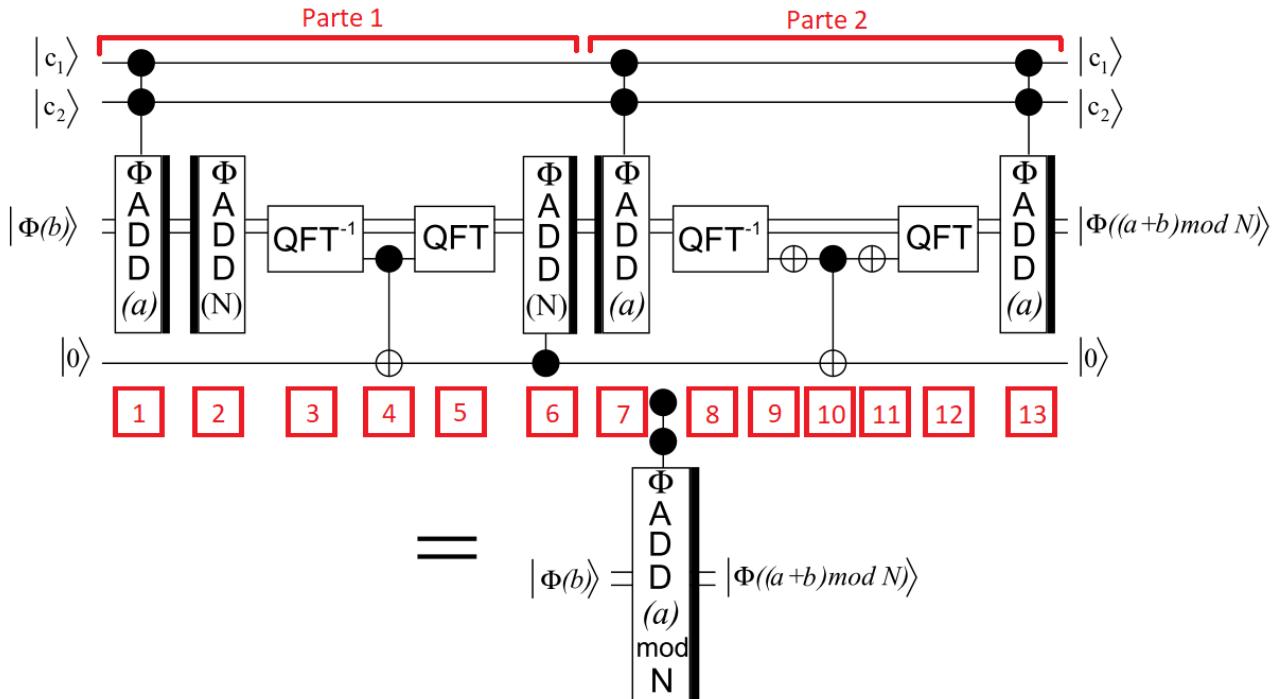


Figura 17.4: Puerta $\phi ADD(a)MOD(N)$ (en el convenio de **Qiskit**). Figura tomada de [26].

Nota

Un qúbit **ancila** es un qúbit auxiliar que se usa para hacer un calculo intermedio pero que no forma parte de la solucion. Estos qúbit hay que volver a ponerlos en el estado inicial.

Nota importante

El qúbit ancila **no es qúbit extra del registro** $|\phi(b)\rangle$. Tenemos pues los 2 qúbits de control, los $n + 1$ de la entrada $|\phi(b)\rangle$ y el qúbit de la ancila, con lo cual usamos $n + 4$ qúbits.

Veamos todos los posibles casos que nos podemos encontrar al aplicar el circuito de la Fig 17.4.

17.2.3.1. Caso con $c_1 = c_2 = 1$

Partimos del estado $|\phi(b)\rangle$.

- **Puerta 1:** Despues de aplicar la puerta 1 (puerta $\phi ADD(a)$) tenemos el estado $|\phi(a + b)\rangle$.
- **Puertas 2, 3, 4 y 5:** Lo que vamos a hacer con estas puestas es ver si estamos en el caso de $a + b \geq N$ o $a + b < N$. Para ello, aplicamos una puerta $\phi ADD^{-1}(N)$ (puerta 2), con lo cual tenemos:
 1. El estado $|\phi(a + b - N)\rangle$ si $a + b \geq N$.
 2. El estado $|\phi(2^{n+1} - (a + b - N))\rangle$ si $a + b < N$.

Las puertas 3, 4 y 5 lo que hacen es poner el qúbit ancila a 1 si el bit más significativo es 1, es decir, si estamos en el segundo caso.

- **Puerta 6:** Esta puerta solo se activa si la ancila es 1, es decir, si estamos en el segundo caso del paso anterior ($a + b < N$). Esta puerta lo que hace es sumar N , deshaciendo la resta de la

puerta 2 si estamos en el caso $a + b < N$, es decir, si no teníamos que haber restado N . Después de esta puerta ya tenemos el resultado de la suma modulada. Solo nos queda limpiar la ancila.

- **Puerta 7:** Partimos del valor $(a + b) \bmod N$. Con esta puerta restamos a , con lo cual tenemos dos casos.

$$\begin{aligned} 3. \text{ Si } (a + b) > N &\Rightarrow (a + b) \bmod N = a + b - N < a \Rightarrow \\ &\Rightarrow \phi ADD^{-1}(a) |\phi((a + b) \bmod N)\rangle = |\phi(2^{n+1} - (N - b))\rangle \\ 4. \text{ Si } (a + b) < N &\Rightarrow (a + b) \bmod N = a + b > a \Rightarrow \\ &\Rightarrow \phi ADD^{-1}(a) |\phi((a + b) \bmod N)\rangle = |\phi(b)\rangle \end{aligned}$$

- **Puertas 8, 9, 10, 11, 12:** Estas puertas lo que hacen es cambiar la ancila si el bit más significativo es cero. Vemos que este es el caso 4 del paso anterior, es decir, cuando $(a + b) < N$. Si nos fijamos en cuando aplicamos las puertas 2, 3, 4 y 5, esto corresponde al caso 2, justo aquel en el que cambiamos la ancila. Es decir, si hubiéramos cambiado la ancila, ahora la habríamos vuelto a poner a cero.

- **Puerta 13:** Deshace el cambio producido por la puerta 7.

17.2.3.2. Caso con $c_1 = 0$ y/o $c_2 = 0$.

En estos casos la puerta $\phi ADD(a)MOD(N)$ deja invariante la entrada. Partimos del estado $|\phi(b)\rangle$.

- **Puerta 1:** No se aplica.
- **Puerta 2:** Restamos N , con lo cual pasamos a tener $\phi(2^{n+1} - (N - b))$.
- **Puertas 2, 3, 4 y 5:** Como el bit más significativo en este caso es siempre 1, estas puertas ponen la ancila a 1.
- **Puerta 6:** Como la ancila es 1, esta puerta se activa y deshace los cambios de la puerta 2. Pasamos pues a tener el estado inicial $|\phi(b)\rangle$ pero con la ancila a 1.
- **Puerta 7:** No se aplica.
- **Puertas 8, 9, 10, 11, 12:** Como comentamos antes, estas puertas cambian la ancila si el estado que entra tiene el bit más significativo a cero. Como tenemos el estado $|\phi(b)\rangle$, este es nuestro caso así que se vuelve a cambiar la ancila. Pasamos pues a tener el estado inicial, sin ningún cambio.
- **Puerta 13:** No se aplica.

17.2.4. Multiplicación modulada (puerta $CMULT(a)MOD(N)$)

El siguiente paso es usar la puerta $\phi ADD(a)MOD(N)$ para construir una puerta controlada de multiplicación modulada que denominaremos como $CMULT(a)MOD(N)$. La entrada de esta puerta serán tres registros $|c\rangle|x\rangle|b\rangle$, donde $|c\rangle$ es un qubit controlador:

$$CMULT(a)MOD(N) |c\rangle|x\rangle|b\rangle = |c\rangle|x\rangle|(b + ax) \bmod N\rangle, \quad \text{si } c = 1 \quad (17.15)$$

$$CMULT(a)MOD(N) |c\rangle|x\rangle|b\rangle = |c\rangle|x\rangle|b\rangle, \quad \text{si } c = 0 \quad (17.16)$$

Para implementar esta puerta recurrimos a las puertas $\phi ADD(a)MOD(N)$ de la sección anterior y a

la identidad

$$(ax) \bmod N = (2^0 ax_0 + 2^1 ax_1 + \cdots + 2^{n-1} ax_{n-1}) \bmod N$$

$$= \left\{ \dots \left[(2^0 ax_0) \bmod N + 2^1 ax_1 \right] \bmod N + \cdots + 2^{n-1} ax_{n-1} \right\} \bmod N$$

Es fácil entender esta identidad: es lo mismo sumar todos los términos y finalmente tomar el módulo de la suma completa que tomar el módulo del primer término, sumárselo al segundo término, volver a tomar el módulo, etc.

Vemos que esto se puede implementar aplicando:

- primero una puerta $\phi ADD(2^0 a) MOD(N)$ sobre $|b\rangle$ controlada por $|c\rangle$ y $|x_0\rangle$,
- después una puerta $\phi ADD(2^1 a) MOD(N)$ sobre el resultado de la anterior controlada por $|c\rangle$ y $|x_1\rangle$, etc.

Con lo cual, **solo necesitamos aplicar n puertas doblemente controladas $\phi ADD(2^i a) MOD(N)$** con $0 \leq i < N$. Podemos ver esta implementación en la Fig. 17.5.

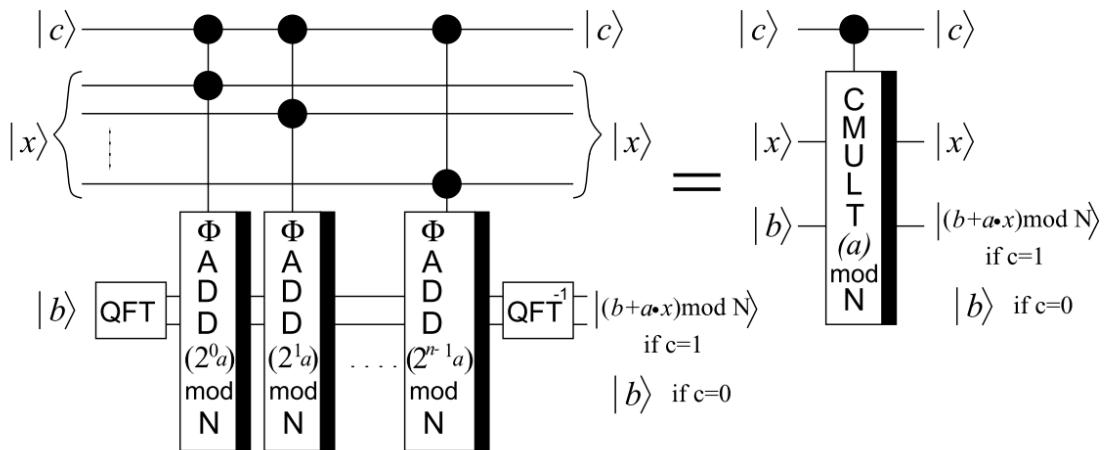


Figura 17.5: Puerta $CMULT(a)MOD(N)$ (en el convenio de **Qiskit**). Figura tomada de [26].

17.2.5. Puerta controlada $C-U_a$

En la sección anterior vimos como construir una puerta controlada que aplica la operación

$$|x\rangle|b\rangle \rightarrow |x\rangle|(b + ax) \bmod N\rangle \quad (17.17)$$

Pero esto no es lo que queremos, nosotros queremos una puerta controlada que nos lleve el estado $|x\rangle$ al estado $|(ax) \bmod N\rangle$. Lo que podemos hacer para solventar esto es lo siguiente:

- Aplicar primero una puerta $CMULT(a)MOD(N)$ sobre el estado $|c\rangle|x\rangle|0\rangle$, con lo cual obtenemos el estado $|c\rangle|x\rangle|(ax) \bmod N\rangle$.
- A continuación, si $|c\rangle = |1\rangle$ aplicamos puertas SWAP controladas para cambiar los registros $|x\rangle$ y $|(b + ax) \bmod N\rangle$, con lo cual pasamos a tener el estado $|c\rangle|(ax) \bmod N\rangle|x\rangle$. Solo necesitamos aplicar puertas controladas SWAP a n qubits, no a $n + 1$, ya que el qubit más significativo de $(ax) \bmod N$ es siempre 0, ya que es el qubit extra que incluimos para evitar overflow en las puertas $\phi ADD(a)$.

- Finalmente, aplicamos la inversa de la puerta controlada $C\text{MULT}(a^{-1})\text{MOD}(N)$ (donde a^{-1} es el inverso de $a \bmod N$), para poner el tercer registro a $|0\rangle$ otra vez. Este valor se calcula clásicamente en tiempo polinómico usando el algoritmo de Euclides y tenemos asegurado que siempre existe ya que $\gcd(a, N) = 1$. En resumen, si la entrada de esta puerta es el estado $|c\rangle|x\rangle|b\rangle$ con $|c\rangle = |1\rangle$ tenemos:

$$[C\text{MULT}(a^{-1})\text{MOD}(N)]^{-1}|c\rangle|x\rangle|b\rangle = |c\rangle|x\rangle|(b - a^{-1}x)\rangle \quad (17.18)$$

En nuestro caso tenemos que el estado de entrada es $|c\rangle|(ax) \bmod N\rangle|x\rangle$, con lo cual:

$$\begin{aligned} [C\text{MULT}(a^{-1})\text{MOD}(N)]^{-1}|c\rangle|(ax) \bmod N\rangle|x\rangle &= |c\rangle|(ax) \bmod N\rangle|(x - a^{-1}ax) \bmod N\rangle = \\ &= |c\rangle|(ax) \bmod N\rangle|0\rangle \end{aligned}$$

Denominaremos al conjunto de aplicar estas tres puertas controladas la **puerta controlada U_a** , es decir, $C\text{-}U_a$. En resumen, esta puerta lo que hace es tomar como entrada $|c\rangle|x\rangle|0\rangle$ y devolver $|c\rangle|(ax) \bmod N\rangle|0\rangle$ si $c = 1$:

$$C\text{-}U_a|c\rangle|x\rangle|0\rangle = |c\rangle|(ax) \bmod N\rangle|0\rangle \quad \text{si } c = 1 \quad (17.19)$$

Si $c = 0$, aplica la identidad.

Para resumir, si $|c\rangle = |1\rangle$ los pasos que hace la puerta $C\text{-}U_a$ son los siguientes (ver Fig. 17.6)

$$\begin{aligned} |x\rangle|0\rangle &\rightarrow |x\rangle|(ax) \bmod N\rangle \rightarrow |(ax) \bmod N\rangle|x\rangle \rightarrow \\ &\rightarrow |(ax) \bmod N\rangle|(x - a^{-1}ax) \bmod N\rangle = |(ax) \bmod N\rangle|0\rangle \end{aligned}$$

Si nos fijamos, el último registro, al estar a $|0\rangle$ al inicio y al final, podemos considerarlo parte de la puerta $C\text{-}U_a$ (una ancila):

$$C\text{-}U_a|x\rangle|0\rangle = |(ax) \bmod N\rangle|0\rangle \Rightarrow C\text{-}U_a|x\rangle = |(ax) \bmod N\rangle \quad (17.20)$$

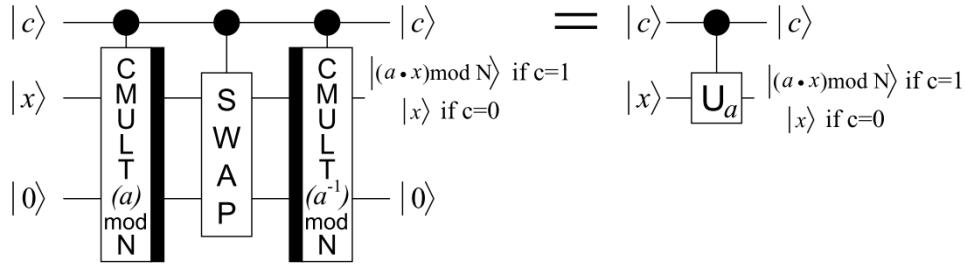


Figura 17.6: Puerta $C\text{-}U_a$

17.2.6. Exponencial modulada (puerta $C\text{-}U_{a^s}$)

Una vez construida la puerta $C\text{-}U_a$ uno podría pensar que para aplicar la exponencial modulada lo que hay que hacer es aplicar varias veces esta puerta, es decir:

$$(C\text{-}U_a)^s|x\rangle = |(a^s x) \bmod N\rangle \quad (17.21)$$

Aunque esta implementación es posible, tenemos la opción de hacer una mucho más optima. Para ello nos servimos de la propiedad

$$(a^s x) \bmod N = \underbrace{\{ \dots [a(ax) \bmod N] \bmod N \dots \}}_{s \text{ veces}} \bmod N = [x(a^s) \bmod N] \bmod N \quad (17.22)$$

En vez de aplicar s veces la puerta $C-U_a$ podemos aplicar una sola vez la puerta $C-U_{a^s}$ donde el subíndice a^s hace referencia que le pasamos a la puerta el valor $a^s \bmod N$ (este se calcula clásicamente)

$$C-U_{a^s} = (C-U_a)^s \quad (17.23)$$

17.2.7. Circuito final con $4n+2$ qúbits (sin la simplificación del registro de conteo)

Solo nos queda ver el circuito con la implementación completa del algoritmo de Shor, representado en la Fig. 17.7. (Recordemos que n es el número de qúbits que necesitamos para codificar n y que en el registro de conteo necesitamos $2n$ qúbits.)

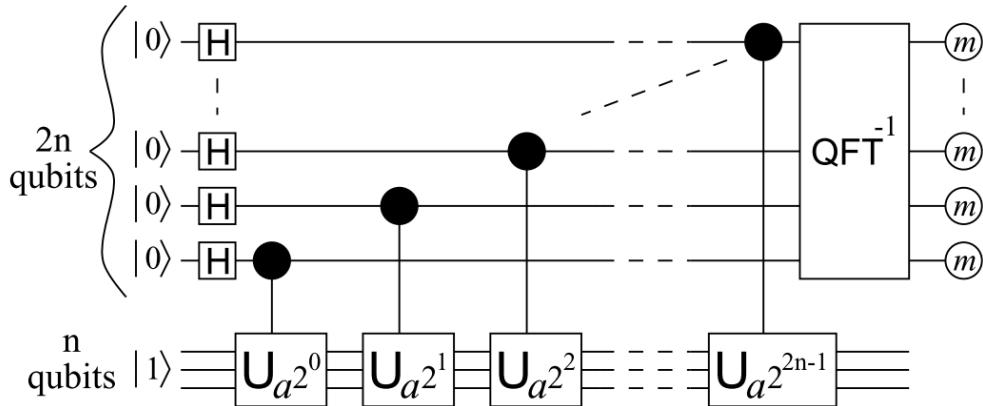


Figura 17.7: Circuito final con $4n+2$ qúbits sin la simplificación del registro de conteo (en el convenio estándar).

Esta implementación usa $4n + 2$ qúbits:

- $2n$ qúbits en el registro de conteo
- n qúbits para el estado $|1\rangle$ (este es, el estado $|x\rangle$ de las secciones 17.2.4 a la 17.2.5).
- $n + 2$ qúbits para las ancillas:
 - $n + 1$ qúbits para el estado $|b\rangle = |0\rangle$ (ver Figs. 17.5 y 17.6).
 - 1 qúbit para la ancila de la puerta $\phi ADD(a)MOD(N)$ (ver Fig. 17.4).

17.2.8. Circuito final con $2n + 3$. Algoritmo de estimación iterativa de fase (IPE)

Nos faltaría implementar la simplificación del registro de conteo, donde se pasa de $2n$ qúbits en el mismo a 1 (ver Fig. 17.11). Esta simplificación consiste en usar una versión mejorada del Algoritmo de Estimación de Fases Cuántico (QPE) denominado **Algoritmo de Estimación Iterativa de Fase** o Iterative Phase Estimation (IPE) Algorithm. Vamos a explicar como pasar del QPE al IPE en nuestro circuito. Para ello, vamos a exemplificar el desarrollo con el circuito de 4 qúbits en el registro de conteo de la Fig. 17.8. Las puertas de colores del final no son más que la transformada de Fourier inversa.

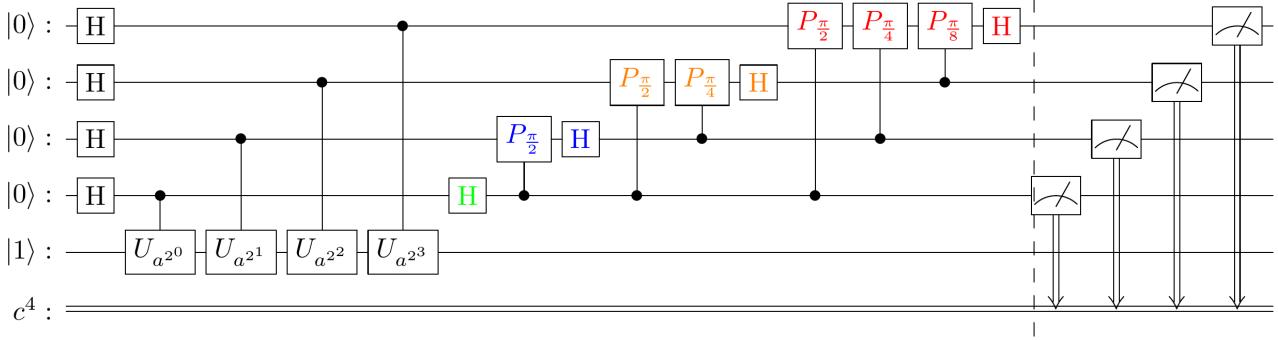


Figura 17.8: Ejemplo del algoritmo de Shor con 4 qúbits (en el convenio estándar). Las puertas de colores del final no son más que la transformada de Fourier inversa.

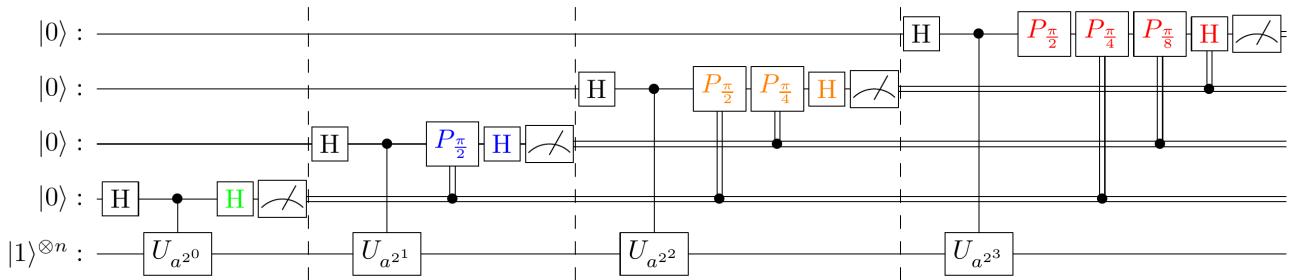


Figura 17.9: Ejemplo del algoritmo de Shor con 4 qúbits usando IPE (en el convenio estándar). Las puertas de colores no son más que la transformada de Fourier inversa.

Vemos que al ser la inversa lo que hay que hacer es invertir el orden de las puertas del QFT normal (ver Fig. 17.12 y Fig. 14.2).

Si nos fijamos, vemos que una vez que un qúbit controla su respectiva puerta $U_{a^{2^k}}$ sobre este ya no aplican (ni controla) más puertas hasta llegar a las de la QFT^{-1} . Podemos pues llevar a cabo sin ningún problema la reordenación de las puertas de color de la Fig. 17.9. Vemos sin embargo que hay otros dos cambios significativos:

- En vez de poner los 4 bit clásicos (en los que se almacenan las medidas) en una linea a parte, se han puesto a continuación del medidor. Esto es simplemente para no añadir 4 líneas más al circuito.
- El gran cambio que introducimos en este circuito es el hecho de controlar puertas con bits clásicos.

En el circuito de la Fig. 17.9 ya se han colocado las puertas para que se vea bien que estas se pueden aplicar de forma secuencial qúbit por qúbit. Es decir, primero se aplican las puertas sobre el qúbit de abajo del todo (en el registro de conteo) y se mide. Después se va a por siguiente qúbit y se mide, y así sucesivamente. La gracia es que, una vez que se mide un qúbit, el valor de esta medida se va usar para controlar puertas que se aplican en los siguientes qúbit. Lo importante es que, como ya comentamos, estas medidas se almacenan en bits clásicos, con lo cual, una vez que se ha medido un qúbit, este deja de ser necesario. De la misma forma, al ir aplicando las puertas de forma secuencial qúbit por qúbit, mientras se aplican las puertas a los qúbits anteriores, los qúbits siguientes también son “inutiles”.

Siguiendo estos argumentos, podemos ver que en realidad, solo nos hace falta un qúbit en el registro de conteo. Esto es lo que podemos ver en la Fig. 17.10. Como vemos, tenemos solo un qúbit en el registro de conteo.

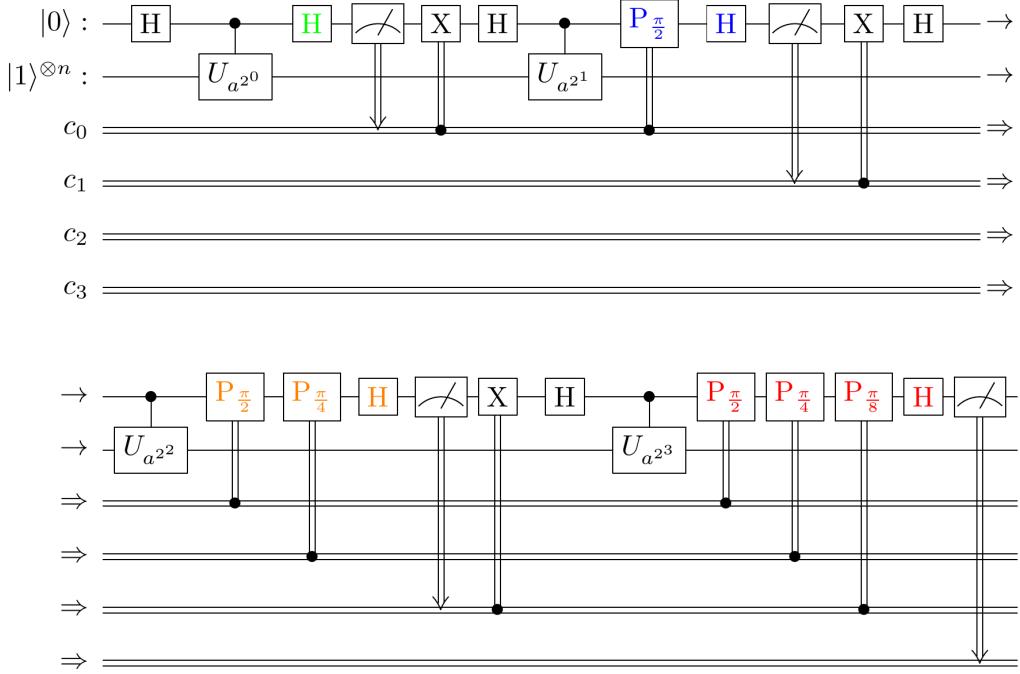


Figura 17.10: Circuito final con $2n + 3$ qúbits (ejemplo con 4 qúbits).

- Lo que se hace es primero aplicar sobre este qúbit las puertas que aplicaríamos sobre el último qúbit del registro de conteo y después medir. Almacenamos esta medida en un bit clásico. Una vez medido, podemos usar el valor del bit clásico para controlar una puerta X . De esta forma, lo que hacemos es devolver el qúbit al estado inicial (el estado $|0\rangle$).
- Una vez que volvemos a tener el qúbit en su estado de partida y la media del mismo a buen recaudo, podemos pasar a aplicar sobre este qúbit las puertas que aplicaríamos sobre el siguiente qúbit del registro de conteo (una de ellas controlada por el bit clásico anterior) y medirlo, almacenando su valor en un segundo bit clásico. Nuevamente, usamos una puerta X controlada por este segundo bit clásico para devolver el qúbit al estado inicial.

Y así, sucesivamente. Para más detalles, pueden verse las referencias [28], [29] y [30]

Extrapolando al caso de $2n$ qúbits en el registro de conteo (el caso del algoritmo de Shor), nuestro circuito final sería el de la Fig. 17.11, donde las m_k se refiere a medidas, las X^{m_k} se refiere a aplicar la puerta X controlada por las medida anterior y las R_k se refiere a aplicar las puertas P_ϕ de la QFT controladas por las medias anteriores.

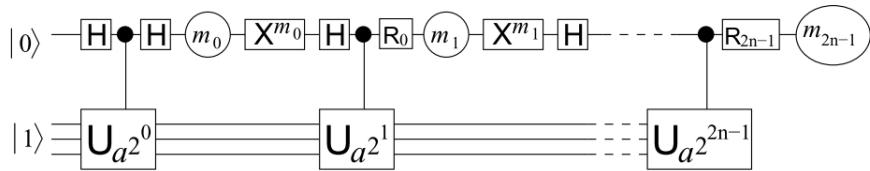


Figura 17.11: Circuito final con $2n + 3$ qúbits (en el convenio estándar).

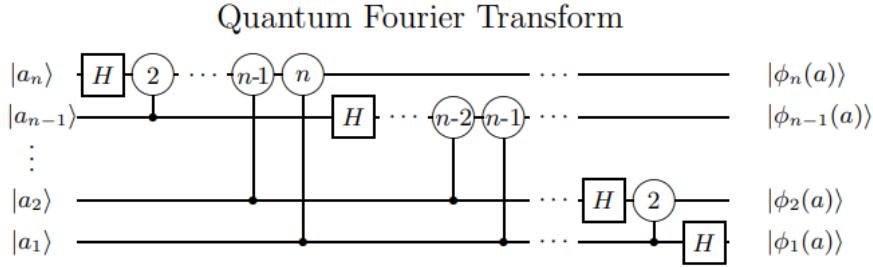


Figura 17.12: Implementación de la trasformada de Fourier (exacta).

17.3. Implementación aproximada de la QFT

Ya vimos en el capítulo 14.2 la implementación de la trasformada de Fourier cuántica. En la Fig. 17.12 presentamos otra vez la misma implementación pero con la notación de puertas que estuvimos usando en esta sección.

Esta es la implementación exacta de la QFT que, como vemos la Fig. 17.12, cuando la aplicamos a n qubits necesitamos del orden de $\mathcal{O}(n^2)$ operaciones/puertas (en concreto, son $\frac{1}{2}n(n+1)$ operaciones).

Se puede apreciar que cuando k crece mucho, las puertas $CROT_k$ se aproximan a la Identidad. Esto hace que podamos prescindir de las puertas con un valor k mayor que un cierto umbral k_{max} y aplicar así una versión **aproximada** (con menos operaciones) de la trasformada de Fourier. Puede demostrarse (ver [22]) que el error introducido por ignorar todas las puertas con un valor $k > k_{max}$ es proporcional a $n2^{-k_{max}}$. Podemos pues tomar k_{max} del orden de $\mathcal{O}(\log_2 n)$, pasando de tener del orden de $\mathcal{O}(n^2)$ operaciones (puertas) a $\mathcal{O}(n \log_2(n))$ operaciones [en concreto $\frac{1}{2}(2n - \log_2 n)(\log_2 n - 1)$ operaciones].

17.4. Implementación de las SWAP controladas

En la Fig. 17.13 podemos ver la puerta *SWAP* controlada. Esta es una puerta que admite tres qubits (uno de control y dos de operación). Tenemos dos puertas *CNOT* que rodean a una puerta *Toffoli*. La función de esta puerta es intercambiar el valor de los dos qubits de operación si el qubit de control está activado. Como vemos, para intercambiar dos qubits necesitamos 3 puertas, con lo cual las puertas necesarias para aplicar una puerta *SWAP* controlada a n qubit necesitamos $\mathcal{O}(n)$ puertas.

Vemos que la puertas *SWAP* controlada no es las que la puerta *SWAP* normal (tres puertas *CNOT*) donde la puerta *CNOT* central se controla, convirtiéndola en una puerta *Toffoli*.

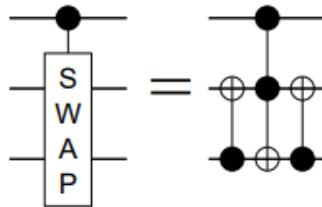


Figura 17.13: Puerta SWAP

Capítulo 18

Algoritmo de Grover (Amplificación de amplitud)

Diccionario de notaciones

Notación	Explicación
$\mathcal{O}(z)$	Esta notación quiere decir “del orden de z ”.
N	Número de elemento del dataset en el que buscamos
n	Número de qubits que nos hacen falta
M	Número de soluciones que buscamos en el dataset de N elementos
t	Número de iteraciones del algoritmo de Grover
T	Número óptimo de iteraciones del algoritmo de Grover
$\lfloor z \rfloor$	Redondear z a un entero por truncamiento
$\lceil z \rceil$	Redondear z al siguiente entero.

18.1. Introducción

El algoritmo sobre el que trata este capítulo, el conocido **Algoritmo de Grover**, no ofrece una ventaja exponencial respecto al mejor algoritmo clásico (como es el caso del algoritmo de Shor), pero si ofrece una mejora sustancial (cuadrática).

Resumiendo mucho, el algoritmo de Grover es un algoritmo de búsqueda no estructurada. El objetivo del algoritmo es encontrar un valor (o varios) en una secuencia no ordenada de N componentes. Supongamos que tenemos una lista L con N elementos donde denominamos L_i , con $i = 0, 1, \dots, N-1$, al i -ésido elemento de la lista. Supongamos que queremos encontrar un elemento q en la lista, es decir, lo que queremos encontrar es el índice ω tal que $L_\omega = q$. Si asumimos que la lista está desordenada, ningún algoritmo clásico conocido puede darnos un tiempo de búsqueda mejor que $\mathcal{O}(N)$. Es decir, el tiempo promedio de búsqueda crece linealmente con N , con el número de elementos entre los que buscamos. Esto es fácil de ver, pues para encontrar nuestro elemento tendremos que ir probando uno a uno los elementos de la lista. De esta forma en promedio tendremos que hacer $N/2$ pruebas para encontrar el resultado deseado.

Este tipo de problemas se denominan problemas de **búsqueda no estructurada** o problemas de búsqueda en **conjuntos de datos (datasets) no estructurados**. Como acabamos de comentar, estos problemas requieren un tiempo polinómico (en concreto, lineal) para ser resueltos. Se engloban dentro de esta categoría también problemas que presenten un dataset con alguna clase de estructura siempre que esta no se pueda aprovechar para acelerar la búsqueda.

La ventaja que aporta el algoritmo de Grover es **cuadrática**, ya que pasamos de necesitar un tiempo $\mathcal{O}(N)$ a un tiempo $\mathcal{O}(\sqrt{N})$. Esto puede parecer decepcionante si lo comparamos con la mejora exponencial que promete el algoritmo de Shor, pero está lejos de serlo. Aunque a priori no parezca una mejora sustancial, cuando tratamos con valores suficientemente grandes de N (datasets inmenso), la mejora en tiempo respecto a su contrapartida clásica puede ser de ordenes de magnitud, es decir, para nada despreciable.

El algoritmo de Grover no hace uso de la estructura interna del dataset en el que realiza la búsqueda, lo cual lo hace genérico y aplicable a una gran variedad de casos. Los problemas de búsqueda aparecen por doquier en las ciencias computacionales, con lo que una mejora en la eficiencia de estos es de gran interés. Además, este algoritmo no solo nos sirve para búsquedas, sino que nos sirve como subrutina para conseguir un aumento de velocidad cuadrático en otros algoritmo. A esto último se lo denomina el truco de la **amplificación de la amplitud**.

En este capítulo vamos a ver todo lo necesario para entender el algoritmo de Grover, hablando tanto de matemática como de implementaciones. La estructura de las notas es la siguiente. En la sección 18.2 veremos una explicación geométrica del algoritmo, donde se usará como ejemplo el caso más simple (una solución, $N = 2^n$ y distribución uniforme) y se verán las diferentes partes del algoritmo.

En las siguientes secciones iremos generalizando el algoritmo a medida que relajamos las condiciones del ejemplo anterior. En la sección 18.3 veremos que pasa cuando tenemos un número M de soluciones (conocido M). En la sección 18.4 relajaremos la condición de conocer el número M de soluciones y veremos que el algoritmo sigue siendo eficiente. Como continuación lógica de la sección anterior, en la sección 18.5 veremos un algoritmo inspirado en el algoritmo de Shor que usa el operador de Grover para contar el número de soluciones (obtener M). En la sección 18.6 hablaremos sobre la implementación del difusor y como la formulación más habitual del mismo es la responsable de imponer la condición $N = 2^n$. Veremos en la subsección 18.6.1 como podemos eliminar también esta condición. Por último, en la sección 18.7 veremos que también es posible relajar la condición de partir de una distribución de probabilidad uniforme.

18.2. Explicación geométrica del algoritmo

Para facilitar la explicación y sin perdida de generalidad, pongámonos en el caso más simple de todos: queremos encontrar una única solución $|\omega_0\rangle$ y tenemos $N = 2^n$. Supongamos también que podemos partir del caso más simple, de una superposición uniforme de todos los estados. Más adelante iremos viendo generalizaciones del algoritmo para casos más complicados, como el caso de varias soluciones o el caso en el que partimos de una distribución de probabilidad aleatoria.

18.2.1. Estado inicial: superposición

Partimos de una superposición uniforme de N estados

$$|\Psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^N |i\rangle. \quad (18.1)$$

Esto no es más que decir que al inicio de la búsqueda, todas las opciones son igualmente probables (cualquier suposición de la ubicación de la solución es tan buena como cualquier otra).

Nota

Recordemos que si $N = 2^n$, donde n es el número de qubits, este estado es fácil de construir, pues solo tenemos que aplicar puertas de Hadamard en todos los qubits (partiendo estos del

estado $|0\rangle$):

$$|\Psi_0\rangle = H^{\otimes n}|0\rangle^n = \frac{1}{\sqrt{N}} \sum_{i=0}^{N=2^n} |i\rangle. \quad (18.2)$$

El objetivo del algoritmo de Grover es aumentar el valor del coeficiente que acompaña al estado $|\omega_0\rangle$ que queremos encontrar, es decir, aumentar la probabilidad de que al medir, obtengamos este estado. Como la suma de todas las probabilidades tiene que ser 1, este aumento en la probabilidad del estado $|\omega_0\rangle$ se produce a expensas de reducir la probabilidad del resto.

Nota

Recordemos que en mecánica cuántica podemos tener un estado que sea la **superposición** de varios estado (como es el caso anterior), pero al medir solo obtenemos **uno de estos estado**. La probabilidad de medir cada uno de estos estados que forman la superposición es igual al módulo cuadrado del coeficiente que lo acompaña en el vector de esto $|\Psi\rangle$ (el módulo cuadrado de la amplitud). Este caso, vemos que todos los estados tiene probabilidad $1/N$.

18.2.2. Amplificación de amplitud mediante iteraciones del algoritmo

Como vamos a ver a continuación, el algoritmo de Grover tiene una interpretación geométrica muy simple: dos reflexiones que rotan el vector de estado en un plano bidimensional.

El algoritmo de Grover consta de dos partes: un **oráculo** y un **difusor**. Para que el algoritmo maximice la probabilidad de la solución deseada tenemos realizar un número concreto de **iteraciones**. En cada iteración del algoritmo, primero se aplica el oráculo y después el difusor. Con cada iteración la probabilidad de medir la solución deseada va aumentando. Sin embargo, es muy importante aplicar el número correcto de iteraciones que nos maximiza la probabilidad, pues tanto si nos quedamos cortos como si nos pasamos, la probabilidad de medir $|\omega_0\rangle$ disminuye. Esto se va a entender muy bien a continuación, cuando veamos la explicación geométrica.

Pongamos el estado inicial de la Ec. (18.1) de una forma más adecuada para nuestro propósito

$$|\Psi_0\rangle = |\Psi(k(0), l(0))\rangle = k(0)|\omega_0\rangle + \sum_{i \neq \omega_0} l(0)|i\rangle, \quad \text{donde} \quad k(0) = l(0) = \frac{1}{\sqrt{N}}. \quad (18.3)$$

Denominaremos $k(t)$ y $l(t)$ a los coeficientes que tendremos en la t-esima iteración del algoritmo, es decir

$$|\Psi(t)\rangle = |\Psi(k(t), l(t))\rangle = k(t)|\omega_0\rangle + \sum_{i \neq \omega_0} l(t)|i\rangle. \quad (18.4)$$

Véase que aquí ya estamos adelantando que los coeficientes de todos los estados que no son el deseado son iguales en cada iteración. Nuestro vector de estado $|\Psi(t)\rangle$ vive en un espacio de Hilbert de N dimensiones y los estados $|i\rangle$, con $i = 0, 1, \dots, N - 1$, forman una base ortogonal del espacio de Hilbert (véase que aquí se incluye ω_0 , pues no es nada más que un valor de i concreto). Para entender bien esto, podemos fijarnos en las Ecs. (18.1) y (18.3) y ver que estas no son más que la expresión de un vector como la multiplicación de unos coeficientes por los elementos de la base. Como no podemos dibujar un vectores de más de 3 dimensiones, para explicar geométricamente el algoritmo vamos a recurrir a un truco: vamos a descomponer nuestro vector de estado en la suma de dos vectores $|\omega_0\rangle$ y

$|\omega^\perp\rangle$, donde este último es la suma de todos los demás elementos de la base, es decir:

$$|\Psi(t)\rangle = |\Psi(k(t), l(t))\rangle = k(t)|\omega_0\rangle + l(t)\sqrt{N-1}|\omega^\perp\rangle, \quad \text{donde} \quad |\omega^\perp\rangle = \frac{1}{\sqrt{N-1}} \sum_{i \neq \omega_0}^N |i\rangle. \quad (18.5)$$

Nota

Es importante darse cuenta de donde sale el factor $\sqrt{N-1}$ en la definición de $|\omega^\perp\rangle$. Este es debido a que estamos definiendo $|\omega^\perp\rangle$ como un vector unitario, es decir

$$\langle \omega^\perp | \omega^\perp \rangle = 1. \quad (18.6)$$

Por otro lado, $|\Psi_0\rangle$ también es unitario

$$\| |\Psi(t)\rangle \| ^2 = \langle \Psi(t) | \Psi(t) \rangle = k(t)^2 + \sum_{i \neq \omega_0}^N l(t)^2 = k(t)^2 + l(t)^2(N-1) = 1. \quad (18.7)$$

Con lo cual, ese factor es necesario.

Teniendo en cuenta que $|\omega_0\rangle$ es un elemento de la base, eso quiere decir que es ortogonal al resto de elementos de la base. Concluimos entonces que $|\omega_0\rangle$ y $|\omega^\perp\rangle$ son ortogonales. Podemos pues dibujar nuestro vector de estado en un plano cuyos ejes son $|\omega_0\rangle$ y $|\omega^\perp\rangle$. La imagen de la izquierda de la Fig. 18.1 podemos ver el estado inicial $|\Psi_0\rangle$ dibujado en este plano.

Nota

Veamos un ejemplo sencillo para ver a qué nos referimos con esta descomposición. Pongamos que tenemos un vector tridimensional unitario de la forma

$$\vec{r} = a\vec{x} + b\vec{y} + c\vec{z}, \quad \text{donde } a^2 + b^2 + c^2 = 1.$$

y donde \vec{x} , \vec{y} y \vec{z} son los vectores unitarios en las direcciones de los ejes X , Y y Z . Estos vectores unitarios son los **elementos de la base** que comentamos anteriormente (los equivalentes a los estados $|i\rangle$ en \mathbb{R}^3). Supongamos ahora que nuestra solución es \vec{z} . Lo que podemos hacer es juntar $a\vec{x} + b\vec{y}$ en un único vector unitario \vec{v} que represente al plano formado por \vec{x} y \vec{y}

$$\vec{v} = \frac{1}{\sqrt{a^2 + b^2}} (a\vec{x} + b\vec{y}).$$

Con lo que

$$\vec{r} = c\vec{z} + \sqrt{a^2 + b^2} \vec{v}.$$

De esta forma, al aumentar c el vector r se va poniendo cada vez más paralelo al eje Z , mientras que si aumentamos a o b el vector se acerca a \vec{v} , que representa el plano XY .

Sabemos también que todo vector en un plano podemos definirlo mediante su módulo y el ángulo respecto a uno de los ejes. Como el vector de estado tiene módulo uno (la suma de las probabilidades es uno), podemos escribirlo solo en función de un ángulo. Si llamamos θ_0 al ángulo que forman el vector de estado con el eje $|\omega^\perp\rangle$, podemos escribir

$$|\Psi_0\rangle = \sin \theta_0 |\omega_0\rangle + \cos \theta_0 |\omega^\perp\rangle, \quad \text{donde} \quad \sin \theta_0 = \cos \theta_0 = \frac{1}{\sqrt{N}}. \quad (18.8)$$

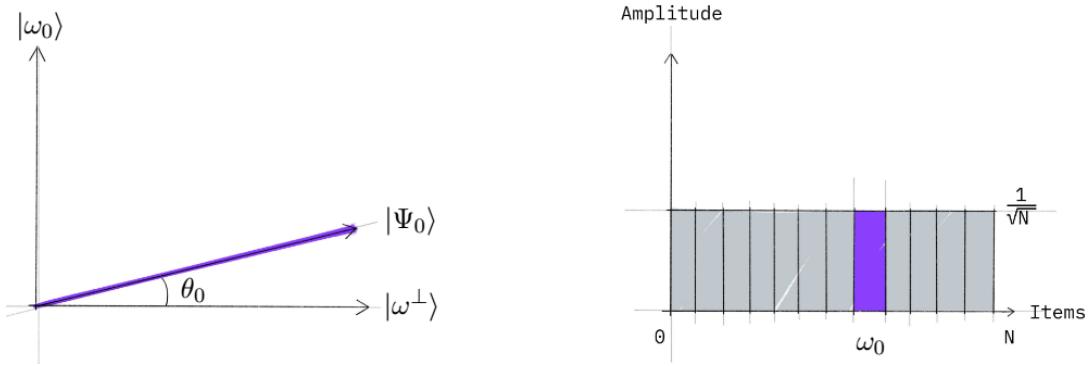


Figura 18.1: Esto inicial del algoritmo de Grover: superposición uniforme de los N estados. En la figura de la izquierda, $|\Psi_0\rangle$ representa el estado inicial, el eje $|\omega_0\rangle$ representa la solución y el eje $|\omega^\perp\rangle$ el resto de estados de espacio de Hilbert. En la figura de la derecha vemos la **amplitud** de cada estado. Recordemos que la probabilidad de cada estado es el cuadrado de la amplitud. Figura tomada de [31].

En el gráfico de barras de la derecha en la Fig. 18.1 podemos ver las amplitudes de los estados. Recordemos que la probabilidad de un estado es el cuadrado de la amplitud.

18.2.2.1. Primera parte de las iteraciones: El oráculo.

El algoritmo de Grover usa un **oráculo** cuya función es aplicar una fase negativa al estado que buscamos, es decir

$$U_{\omega_0}|i\rangle = \begin{cases} |i\rangle & \text{si } i \neq \omega_0 \\ -|i\rangle & \text{si } i = \omega_0. \end{cases} \quad (18.9)$$

Este oráculo no es más que una matriz diagonal con todo 1 en la diagonal menos en el elemento correspondiente al estado $|\omega_0\rangle$, donde tenemos un -1 . Veremos más adelante ejemplos de como construir este tipo de oráculos. Podemos adelantar que, aunque no es trivial construirlos pues a priori parece que tenemos que conocer la solución de antemano, tampoco es (en muchos caso) excesivamente complicado.

Una de las características importantes de este algoritmo es lo fácil que resulta convertir un problema a un oráculo de esta forma. Hay muchos problemas computacionales en los que es difícil encontrar una solución, pero relativamente fácil verificarla (problemas NP). Por ejemplo, podemos verificar fácilmente la solución de un sudoku comprobando que se cumplen todas las reglas. Para estos problemas, podemos crear una función f que tome una propuesta de solución i y nos devuelva $f(i) = 0$ si i no es solución ($i \neq \omega_0$) y $f(i) = 1$ si i es solución ($i = \omega_0$). Podemos entonces definir el oráculo de la forma

$$U_{\omega_0}|i\rangle = (-1)^{f(i)}|i\rangle. \quad (18.10)$$

De esta forma, a matriz es ahora una matriz diagonal con

$$\text{Diag}(U_{\omega_0}) = \left[(-1)^{f(0)}, (-1)^{f(1)}, \dots, (-1)^{f(2^n-1)} \right]. \quad (18.11)$$

Nota

Podemos usar el retorno de fase (*phase kickback*) para construir este tipo de oráculos. Si tenemos nuestra función clásica $f(x)$, podemos convertirla en un circuito reversible de la forma (ver Fig. 18.2(a))

$$|x\rangle|0\rangle \rightarrow |x\rangle|0 \oplus f(x)\rangle = |x\rangle|f(x)\rangle.$$

Si inicializamos la ancilla en el estado $|-\rangle$, tenemos (ver Fig. 18.2(b))

$$|x\rangle|-\rangle = \frac{1}{\sqrt{2}}|x\rangle(|0\rangle - |1\rangle) \rightarrow |x\rangle(|0\rangle + f(x)|1\rangle - |1\rangle + f(x)|0\rangle) = (-1)^{f(x)}|x\rangle|-\rangle.$$

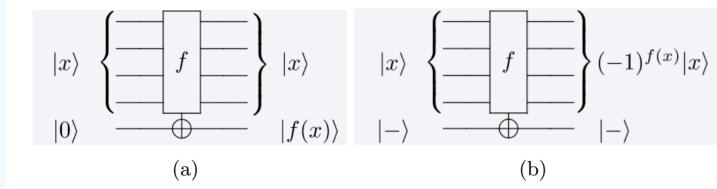


Figura 18.2: Retorno de fase (*phase kickback*) para construir un oráculo de la forma de la Ec. (18.10).

En la Fig. 18.3 podemos ver el efecto del oráculo U_{ω_0} sobre el vector y las amplitudes. El cambio de signo en la amplitud del estado $|\omega_0\rangle$ se traduce en un cambio de signo en la proyección del vector de estado sobre este eje. A efectos prácticos, esto no es más que una reflexión del vector de estado respecto al eje $|\omega^\perp\rangle$. Como las probabilidades son el cuadrado de las amplitudes, este cambio de signo no se traduce en un cambio en la probabilidades. A efectos de las medidas, nada ha cambiado. En esta figura también se representa la **media de las amplitudes** (línea punteada). Vemos que ha disminuido la media al cambiar el signo de la amplitud del estado $|\omega_0\rangle$.

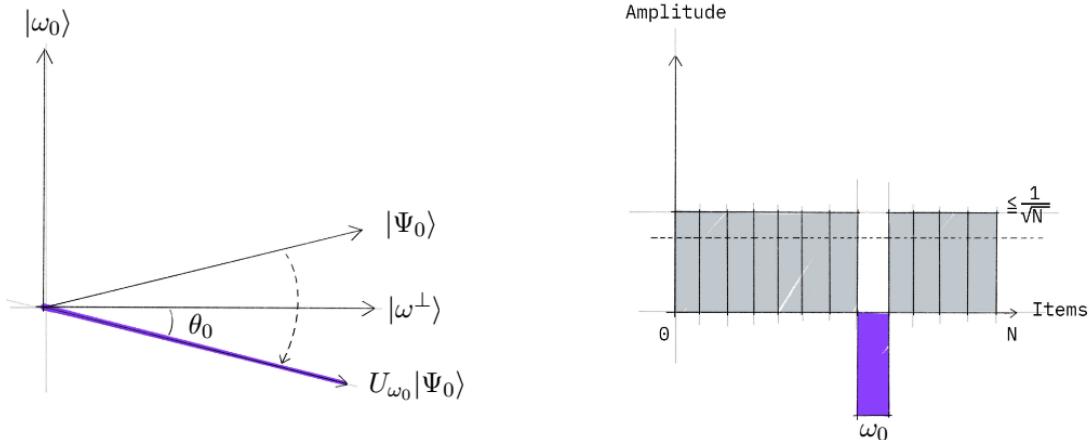


Figura 18.3: Primer paso del algoritmo de Grover: aplicación del **oráculo** U_{ω_0} para cambiar el signo de la amplitud del estado deseado. En la figura de la izquierda, $|\Psi_0\rangle$ representa el estado inicial, el eje $|\omega_0\rangle$ representa la solución y el eje $|\omega^\perp\rangle$ el resto de estados de espacio de Hilbert. En la figura de la derecha vemos la **amplitud** de cada estado, donde la línea punteada representa la media. Figura tomada de [31].

18.2.2.2. Segunda parte de las iteraciones: El difusor.

El **difusor** consiste en aplicar el operador

$$U_{\Psi_0} = 2|\Psi_0\rangle\langle\Psi_0| - I. \quad (18.12)$$

Este operador no es más que una reflexión respecto el estado inicial $|\Psi_0\rangle$.

En la Fig. 18.4 podemos ver el efecto del difusor. En el diagrama de barras de la amplitud, podemos entender esta transformación como una reflexión respecto a la media de las amplitudes (la media queda igual). Como habíamos disminuido la media al aplicar el oráculo, lo que tenemos ahora es una

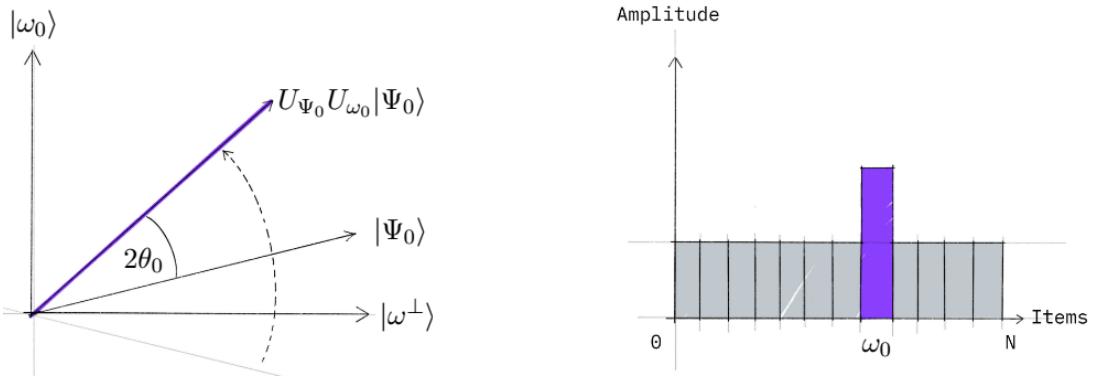


Figura 18.4: Segundo paso del algoritmo de Grover: aplicar el operador de **difusión** (o reflexión) $U_{\Psi_0} = 2|\Psi_0\rangle\langle\Psi_0| - I$. En la figura de la izquierda, $|\Psi_0\rangle$ representa el estado inicial, el eje $|\omega_0\rangle$ representa la solución y el eje $|\omega^\perp\rangle$ el resto de estados de espacio de Hilbert. En la figura de la derecha vemos la **amplitud** de cada estado. Recordemos que la probabilidad de cada estado es el cuadrado de la amplitud. Figura tomada de [31].

amplificación de la amplitud del estado deseado. Esto se ve también en el plano de la izquierda, pues las amplitudes no son más que las proyecciones del vector sobre los ejes.

Lo que estamos haciendo mediante la aplicación del oráculo y el difusor no es mas que **rotar el vector de estado un ángulo $2\theta_0$** (donde θ_0 está definido en la ec. (18.8)) hacia el eje que representa nuestra solución, aumentando así su proyección, es decir, su amplitud, y con ello la probabilidad de medirlo.

Después de t iteraciones hemos aumentado el ángulo en $2t\theta_0$, con lo que tenemos estado:

$$|\Psi(t)\rangle = (U_{\Psi_0} U_{\omega_0})^t |\Psi_0\rangle = \sin((2t+1)\theta_0) |\omega_0\rangle + \cos((2t+1)\theta_0) |\omega^\perp\rangle. \quad (18.13)$$

Podemos ahora definir el **operador de Grover**

$$G = U_{\Psi_0} U_{\omega_0} \Rightarrow |\Psi(t)\rangle = G^t |\Psi_0\rangle. \quad (18.14)$$

Ahora es fácil entender porqué tenemos que aplicar un número concreto de iteraciones y porqué si nos pasamos, la probabilidad disminuye. Como acabamos de ver, el resultado después de cada iteración es que rotamos el vector de estado $2\theta_0$ en el sentido contrario de las agujas del reloj. Lo que queremos es que el vector quede lo más vertical posible, es decir, que quede lo más cerca posible del eje $|\omega\rangle$. Si hacemos demasiadas iteraciones, lo que vamos a conseguir es “pasarnos de largo” del eje. Discutiremos el número exacto de iteraciones en la sección 18.3, pero ya comentamos que es del orden de \sqrt{N} .

Si comparamos las ecuaciones (18.5) y (18.13) vemos que

$$k(t) = \sin[(2t+1)\theta_0] \quad l(t) = \frac{1}{\sqrt{N-1}} \cos[(2t+1)\theta_0]. \quad (18.15)$$

18.3. Número conocido de soluciones.

Vamos a empezar el análisis formal tratando el caso en el que **conocemos el número M de soluciones** que hay en nuestro dataset. Es decir, tenemos M valores diferentes i que cumplen $L_i = x$. Denominemos ω al conjunto de los M valores i que son solución, y denominemos ω^\perp al conjunto de los $N - M$ valores i que no son solución

$$\omega = \{i | L_i = x\} \quad \omega^\perp = \{i | L_i \neq x\}. \quad (18.16)$$

Supondremos también que **estamos en el caso en el que** $N = 2^n$ y que además partimos del estado de la Ec. (18.1), es decir, de una **superposición uniforme**.

Nota Importante!!

Uno de los pasos del algoritmo de Grover es una reflexión de las amplitudes respecto a la media.

Esto implica que podemos tener los siguientes casos:

- $M < N/2$: El algoritmo funciona normal.
- $M = N/2$: El algoritmo no funciona
- $M > N/2$: El algoritmo amplifica las soluciones incorrectas.

18.3.1. Generalización de las expresiones de la sección 18.2 para M soluciones.

Vemos a reescribir las ecuaciones enmarcadas de la sección 18.2 para el caso de M soluciones. Empecemos reescribiendo las expresiones del estado inicial de la Ec. (18.3) y del estado $|\Psi_j\rangle$ de la Ec. (18.13)

$$|\Psi_0\rangle = |\Psi(k(0), l(0))\rangle = \sum_{i \in \omega} k(0)|i\rangle + \sum_{i \in \omega^\perp} l(0)|i\rangle, \quad \text{donde} \quad k(0) = l(0) = \frac{1}{\sqrt{N}}. \quad (18.17)$$

$$|\Psi(t)\rangle = |\Psi(k(t), l(t))\rangle = \sum_{i \in \omega} k(t)|i\rangle + \sum_{i \in \omega^\perp} l(t)|i\rangle. \quad (18.18)$$

Vemos que ahora el primer sumatorio tiene M elementos, mientras que el segundo tiene $N - M$. Por supuesto, se cumple que $|\Psi(t)\rangle$ tiene módulo 1, es decir

$$\langle \Psi(t) | \Psi(t) \rangle = [Mk(t)^2 + (N - M)l(t)^2] = 1 \quad \forall t. \quad (18.19)$$

Podemos ahora también redefinir el estado $|\omega^\perp\rangle$ de la Ec. (18.5) y definir $|\omega\rangle$

$$|\omega^\perp\rangle = \frac{1}{\sqrt{N - M}} \sum_{i \in \omega^\perp} |i\rangle \quad |\omega\rangle = \frac{1}{\sqrt{M}} \sum_{i \in \omega} |i\rangle \quad (18.20)$$

de forma que

$$|\Psi(t)\rangle = k(t)\sqrt{M}|\omega\rangle + l(t)\sqrt{N - M}|\omega^\perp\rangle. \quad (18.21)$$

Nota

Nuevamente, los factores $\sqrt{N - M}$ y \sqrt{M} en las definiciones de $|\omega^\perp\rangle$ y $|\omega\rangle$ son para hacerlos vectores unitarios

$$\langle \omega^\perp | \omega^\perp \rangle = 1 \quad \langle \omega | \omega \rangle = 1. \quad (18.22)$$

Veamos ahora la expresión del esto inicial en función del ángulo θ de la Ec. (18.8)

$$|\Psi_0\rangle = \sin \theta |\omega\rangle + \cos \theta |\omega^\perp\rangle, \quad \text{donde} \quad \sin \theta = \sqrt{\frac{M}{N}}. \quad (18.23)$$

Nuevamente, comparando las Ecs. (18.23) y (18.21) obtenemos las expresiones de $k(t)$ y $l(t)$ en función del ángulo θ

$$k(t) = \frac{1}{\sqrt{M}} \sin [(2t + 1)\theta] \quad l(t) = \frac{1}{\sqrt{N - M}} \cos [(2t + 1)\theta]. \quad (18.24)$$

Vemos que estas expresiones cumplen la Ec. (18.19) Finalmente, por ser rigurosos (y por recopilar todas las ecuaciones juntas), tenemos que el oráculo (18.9), el difusor (18.12) nos quedan:

$$\boxed{U_\omega|i\rangle = \begin{cases} |i\rangle & \text{si } i \notin \omega \\ -|i\rangle & \text{si } i \in \omega \end{cases}} \quad \boxed{U_{\Psi_0} = 2|\Psi_0\rangle\langle\Psi_0| - I}. \quad (18.25)$$

donde recordemos que denominamos operador de Grover a

$$\boxed{G = U_{\Psi_0}U_\omega} \quad \Rightarrow \quad |\Psi(t)\rangle = G^t|\Psi_0\rangle. \quad (18.26)$$

Véase que el caso particular de una solución es, efectivamente, aquel con $M = 1$.

18.3.2. Número de iteraciones.

Denominemos T al **número de iteraciones para el cual la probabilidad de medir la solución correcta se maximiza**. Es decir, $l(T) \approx 0$ y $k(T) \approx 1$. Sabemos que $l(\tilde{T}) = 0$ cuando el coseno es igual cero, es decir:

$$l(\tilde{T}) = 0 \quad \Rightarrow \quad (2\tilde{T} + 1)\theta = \frac{\pi}{2} \quad \Rightarrow \quad \tilde{T} = \frac{\pi}{4\theta} - \frac{1}{2}. \quad (18.27)$$

Por regla general, este valor \tilde{T} no será un entero. Tomemos

$$T = \lfloor \pi/4\theta \rfloor \quad (18.28)$$

donde la notación $\lfloor \alpha \rfloor$ quiere decir que **redondeamos a un entero por truncamiento**. Véase que $|T - \tilde{T}| \leq 1/2$. Se sigue que $|(2T + 1)\theta - (2\tilde{T} + 1)\theta| \leq \theta$. Pero $(2\tilde{T} + 1)\theta = \pi/2$ por definición de \tilde{T} . Con lo cual $|\cos((2T + 1)\theta)| \leq |\sin \theta|$. Concluimos entonces que la probabilidad de fallo después de $T = \lfloor \pi/4\theta \rfloor$ es

$$(N - M)l^2(T) = \cos^2((2T + 1)\theta) \leq \sin^2 \theta = \frac{M}{N} \quad (18.29)$$

que es despreciable si $M \ll N$. Véase que el $(N - M)$ de la expresión anterior es porque hay $(N - M)$ estados incorrectos que podemos medir, cada uno de ellos con probabilidad $l^2(T)$.

Véase que el algoritmo corre en un tiempo del orden de $\mathcal{O}(\sqrt{N/M})$ ya que

$$T \leq \frac{\pi}{4\theta} \approx \frac{\pi}{4\sin \theta} \approx \frac{\pi}{4}\sqrt{\frac{N}{M}} \quad \Rightarrow \quad \boxed{T = \left\lfloor \frac{\pi}{4}\sqrt{\frac{N}{M}} \right\rfloor}. \quad (18.30)$$

18.3.3. Extra: Formulación recursiva de $k(t)$ y $l(t)$.

Ya hemos comentado en cada iteración se aplican los dos operadores de la Ec. (18.25), es decir

$$|\Psi(t+1)\rangle = U_{\Psi_0}U_\omega|\Psi(t)\rangle. \quad (18.31)$$

Desarrollamos esta expresión para ver la relación de recursividad de los coeficientes, es decir, para ver la expresión de $k(t+1)$ y $l(t+1)$ en función de $k(t)$ y $l(t)$. Primero vamos el término $U_\omega|\Psi(t)\rangle$ usando la expresión de U_ω de la Ec. (18.25):

$$U_\omega|\Psi(t)\rangle = -k(t)\sqrt{M}|\omega\rangle + l(t)\sqrt{N-M}|\omega^\perp\rangle. \quad (18.32)$$

Desarrollamos un poco la Ec. (18.31) usando la expresión de U_{Ψ_0} de la Ec. (18.25)

$$\begin{aligned} |\Psi(t+1)\rangle &= U_{\Psi_0}U_\omega|\Psi(t)\rangle \\ &= \left(2|\Psi_0\rangle\langle\Psi_0| - I\right)U_\omega|\Psi(t)\rangle \\ &= 2|\Psi_0\rangle\langle\Psi_0|U_\omega|\Psi(t)\rangle - U_\omega|\Psi(t)\rangle. \end{aligned} \quad (18.33)$$

Usando la Ec. (18.32) el término $\langle \Psi_0 | U_\omega | \Psi(t) \rangle$ nos queda

$$\langle \Psi_0 | U_\omega | \Psi(t) \rangle = \frac{1}{\sqrt{N}} \left(-k(t)M + (N - M)l(t) \right). \quad (18.34)$$

Calculo de $\langle \Psi_0 | U_\omega | \Psi(t) \rangle$

Hacemos ahora el producto $\langle \Psi_0 | U_\omega | \Psi(t) \rangle = \langle \Psi_0 | \left(U_\omega | \Psi(t) \rangle \right)$ usando la Ec. (18.32)

$$\begin{aligned} \langle \Psi_0 | U_\omega | \Psi(t) \rangle &= \langle \Psi_0 | \left(-k(t)\sqrt{M}|\omega\rangle + l(t)\sqrt{N-M}|\omega^\perp\rangle \right) \\ &= \left(k(0)\sqrt{M}\langle\omega| + l(0)\sqrt{N-M}\langle\omega^\perp| \right) \left(-k(t)\sqrt{M}|\omega\rangle + l(t)\sqrt{N-M}|\omega^\perp\rangle \right). \end{aligned}$$

Debido a la ortogonalidad de los estado (esto es, $\langle i|j\rangle = 0$ si $j \neq i$), y teniendo en cuenta que $k(0) = l(0) = 1/\sqrt{N}$ tenemos

$$\langle \Psi_0 | U_\omega | \Psi(t) \rangle = -k(0)k(t)M + (N - M)l(0)l(t) = \frac{1}{\sqrt{N}} \left(-k(t)M + (N - M)l(t) \right).$$

Sustituyendo (18.32) y (18.34) en (18.33) tenemos:

$$|\Psi(t+1)\rangle = \left(\frac{N-2M}{N}k(t) + \frac{2(N-M)}{N}l(t) \right) \sum_{i \in \omega} |i\rangle + \left(\frac{N-2M}{N}l(t) - \frac{2M}{N}k(t) \right) \sum_{i \in \omega^\perp} |i\rangle. \quad (18.35)$$

Calculo de la Ec. (18.35)

Sustituyendo (18.32) y (18.34) en (18.33) tenemos:

$$\begin{aligned} |\Psi(t+1)\rangle &= 2 \frac{1}{\sqrt{N}} \left(-k(t)M + (N - M)l(t) \right) |\Psi_0\rangle - k(t)\sqrt{M}|\omega\rangle + l(t)\sqrt{N-M}|\omega^\perp\rangle = \\ &= \frac{2}{\sqrt{N}} \left(-k(t)M + (N - M)l(t) \right) \frac{1}{\sqrt{N}} \left(\sqrt{M}|\omega\rangle + \sqrt{N-M}|\omega^\perp\rangle \right) \\ &\quad + k(t)\sqrt{M}|\omega\rangle + l(t)\sqrt{N-M}|\omega^\perp\rangle = \\ &= \left[\frac{2}{N} \left(-k(t)M + (N - M)l(t) \right) + k(t) \right] \sqrt{M}|\omega\rangle \\ &\quad + \left[\frac{2}{N} \left(-k(t)M + (N - M)l(t) \right) - l(t) \right] \sqrt{N-M}|\omega^\perp\rangle. \end{aligned}$$

Simplificando y teniendo en cuenta que las expresión de $|\omega\rangle$ y $|\omega^\perp\rangle$ en la Ec. (18.20), llegamos a la Ec. (18.35).

Finalmente llegamos a las relaciones:

$$\boxed{k(t+1) = \frac{N-2M}{N}k(t) + \frac{2(N-M)}{N}l(t)}, \quad \boxed{l(t+1) = \frac{N-2M}{N}l(t) - \frac{2M}{N}k(t)}. \quad (18.36)$$

18.4. Número desconocido de soluciones

Como ya hemos comentado, para poder aplicar el algoritmo de Grover tal y como lo hemos visto hasta ahora, nos hace falta conocer el número de soluciones. Esto es debido a que debemos aplicar un número concreto de iteraciones para maximizar la probabilidad de las soluciones correctas. Si no

aplicamos el número correcto de iteraciones, la probabilidad puede ser incluso nula. Como vemos en la Ec. (18.30), para saber el número de iteraciones tenemos que conocer el número de soluciones.

En esta sección vamos a ver un algoritmo para poder abordar el caso en el que no conocemos el número de soluciones y por ende, no sabemos cuantas iteraciones debemos aplicar. Este algoritmo no es más que una forma inteligente de aplicar el algoritmo de Grover. Lo bueno de este algoritmo es que nuevamente podemos encontrar una solución con un número de iteraciones $\mathcal{O}(\sqrt{N/M})$, aunque a priori no sabemos cuantas iteraciones son. Vamos a ver primero dos Lemmas que nos servirán para entender el algoritmo que se plantea en el Teorema 31.

Nota Importante!!

Uno de los pasos del algoritmo de Grover es una reflexión de las amplitudes respecto a la media. Esto implica que podemos tener los siguientes casos:

- $M < N/2$: El algoritmo funciona normal.
- $M = N/2$: El algoritmo no funciona
- $M > N/2$: El algoritmo amplifica las soluciones incorrectas.

18.4.1. Conocimientos previos.

Lemma 7 *Para cualquier par de números α y β , y cualquier posible entero r tenemos*

$$\sum_{j=0}^{r-1} \cos(\alpha + 2\beta j) = \frac{\sin(r\beta) \cos(\alpha + (r-1)\beta)}{\sin \beta}. \quad (18.37)$$

En particular, si $\alpha = \beta$,

$$\sum_{j=0}^{r-1} \cos((2j+1)\alpha) = \frac{\sin(2r\alpha)}{2 \sin \alpha}. \quad (18.38)$$

Lemma 8 *Sea M el número (desconocido) de soluciones de un problema tipo Grover con N elementos y sea θ tal que $\sin^2 \theta = M/N$. Sea r un entero positivo aleatorio. Sea t un entero aleatorio (con distribución uniforme) entre 0 y $r-1$. Si observamos el registro después de t iteraciones del algoritmo de Grover empezando desde el estado $|\Psi_0\rangle = \sum_i \frac{1}{\sqrt{N}}|i\rangle$, la probabilidad de obtener la solución correcta es exactamente*

$$P_r = \frac{1}{2} - \frac{\sin(4r\theta)}{4r \sin(2\theta)}. \quad (18.39)$$

En particular, tenemos $P_r \geq 1/4$ si $r \geq 1/\sin(2\theta)$.

Demostración: Como ya sabemos, si tenemos M soluciones la probabilidad de que tras t iteraciones se mida el resultado correcto es M veces el valor del coeficiente $k^2(t)$ (la amplitud al cuadrado del estado $i \in \omega$ en la Ec. (18.31)). Teniendo en cuenta la Ec. (18.36) sabemos que la probabilidad es

$$P(t) = Mk^2(t) = \sin^2((2t+1)\theta).$$

De esto se sigue que la probabilidad promedio si tomamos un número de iteraciones t tal que

$0 \leq t < r$ es

$$P_r = \frac{1}{r} \sum_{t=0}^{r-1} P(t) = \frac{1}{r} \sum_{t=0}^{r-1} \sin^2((2t+1)\theta) = \frac{1}{2r} \sum_{t=0}^{r-1} \left(1 - \cos(2(2t+1)\theta)\right) = \frac{1}{2} - \frac{\sin(4r\theta)}{4r \sin(2\theta)},$$

donde en la última igualdad se ha usado el Lemma 7 y al identidad trigonométrica

$$\sin^2 \theta = \frac{1 - \cos(2\theta)}{2}. \quad (18.40)$$

Si estamos en el caso de $r \geq 1/\sin(2\theta)$ tenemos

$$\frac{\sin(4r\theta)}{4r \sin(2\theta)} \leq \frac{1}{4r \sin(2\theta)} \leq \frac{1}{4} \Rightarrow P_r \geq \frac{1}{2} - \frac{1}{4},$$

donde la primera desigualdad se cumple siempre (independientemente del valor de r) pues viene de que la función seno está acotada entre -1 y 1 . ■

Nota

A veces al ver las formulaciones tan formales que se dan en los teoremas o lemmas, perdemos un poco el norte sobre lo que nos quieren transmitir. Veámoslos en palabras más llanas:

- El Lemma 7 simplemente es una relación trigonométrica un poco exótica y nos sirve para demostrar el Lemma 8
- El Lemma 8 nos da la expresión de la probabilidad promedio que tenemos de medir el resultado correcto usando el algoritmo de Grover en un caso particular. En este caso lo que hacemos es, dado un número r , ejecutar t iteraciones de Grover donde t es un número aleatorio entre 0 y $r-1$. Probando muchas veces con muchos número aleatorio t , en promedio la probabilidad P_r de obtener el resultado correcto está dada por (18.39). Véase que, para un problema de Grover concreto (θ fijo), esta probabilidad depende solo de r .

18.4.2. Algoritmo para el caso de M desconocido.

Teorema 31 *El siguiente algoritmo encuentra una solución en un tiempo esperado $\mathcal{O}(\sqrt{N/M})$ (si tomamos $1 \leq M \leq 3N/4$)*

1. Inicializamos $r = 1$ y $\lambda = 6/5$. (En realidad, cualquier valor $1 \leq \lambda \leq 4/3$ sirve).
2. Elegimos un valor aleatorio t con distribución uniforme tal que $0 \leq t < r$.
3. Aplicamos t iteraciones del algoritmo de Grover empezando desde el estado inicial $|\Psi_0\rangle = \sum_i \frac{1}{\sqrt{N}} |i\rangle$.
4. Medimos, de forma que obtenemos uno de los estados $|i\rangle$ del paso anterior.
5. Si $L_i = x$, hemos encontrado una solución: **Exit**.
6. En caso contrario, tomemos $r = \min(\lambda r, \sqrt{N})$ y volvemos al paso 2.

Demostración: Sea θ un ángulo tal que

$$\sin \theta = \sqrt{N/M}, \quad (18.41)$$

(como en la Ec. (18.23)). Denominemos r_s al valor de r en la s -esima iteración del bucle principal, es decir

$$r_s = \lambda^{s-1}, \quad \text{con } s = 1, 2, \dots \quad \text{y } \lambda = \frac{4}{3}. \quad (18.42)$$

(No confundir las **iteraciones del bucle principal** con las **iteraciones de Grover**: en cada iteración s del bucle principal hacemos t iteraciones de Grover.) Sea r_0 tal que

$$r_0 = \frac{1}{\sin(2\theta)}. \quad (18.43)$$

(Véase que r_0 no es valor inicial de r_s). Aplicando un poco de trigonometría

$$\sin^2 \theta = \frac{1 - \cos(2\theta)}{2} \Rightarrow 1 - 2\sin^2 \theta = \cos(2\theta) = \sqrt{1 - \sin^2(2\theta)} \Rightarrow \sin^2(2\theta) = 1 - (1 - 2\sin^2 \theta)^2.$$

Desarrollando el cuadrado y teniendo en cuenta la Ec. (18.41) llegamos a

$$r_0 = \frac{1}{\sin(2\theta)} = \frac{N}{2\sqrt{(N-M)M}} < \sqrt{\frac{N}{M}}. \quad (18.44)$$

donde en la desigualdad se ha tenido en cuenta que $M \leq 3N/4$. Veámoslo

$$\frac{N}{2\sqrt{(N-M)M}} < \sqrt{\frac{N}{M}} \Rightarrow \frac{\sqrt{N}}{2\sqrt{(N-M)}} < 1 \Rightarrow \frac{N}{(N-M)} < 4 \Rightarrow 1 - \frac{1}{4} > \frac{M}{N}.$$

Debemos estimar el número esperado total de iteraciones de Grover que tenemos que hacer (sumando todas las interacciones de Grover en todas las iteraciones del bucle principal). El tiempo total será del orden de este número. Sabemos que **el número promedio de iteraciones de Grover que se realizan en la iteración s -esima del bucle principal es menor que $r_s/2$** . Escribiéndolo de forma bonita:

$$\bar{t}_s < \frac{r_s}{2} = \frac{\lambda^{s-1}}{2} \quad \text{donde la barra significa promedio.} \quad (18.45)$$

Esto es debido a que en cada iteración del bucle principal hacemos un número aleatorio de iteraciones t_s de Grover con $0 \leq t_s < r_s$. Denominamos **fase crítica** al momento en el que hemos realizado suficientes iteraciones del bucle principal como para tener $r_s \geq r_0$, es decir

$$r_s > r_0 \Rightarrow \lambda^{s-1} > r_0 \Rightarrow s > 1 + \log_\lambda r_0 \Rightarrow s > \lceil \log_\lambda r_0 \rceil. \quad (18.46)$$

con lo cual, **estamos en la fase crítica cuando el número de iteraciones del bucle principal es mayor que $\lceil \log_\lambda r_0 \rceil$** (donde esta notación significa *redondear hacia el siguiente entero*). Si nos fijamos en el Lemma 8, esta fase corresponde a aquella donde $P_r > 1/4$.

Teniendo en cuenta la Ec. (18.45), vemos que el número promedio total de iteraciones de Grover que se realizan antes de llegar a la fase crítica es

$$\begin{aligned} \sum_{s=1}^{\lceil \log_\lambda r_0 \rceil} \bar{t}_s &< \frac{1}{2} \sum_{s=1}^{\lceil \log_\lambda r_0 \rceil} \lambda^{s-1} = \frac{1}{2} \sum_{\tilde{s}=0}^{\lceil \log_\lambda r_0 \rceil - 1} \lambda^{\tilde{s}} = \frac{1}{2} \frac{\lambda^{\lceil \log_\lambda r_0 \rceil} - 1}{\lambda - 1} < \\ &< \frac{1}{2} \frac{\lambda^{1+\log_\lambda r_0} - 1}{\lambda - 1} = \frac{1}{2} \frac{r_0 \lambda - 1}{\lambda - 1} < \frac{1}{2} \frac{\lambda}{\lambda - 1} r_0 = \boxed{3r_0}. \end{aligned} \quad (18.47)$$

donde se ha aplicado la formula e la suma de la serie geométrica.

Nota: Suma de serie geométrica

$$\sum_{k=0}^n r^k = \frac{1 - r^{n+1}}{1 - r} = \frac{r^{n+1} - 1}{r - 1}. \quad (18.48)$$

Vemos que el algoritmo llega a la fase crítica en un tiempo $\mathcal{O}(r_0) < \mathcal{O}(\sqrt{N/M})$ (ver Ec. (18.44)). Como ya comentamos, una vez alcanzado el estado crítico, por el Lemma 8, sabemos que en cada nueva iteración del bucle principal la **probabilidad de éxito $P_r \geq 1/4$** ya que $r_s > 1/\sin(2\theta)$. De ello se deduce que el número esperado de iteraciones de Grover necesarias para tener éxito una vez alcanzada la fase crítica está acotado superiormente por

$$\sum_{\tilde{u}=1}^{\infty} (1 - P_r)^{\tilde{u}-1} P_r \bar{t}_{\tilde{u}+1+\lceil \log_{\lambda} r_0 \rceil} < \frac{1}{2} \sum_{\tilde{u}=1}^{\infty} \frac{3^{\tilde{u}-1}}{4^{\tilde{u}-1}} \frac{1}{4} \lambda^{\tilde{u}+1+\lceil \log_{\lambda} r_0 \rceil}, \quad (18.49)$$

donde $(1 - P_r)^{\tilde{u}-1}$ es la probabilidad de fallar $\tilde{u} - 1$ veces, con lo que $(1 - P_r)^{\tilde{u}-1} P_r$ es la probabilidad de haber fallado $\tilde{u} - 1$ veces y acertar en la \tilde{u} -esima. \bar{t}_s viene dada por (18.45). Véase que para acotar las iteraciones nos hemos puesto en el peor caso, en el que la probabilidad de éxito es la mínima y no aumenta ($P_r = 1/4$). Podemos hacer al cambio de variable $u = \tilde{u} - 1$ para poner la expresión anterior más acorde para realizar la suma de la serie geométrica (ver Ec. (18.48))

$$\frac{1}{2} \sum_{u=0}^{\infty} \frac{3^u}{4^u} \frac{1}{4} \lambda^{u+\lceil \log_{\lambda} r_0 \rceil} < \frac{1}{2} \frac{1}{4} \lambda^{1+\log_{\lambda} r_0} \sum_{u=0}^{\infty} \frac{3^u}{4^u} \lambda^u = \frac{r_0 \lambda}{8} \frac{1}{1 - 3\lambda/4} = \frac{\lambda}{8 - 6\lambda} r_0 = \boxed{\frac{3}{2} r_0}. \quad (18.50)$$

El número total esperado de iteraciones de Grover, en caso de que se alcance la fase crítica, está, por tanto, limitado por la suma de (18.47) más (18.50), es decir,

$$T = r_0 + \frac{3}{2} r_0 = \frac{9}{2} r_0 < \frac{9}{2} \sqrt{\frac{M}{N}}, \quad (18.51)$$

con lo que el tiempo esperado total es $\mathcal{O}(\sqrt{N/M})$ siempre que $0 < M \leq 3N/4$. Véase que $\frac{9}{2} r_0 \approx \frac{9}{2} \sqrt{N/M}$ cuando $M \ll N$, que es menos de 4 veces el tiempo esperado en el caso de conocer M de antemano (ver Ec. (18.30)). El caso $M > 3N/4$ puede resolverse en tiempo esperado constante mediante muestreo clásico. El caso $M = 0$ se maneja mediante un tiempo de espera apropiado en el algoritmo anterior, que permite afirmar en un tiempo en $\mathcal{O}(\sqrt{N})$ que no hay soluciones cuando este es el caso, con una probabilidad de fallo arbitrariamente pequeña cuando de hecho hay una solución. ■

18.5. Conteo de soluciones (Quantum counting)

En esta sección vamos a ver un algoritmo para obtener el número M de soluciones. Este algoritmo puede decirse que en cierta medida está inspirado en el algoritmo de Shor, pues lo que vamos a hacer es usar el **algoritmo de estimación de fase cuántico (QPE)** para obtener el ángulo θ a partir del operador de Grover G . Despues, podemos usar la Ec. (18.23) para obtener M partir de θ .

Vamos a seguir en el caso en el que $N = 2^n$ y la distribución de probabilidad inicial es uniforme.

18.5.1. Breve resumen de la estimación de fase cuántica (QPE).

Dado un operador unitario U y un autovector $|\psi\rangle$ del mismo, tenemos:

$$U |\psi\rangle = e^{2\pi i \alpha} |\psi\rangle$$

El **algoritmo de estimación de fase cuántica** lo que hace es calcular un valor aproximado del ángulo α . En la Fig. 18.5 podemos ver el circuito que implementa este algoritmo. No vamos a entrar a hablar en detalle del mismo (puede verse una explicación más detallada en el capítulo 15). Simplemente comentar dos cosas:

- Por el registro de qubits de abajo en la Fig. 18.5 debe de entrar el autoestado de U del cual queremos medir la fase.
- Si por el registro de conteo entran p qubits (el de arriba en la Fig. 18.5), en la salida vamos a medir el estado $|2^p\alpha\rangle$.

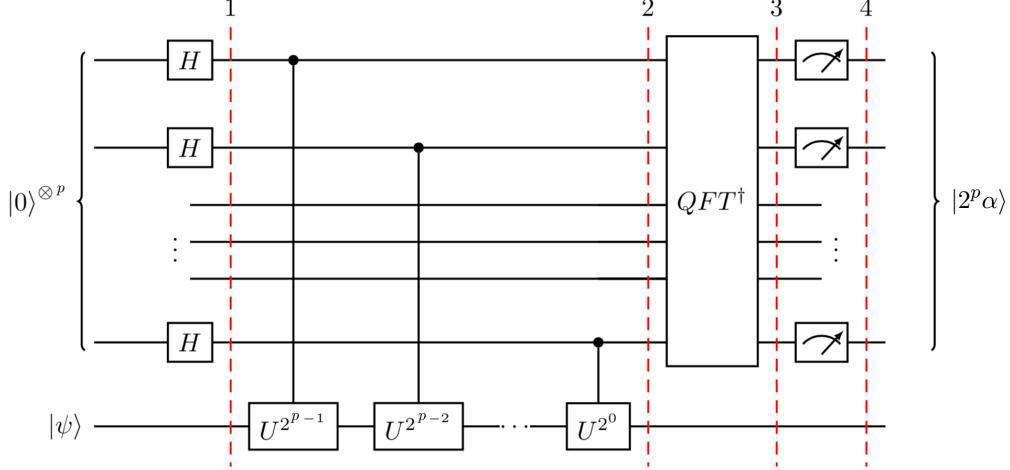


Figura 18.5: Implementación del algoritmo de estimación de fase cuántica (en el convenio estándar, siendo el bit más significativo el de arriba).

18.5.2. Estimación de fase con el operador de Grover.

Como ya comentamos anteriormente, el operador de Grover G (ver Ec. (18.26)) rota el vector de estado en un ángulo 2θ , donde θ está dado por (18.23), es decir

$$|\Psi(t+1)\rangle = G|\Psi(t)\rangle = e^{i2\theta}|\Psi(t)\rangle. \quad (18.52)$$

En concreto, se puede aplicar sobre el estado inicial

$$G|\Psi_0\rangle = e^{i2\theta}|\Psi_0\rangle, \quad \text{donde} \quad |\Psi_0\rangle = H^{\otimes n}|0\rangle = \frac{1}{2^n} \sum_{i=0}^{2^n} |i\rangle \quad (18.53)$$

Podemos pues usar el algoritmo de QPE poniendo el registro de abajo en el estado $|\Psi_0\rangle$ y mediremos a la salida el estado $|2^p 2\theta/2\pi\rangle = |2^p \theta/\pi\rangle$. Podemos ver esto en la Fig. 18.6.

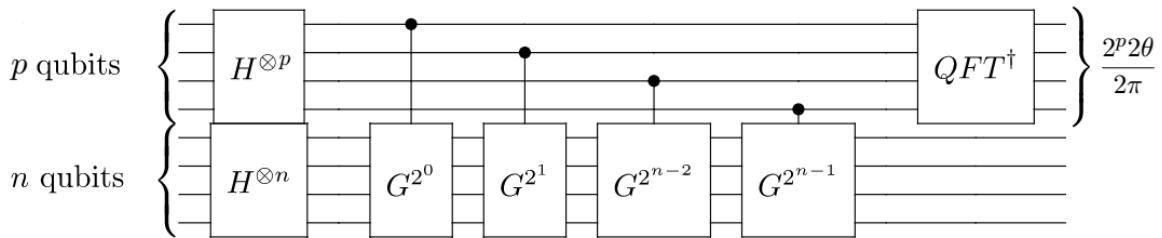


Figura 18.6: Circuito para la QPE en el caso de Grover.

Como acabamos de comentar, al aplicar QPE mediremos un valor $\tilde{f} = 2^p \tilde{\theta}/\pi$. Podemos despejar $\tilde{\theta}$ y usar (18.23) para calcular \tilde{M}

$$\tilde{\theta} = \frac{\tilde{f}\pi}{2^p} \quad \Rightarrow \quad \tilde{M} = 2^n \sin^2 \tilde{\theta} = 2^n \sin^2 \frac{\tilde{f}\pi}{2^p}, \quad (18.54)$$

donde las $\tilde{\theta}$, \tilde{f} y \tilde{M} llevan una tilde para diferenciarlos de los valores reales. Esto es, porque dependiendo del como de grande sea p (cuantos qubits tengamos en el registro de conteo) más nos acercaremos a medir el valor real del ángulo θ .

Nota

Como veremos en la sección 18.6.1, muchas veces en vez de implementar el operador de difusión U_{Ψ_0} (ver Ec. (18.25)) en realidad se implementa $-U_{\Psi_0}$. En una búsqueda de Grover normal, esta fase es global y no afecta al resultado, pero ahora estamos aplicando versiones controladas del operador de Grover G , con lo que esta fase afecta. En esencia, el único cambio es que en realidad estamos contando aquellos estados que *no son solución*. Lo único que tenemos que hacer para obtener el número de soluciones es restar al total de estados, N , el valor que obtenemos de aplicar QPE.

18.5.2.1. Extra: Precisión de \tilde{M} respecto a M .

Podemos evaluar la precisión del valor de \tilde{M} respecto al valor real M , acotando la desviación entre los mismos a medida que variamos p . Para ello, vamos a empezar asumiendo que la diferencia entre el valor real y el medido es menor que uno, es decir, $|f - \tilde{f}| < 1$. Esto sucede con una probabilidad razonable si f es suficientemente grande (si p es suficientemente grande). Teniendo en cuenta la Ec. (18.54), vemos que

$$|f - \tilde{f}| < 1 \Rightarrow |\theta - \tilde{\theta}| < \frac{\pi}{2^p} \Rightarrow |\sin \theta - \sin \tilde{\theta}| < \frac{\pi}{2^p},$$

donde en la última expresión se ha tenido en cuenta que $\sin \theta \approx \theta$ si θ es pequeño, lo cual es lógico si consideramos $M \ll N$. Jugando con las desigualdades puede verse que

$$|M - \tilde{M}| < \frac{2\pi}{2^p} \sqrt{MN} + \frac{\pi^2}{(2^p)^2} N. \quad (18.55)$$

Cálculo: Derivación de la Ec. (18.55)

Teniendo en cuenta las desigualdades anteriores, podemos derivar la expresión deseada

$$|M - \tilde{M}| < N \left| \sin^2 \theta - \sin^2 \tilde{\theta} \right| = N \left| \sin \theta - \sin \tilde{\theta} \right| \left| \sin \theta + \sin \tilde{\theta} \right| < N \frac{\pi}{2^p} \left(\sin \theta + \sin \tilde{\theta} \right).$$

En la última igualdad hemos quitado el valor absoluto porque es una suma de dos términos positivos ($0 < \theta < \pi/2$). Precisamente, como estos dos senos son positivos podemos escribir

$$\left| \sin \theta - \sin \tilde{\theta} \right| < \frac{\pi}{2^p} \Rightarrow \begin{cases} \sin \tilde{\theta} < \sin \theta + \frac{\pi}{2^p} \\ \sin \theta < \sin \tilde{\theta} + \frac{\pi}{2^p}. \end{cases}$$

Esto es porque tenemos dos números positivos que se diferencian en menos de una cierta cantidad α , así que es siempre cierto que la suma de uno de los números más α es mayor que el otro. Podemos usar la primera de estas desigualdades para seguir con el cálculo

$$|M - \tilde{M}| < N \frac{\pi}{2^p} \left(\sin \theta + \sin \tilde{\theta} \right) < N \frac{\pi}{2^p} \left(2 \sin \theta + \frac{\pi}{2^p} \right) \quad (18.56)$$

Finalmente, llegamos a la Ec. (18.55)

Como podemos ver, la precisión depende de p . Además, el tiempo de ejecución depende de p , con lo que lo ideal es elegir un valor de p suficientemente grande como para tener una buena precisión, pero

que este valor de p no sea demasiado grande y el algoritmo no tarde demasiado. Tomemos c como un parámetro y veamos diferentes casos:

- Si tomamos $2^p = c\sqrt{N}$, el error de nuestra estimación de M esta acotado por $\frac{2\pi}{c}\sqrt{M} + \frac{\pi^2}{c^2}$ siempre que $|f - \tilde{f}| < 1$. Esto recuerda a encontrar la respuesta hasta unas pocas desviaciones estándar.
- Si nos conformamos con tener un error *relativo* pequeño, podemos correr el algoritmo para sucesivos valores de p hasta que \tilde{f} sea razonablemente grande. Esto sucederá cuando $2^p = c\sqrt{N/M}$. Después de un tiempo proporcional a $\sqrt{N/M}$, esto nos dará una estimación para M que probablemente estén dentro de un factor $(1 + \pi/c)^2$ de la respuesta correcta.
- Si queremos que el error *absoluto* esté probablemente limitado por una constante, aplicamos el algoritmo una vez para $2^p = c\sqrt{N}$ con el objetivo de estimar M . Entonces, ejecutamos otra vez, pero esta vez con $2^p = c\sqrt{MN}$. De acuerdo con la Ec. (18.55), pero suponiendo $2^p = c\sqrt{MN}$ por simplicidad, el error resultante en nuestra segunda estimación de M es probable que esté acotado por $\frac{2\pi}{c} + \frac{\pi^2}{c^2M}$. En particular, obtenemos una solución exacta, siempre que $|f - \tilde{f}| < 1$, si tomamos $c \geq 14$ ya que $\frac{2\pi}{c} + \frac{\pi^2}{c^2M} < \frac{1}{2}$ en ese caso. (Obsérvese que si aplicaciones sucesivamente el algoritmo de Grover y vamos tachamos las soluciones a medida que se encuentran también nos proporcionará un recuento exacto con alta probabilidad en un tiempo en $\mathcal{O}(\sqrt{MN})$, pero con un consumo enorme de memoria. Ver [32].)
- Finalmente, comentar que si estamos en el caso en el que el número de soluciones es un cuadrado perfecto pequeño, podemos encontrar el valor exacto en un tiempo $\mathcal{O}(\sqrt{N})$ con una probabilidad de error muy pequeña.

Para más detalles sobre el tema, puede verse [32].

18.6. Consideraciones sobre la implementación

18.6.1. Creación de un difusor U_{Ψ_0} .

Vamos a ver como podemos hacer para crear de forma genérica un difusor de la forma de (18.25). Refrescando un poco la memoria, el difusor es un operador que realiza una reflexión respecto al estado inicial $|\Psi_0\rangle$, es decir, **le cambia el signo a las componentes perpendiculares a $|\Psi_0\rangle$** . Lo que vamos a hacer para construir el difusor es, en realidad, construir un operador que **le cambia el signo a las componentes paralelas a $|\Psi_0\rangle$** , es decir, vamos a implementar $-U_{\Psi_0}$.

Empecemos definiendo la familia de **operadores de reflexión** S_A

$$S_A|i\rangle = \begin{cases} |i\rangle & \text{si } i \notin A \\ -|i\rangle & \text{si } i \in A \end{cases} \quad (18.57)$$

Es fácil ver que podemos escribir tanto el oráculo como el difusor en función de los operadores S_A

$$U_\omega = S_\omega \quad U_{\Psi_0} = -S_{\Psi_0} \quad (18.58)$$

Nosotros lo que vamos a construir y implementar es S_{Ψ_0} , no U_{Ψ_0}

18.6.1.1. Caso con $N = 2^n$.

En el caso en el que el estado inicial es una superposición uniforme de la forma (18.2) podemos construir el difusor teniendo en cuenta que

$$|\Psi_0\rangle = H^{\otimes n}|00\dots 0\rangle \Rightarrow H^{\otimes n}|\Psi_0\rangle = |00\dots 0\rangle. \quad (18.59)$$

Viendo esta propiedad, podemos darnos cuenta de que si aplicamos el operador de Walsh-Hadamard $H^{\otimes n}$ a la salida del oráculo lo que obtenemos es el estado $|00\dots 0\rangle$ más una serie de estado que corresponderán a los cambios respecto al estado inicial que ha realizado el oráculo. Lo que tenemos que hacer para aplicar el difusor es cambiarle el signo al estado $|00\dots 0\rangle$ (aplicar S_0) y volver a aplicar $H^{\otimes n}$ para deshacer los cambios introducidos por la última aplicación el mismo. Es decir, el difusor será de la forma

$$U_{\Psi_0} = -S_{\Psi_0} = -H^{\otimes n} S_0 H^{\otimes n} \quad (18.60)$$

Podemos construir S_0 a partir de la **puerta multicontrolada Z (MCZ)**

$$MCZ = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & 0 & -1 \end{pmatrix} \quad (18.61)$$

que lo que hace es cambiarle el signo al estado $|2^n - 1\rangle = |11\dots 1\rangle$. Tenemos pues

$$S_0 = X^{\otimes n} (MCZ) X^{\otimes n} \quad (18.62)$$

La puerta $X^{\otimes n}$ (que consiste en aplicar puertas X a todos los qubits) lo que hace es

$$\begin{aligned} |00\dots 0\rangle &\rightarrow |11\dots 1\rangle \left[|0\rangle \rightarrow |2^n - 1\rangle \right] \\ |11\dots 1\rangle &\rightarrow |00\dots 0\rangle \left[|2^n - 1\rangle \rightarrow |0\rangle \right] \end{aligned}$$

De esta forma, lo que hace S_0 es:

1. Aplicar la puerta $X^{\otimes n}$ para cambiar el estado $|00\dots 0\rangle$ por el estado $|11\dots 1\rangle$. (Ver la Nota de abajo)
2. Aplicar la puerta MCZ con la que cambiamos el signo a $|11\dots 1\rangle$
3. Aplicar la puerta $X^{\otimes n}$ para deshacer los cambios del primer paso. De esta forma, el único cambio real es el del signo del estado $|0\rangle = |00\dots 0\rangle$.

Nota

En realidad la puerta $X^{\otimes n}$ afecta a todos los estados (no solo a $|00\dots 0\rangle$ y $|11\dots 1\rangle$). Sin embargo, como es su propia inversa ($X^{\otimes n} X^{\otimes n} = I$) y como entre la primera y la segunda aplicación de $X^{\otimes n}$ lo único que hacemos es cambiarle al signo a $|11\dots 1\rangle$, todos los cambios se deshacen menos este signo, que pasa a estar en el estado $|00\dots 0\rangle$.

Como ya comentamos, el operador que se implementa es $-U_{\Psi_0}$, es decir

$$\boxed{-U_{\Psi_0} = S_{\Psi_0} = H^{\otimes n} S_0 H^{\otimes n} = H^{\otimes n} X^{\otimes n} (MCZ) X^{\otimes n} H^{\otimes n}} \Rightarrow \quad (18.63)$$

$$\boxed{G_{imple} = -U_{\Psi_0} U_{\omega} = -H^{\otimes n} S_0 H^{\otimes n} U_{\omega}}$$

18.6.1.2. Caso con $N \neq 2^n$.

Las limitación que hemos impuesto hasta ahora diciendo que N debía ser una potencia de 2 viene de la transformación de Walsh-Hadamard

$$H^{\otimes n}|j\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} (-1)^{i \cdot j} |i\rangle, \quad (\text{donde } i \cdot j \text{ denota el producto escalar binario}) \quad (18.64)$$

Esta transformación, que se usa para generar el estado inicial y en el difusor, no está bien definida si no se cumple que $N = 2^n$.

Esta condición puede ser relajada si **sustituimos la transformación de Walsh-Hadamard por cualquier otra transformación unitario F que cumpla**

$$F|0\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \quad (18.65)$$

y seguiremos teniendo interacciones de Grover válidas aplicando

$$FS_0F^{-1}U_\omega \quad (18.66)$$

De hecho, cuando estamos en el caso $N = 2^n$, la Walsh-Hadamard es simplemente la elección más sencilla de F . Para el caso en el que N no es una potencia de dos, podemos usar la transformación de Fourier de Kitaev [33]. Puede verse también [34]

18.7. Distribución de probabilidad inicial aleatoria

En esta sección vamos a ver una generalización del algoritmo de Grover para el caso en el que partimos de una distribución de probabilidad aleatoria (no uniforme). Veremos como este algoritmo generalizado sigue requiriendo un número de iteraciones $\mathcal{O}(\sqrt{N/M})$, aunque veremos también que hay ciertos casos particularmente desfavorables donde no podemos encontrar ninguna solución.

Este caso es de especial interés si tenemos en cuenta que muchas veces hay errores al aplicar las puertas. De esta forma, podemos tener errores en el paso de inicialización de la distribución uniforme y acabar con una distribución que se distancia un poco de esta. Como veremos a continuación, el algoritmo de Grover sigue funcionando en presencia de estos errores modestos.

En la siguientes subsecciones presentaremos el algoritmo y derivaremos las ecuaciones diferenciales que rigen la evolución de las amplitudes. Usaremos la solución (exacta) de las mismas para calcular la probabilidad de éxito y analizaremos la diferente casuística. Como el cálculo, aunque simple, puede hacerse pesado, se incluye al final un resumen de con las conclusiones importantes (sección 18.7.4).

18.7.1. Algoritmo

En realidad, el algoritmo no tiene ningún misterio. Consiste simplemente en saltarse al paso de la inicialización con las puertas de Hadammard y partir del estado con una distribución aleatoria:

1. Utilizar cualquier distribución inicial, por ejemplo, el estado final de cualquier otro algoritmo cuántico (no inicializar el sistema con la distribución uniforme).
2. Aplicar el operador de Grover T veces (calcularemos T). El operador de Grover al que nos referimos aquí es el mismo de las anteriores secciones (con la Walsh-Hadamard o con una trasformación de la forma (18.65)). No cambia nada en ese sentido.
3. Medir el resultado

18.7.2. Evolución de las amplitudes (M soluciones)

Analicemos la evolución de las amplitudes en el algoritmo modificado. Supongamos (como siempre) que tenemos N estados y M soluciones. Asumamos, sin perdida de generalidad, que $0 \leq M \leq N/2$. El estado total ahora será de la forma

$$|\Psi(t)\rangle = \sum_{i \in \omega} k_i(t)|i\rangle + \sum_{i \in \omega^\perp} l_i(t)|i\rangle. \quad (18.67)$$

Nótese que ahora las amplitudes tenemos un índices i que corresponde al estado al que acompaña. Esto es fácil de entender: como la distribución de partida ahora no es uniforme, cada estado tendrá su amplitud. Las medias de las amplitudes de los estados solución ($i \in \omega$) y no solución ($i \in \omega^\perp$) son

$$\bar{k}(t) = \frac{1}{M} \sum_{i \in \omega} k_i(t) \quad \bar{l}(t) = \frac{1}{N-M} \sum_{i \in \omega^\perp} l_i(t). \quad (18.68)$$

Una de las conclusiones clave del desarrollo que vamos a ver a continuación es que *la dinámica dictada por el algoritmo de Grover se puede describir por completo basándose en la dependencia de las medias de las amplitudes con las iteraciones.*

18.7.2.1. Derivación de las ecuaciones diferenciales

Comencemos viendo la expresión de la media total de las amplitudes el estado $|\Psi(t)\rangle$

$$\bar{\psi}(t) = \frac{1}{N} [M\bar{k}(t) + (N-M)\bar{l}(t)], \quad (18.69)$$

y definamos la cantidad $C(t)$

$$C(t) = \frac{2}{N} [(N-M)\bar{l}(t) - M\bar{k}(t)]. \quad (18.70)$$

Veamos como evolucionan las medias de las amplitudes con cada iteracións:

- Después a aplicar el oráculo sobre el esto $|\Psi(t)\rangle$ lo que sucede es el signo de la amplitud de los estados solución se invierte, con lo que

$$\begin{cases} \bar{k}(t) \rightarrow \bar{k}'(t) = -\bar{k}(t) \\ \bar{l}(t) \rightarrow \bar{l}'(t) = \bar{l}(t) \end{cases} \Rightarrow \overline{U_\omega \psi}(t) = \frac{1}{N} [-M\bar{k}(t) + (N-M)\bar{l}(t)] = \frac{C(t)}{2}.$$

- Después a aplicar el difusor sobre el esto $|\Psi(t)\rangle$ lo que sucede es una reflexión respecto a la media $\overline{U_\omega \psi}(t)$. Esto se pude expresar de la siguiente manera

$$\begin{aligned} k'_i(t) &\rightarrow k''_i(t) = 2\overline{U_\omega \psi}(t) - k'_i(t) = C(t) - k'_i(t) = \boxed{C(t) + k_i(t) = k_i(t+1)} \\ l'_i(t) &\rightarrow l''_i(t) = 2\overline{U_\omega \psi}(t) - l'_i(t) = C(t) - l'_i(t) = \boxed{C(t) - l_i(t) = l_i(t+1)}. \end{aligned} \quad (18.71)$$

Promediando

$$\begin{aligned} \bar{k}(t+1) &= C_j + \bar{k}(t) \\ \bar{l}(t+1) &= C_j - \bar{l}(t) \end{aligned} \quad (18.72)$$

Las Ecs. (18.72) son las ecuaciones diferenciales que rigen la evolución de la media de las amplitudes.

18.7.2.2. Soluciones de las ecuaciones diferenciales

Vamos a resolver de forma analítica las Ecs. (18.72) para ver la evolución exacta de las medias de las amplitudes de los estados solución y no solución. Procedemos a resolver las fórmulas de recursión para condiciones iniciales complejas arbitrarias. Empecemos definiendo las funciones

$$\begin{aligned} f_+(t) &= \bar{l}(t) + i\sqrt{\frac{M}{N-M}}\bar{k}(t) \\ f_-(t) &= \bar{l}(t) - i\sqrt{\frac{M}{N-M}}\bar{k}(t). \end{aligned} \quad (18.73)$$

Vemos que no son más que un **cambio de variables** para facilitarnos la resolución de las ecuaciones. Empleando este cambio de variables podemos reescribir las Ecs. (18.72) de la siguiente forma

$$\begin{aligned} f_+(t+1) &= e^{i\beta} f_+(t) \\ f_-(t+1) &= e^{-i\beta} f_-(t), \end{aligned} \quad (18.74)$$

donde β es **real** y cumple

$$\boxed{\cos \beta = 1 - 2 \frac{M}{N}}. \quad (18.75)$$

Vemos que ahora la solución de las ecuaciones es trivial:

$$\begin{aligned} f_+(t) &= e^{i\beta t} f_+(0) \\ f_-(t) &= e^{-i\beta t} f_-(0). \end{aligned} \quad (18.76)$$

Vemos claramente que $|f_+(0)|$ y $|f_-(0)|$ son independientes de la iteración t en la que estemos. Podemos ahora deshacer el cambio de variable

$$\begin{aligned} \bar{k}(t) &= -i \sqrt{\frac{N-M}{4M}} [e^{i\beta t} f_+(0) - e^{-i\beta t} f_-(0)] \\ \bar{l}(t) &= \frac{1}{2} [e^{i\beta t} f_+(0) - e^{-i\beta t} f_-(0)]. \end{aligned} \quad (18.77)$$

18.7.3. Propiedades de las soluciones: Probabilidad de acierto.

18.7.3.1. Evolución de las amplitudes en función de la evolución de las medias.

Lo primero que vamos a hacer es ver que, efectivamente, solo con la evolución de las medias podemos describir la evolución. Para ello, restemos a las Ecs. (18.71) a las Ecs. (18.72)

$$\begin{aligned} k_i(t+1) - \bar{k}(t+1) &= k_i(t) - \bar{k}(t) \\ l_i(t+1) - \bar{l}(t+1) &= -[l_i(t) - \bar{l}(t)]. \end{aligned}$$

Esto significa que las cantidades

$$\begin{aligned} \Delta k_i &\equiv k_i(0) - \bar{k}(0) \\ \Delta l_i &\equiv l_i(0) - \bar{l}(0) \end{aligned} \quad (18.78)$$

son *constantes del movimiento*. Esto nos permite simplificar las expresiones para la dependencia temporal de las amplitudes:

$$\begin{aligned} k_i(t) &= \bar{k}(t) + \Delta k_i \\ l_i(t) &= \bar{l}(t) + (-1)^t \Delta l_i. \end{aligned} \quad (18.79)$$

Vemos que la distribución de las amplitudes de los estados solución $k_i(t)$ respecto a su medias $\bar{k}(t)$ es constante. Es decir, las amplitudes varían al unísono entorno a la medias. De esta forma, con saber las distribución inicial respecto a las media (Δk_i) podemos describir la evolución de las amplitudes solo conociendo la evolución de las medias. Lo mismo sucede para los estados no solución $l_i(t)$, salvo que estos se invierten entorno a su media en cada iteración.

De las Ecs. (18.79) se ve inmediatamente que las varianzas

$$\begin{aligned} \sigma_k^2(t) &= \frac{1}{M} \sum_{i \in \omega} |k_i(t) - \bar{k}(t)|^2 \\ \sigma_l^2(t) &= \frac{1}{N-M} \sum_{i \in \omega^\perp} |l_i(t) - \bar{l}(t)|^2 \end{aligned}$$

son independientes del tiempo [$\sigma_k^2(t) = \sigma_k^2(0)$ y $\sigma_l^2(t) = \sigma_l^2(0) \forall t$].

18.7.3.2. Probabilidad de acierto.

Para facilitar el análisis de las soluciones, definamos las variables (complejas) α y ϕ de la forma

$$\alpha = \sqrt{f_+(0)f_-(0)}, \quad e^{2i\phi} = f_+(0)/f_-(0). \quad (18.80)$$

Reescribamos las Ecs. (18.77) en función de estas variables

$$\boxed{\bar{k}(t) = \sqrt{\frac{N-M}{M}}\alpha \sin(\beta t + \phi)}$$

$$\boxed{\bar{l}(t) = \alpha \cos(\beta t + \phi)} \quad (18.81)$$

Esto nos permite ver que hay una diferencia de fase de $\pi/2$ entre las medias de las amplitudes de los estados solución y los no solución. Se ve fácilmente que cuando una es máxima, la otra es mínima.

La probabilidad de medir alguno de los estados solución será $P(t) = \sum_{i \in \omega} |k_i(t)|^2$. Como todos los operadores son unitarios, las amplitudes cumplen la condición de normalización:

$$\sum_{i \in \omega} |k_i(t)|^2 + \sum_{i \in \omega^\perp} |l_i(t)|^2 = 1 \quad \forall t \quad (18.82)$$

Usando la identidad

$$\overline{(y - \bar{y})^2} = \overline{y^2} - \bar{y}^2 \quad \Rightarrow \quad \sigma_y^2 = \overline{y^2} - \bar{y}^2 \quad \Rightarrow \quad \overline{y^2} = \sigma_y^2 + \bar{y}^2 \quad (18.83)$$

podemos escribir

$$\overline{|k_i(t)|^2} = \frac{1}{M} \sum_{i \in \omega} |k_i(t)|^2 = \sigma_k^2 + |\bar{k}(t)|^2$$

$$\overline{|l_i(t)|^2} = \frac{1}{N-M} \sum_{i \in \omega} |l_i(t)|^2 = \sigma_l^2 + |\bar{l}(t)|^2$$

Despejando $\sum_{i \in \omega} |k_i(t)|^2$ en (18.82), sustituyendo en $P(t)$, usando la expresión anterior de $\sum_{i \in \omega} |l_i(t)|^2$ y desarrollando con cuidado, puede verse que

$$\boxed{P(t) = P_{av} - \Delta P \cos 2 [\beta t + \operatorname{Re}(\phi)]}, \quad (18.84)$$

donde

$$P_{av} = 1 - (N-M)\sigma_l^2 - \frac{1}{2} \left[(N-r)|\bar{l}(0)|^2 + r|\bar{k}(0)|^2 \right]$$

$$\Delta P = \frac{1}{2} \left| (N-M)\bar{l}(0)^2 + M\bar{k}(0)^2 \right|$$

18.7.3.3. Número óptimo, T , de iteraciones.

El valor máximo de la probabilidad que se puede obtener durante la evolución del algoritmo es

$$P_{max} = P_{av} + \Delta P \quad (18.85)$$

Dada una distribución arbitraria inicial de M estados solución y $N-M$ estados no solución, con medias $\bar{k}(0)$ y $\bar{l}(0)$ respectivamente, el valor máximo de la probabilidad P_{max} se alcanzará cuando realicemos T iteraciones tal que

$$\cos 2 [\beta T + \operatorname{Re}(\phi)] = -1 \quad \Rightarrow \quad \boxed{T = [(u + 1/2)\pi - \operatorname{Re}(\phi)] / \beta} \quad (18.86)$$

con $u = 0, 1, 2, \dots$. Una importante conclusión es que para determinar el valor óptimo de iteraciones, todo lo que necesitamos es conocer *las medias iniciales de las amplitudes y el número de estados marcados*.

Si M es pequeño tenemos que $1 - 2M/N \approx 1$, así que podemos aproximar β a segundo orden en la Ec. (18.75)

$$\cos \beta \approx 1 - \frac{1}{2}\beta^2 = 1 - 2\frac{M}{N} \quad \Rightarrow \quad \beta = 2\sqrt{\frac{M}{N}} \quad (18.87)$$

Así que tenemos que T es del orden de $\mathcal{O}(\sqrt{N/M})$ (para $u = 0$).

18.7.3.4. Casos particulares.

El valor máximo de la probabilidad puede variar mucho dependiendo de las propiedades estáticas (medias y varianzas) de la distribución inicial de amplitudes. Como es lógico, cuanto mayor sea P_{max} menos repeticiones del algoritmo tendremos que hacer para encontrar una solución (donde cada repetición son m iteraciones). Es fácil darse cuenta de que el número esperado de repeticiones del algoritmo hasta encontrar un estado solución es $1/P_{max}$.

Nota

Porque *repeticiones del algoritmo*? En el párrafo anterior llamamos repeticiones del algoritmo a realizar varias veces las m iteraciones necesarias para maximizar la probabilidad.

Si la probabilidad máxima es menor de uno, tenemos entonces la posibilidad de tras realizar las m iteraciones no midamos un estado solución. Si esto sucede, como al medir se destruye el estado, lo que hay que hacer es volver a empezar: volver a partir de la distribución inicial y aplicar m iteraciones. En promedio, tendremos que repetir este proceso $1/P_{max}$ veces. Vemos que si la probabilidad máxima es, por ejemplo, $1/4$, tendremos que medir en promedio 4 veces para obtener un estado solución.

Veamos los diferentes casos:

- Cuando el ratio $\bar{l}(0)/\bar{k}(0)$ es real, puede verse fácilmente que $|f_+(0)| = |f_-(0)|$. En este caso $P_{max} = 1 - (N - r)\sigma_l^2$. El mejor caso, aquel con $P_{max} = 1$, se obtiene cuando $\sigma_l = 0$, es decir, cuando la distribución inicial es uniforme.
- Cuando tenemos $f_+(0) = 0$ o $f_-(0) = 0$, el algoritmo es inútil, pues $P(t)$ es constante.
- El peor caso se da cuando $\sigma_k^2 = f_+(0) = f_-(0) = \bar{k}(0) = \bar{l}(0) = 0$ y $(N - r)\sigma_l^2 = 1$. En este caso tenemos $P_{max} = P(t) = 0 \forall t$, con lo que nunca encontraremos una solución.

18.7.3.5. Distribución de probabilidad desconocida.

Veamos que sucede si no conocemos de antemano las medias y las varianzas de la distribución inicial de amplitudes. La solución es más sencilla de lo que podríamos pensar en un principio. Lo único que hay que hacer es ejecutar el algoritmo dos veces, ejecutando en un caso m_1 iteraciones y en el otro m_2 iteraciones, tal que $m_2 - m_1 = \pi/(2\beta)$. De la Ec. (18.84) está claro que al menos en uno de los dos casos vamos a tener $P(t) \geq P_{av} \geq P_{max}/2$. En este caso, necesitamos el doble de repeticiones para obtener al menos la mitad de probabilidades de éxito que cuando se conoce el tiempo de medición óptimo. Vemos además que lo único que necesitamos conocer en este caso es β , que según la Ec. (18.75) podemos calcularlo a partir del número de soluciones M .

18.7.4. Resumen de la sección.

Uno de los puntos importantes del análisis de esta sección es el hecho de demostrar que el algoritmo de Grover funciona incluso en el caso en el que tenemos pequeños errores a la hora de generar la distribución de probabilidad inicial. En realidad, en esta sección no se presenta ningún algoritmo nuevo. Simplemente se analiza que pasa si usamos el algoritmo de Grover donde en el primer paso sustituimos la inicialización de la distribución uniforme por una distribución no uniforme.

Otra de las conclusiones clave del desarrollo es que la dinámica dictada por el algoritmo de Grover se puede describir por completo basándose en la dependencia de las medias de las amplitudes $\bar{k}(t)$ y $\bar{l}(t)$ (definidas en la Ec. (18.68)).

Para estudiar la dependencia con las iteraciones de las medias $\bar{k}(t)$ y $\bar{l}(t)$, en la sección 18.7.2 deducimos las ecuaciones diferenciales que rigen su evolución (Ecs. (18.72)) y las resolvemos. Para ello primero hacemos un cambio de variable de $\bar{k}(t)$ y $\bar{l}(t)$ a $f_+(t)$ y $f_-(t)$ (Ecs. (18.73)). Resolvemos usando estas variables y deshacemos el cambio. Nos quedan pues $\bar{k}(t)$ y $\bar{l}(t)$ en función de $f_+(0)$, $f_-(0)$ y β (definido β en la Ec. (18.75), que solo depende de N y M).

Una vez tenemos las soluciones, en las Ecs. (18.79) demostramos que las amplitudes mantienen su distribución respecto a las medias invariante (con una salvedad en $\bar{l}(t)$). De esta forma, confirmamos que solo tenemos que conocer la distribución inicial respecto a las medias para describir la evolución de las amplitudes usando solo la evolución de las medias.

En la sección 18.7.3 definimos dos parámetros más, α y ϕ (Ec. (18.80)), que depende de $f_+(0)$ y $f_-(0)$. Tras unas líneas de cálculo podemos llegar a la Ec. (18.84) que nos da la probabilidad $P(t)$ de medir un estado solución en función del número de iteraciones. Una vez tenemos esta expresión, en la sección 18.7.3 buscamos el número T de iteraciones que nos maximizan la probabilidad (Ec. (18.86)). Una importante conclusión es que para determinar el valor óptimo de iteraciones, todo lo que necesitamos es conocer las medias iniciales de las amplitudes y el número de estados marcados. Concluimos también que si M es pequeño, tenemos que T es del orden de $\mathcal{O}(\sqrt{N/M})$.

Finalmente, analizamos casos particulares de distribuciones iniciales favorables y desfavorables (sección 18.7.3) y vemos que pasa si no conocemos la distribución de probabilidad inicial. Es este último caso, nuevamente podemos hallar una solución en tiempo $\mathcal{O}(N/M)$ conociendo solo M .

18.8. Implementaciones con qiskit.

18.8.1. Puerta multicontrolada Z (MCZ).

Vamos a empezar viendo la implementación de la puerta multicontrolada Z (MCZ) en qiskit, pues la usaremos bastante en las siguientes secciones. Esta puerta podemos construirla a partir de la puerta multicontrolada Toffoli (MCT) de forma muy sencilla. Para ello, recordemos que la MCT no es más que una CNOT (es decir, una puerta X) con varios controles y recordemos también la propiedad

$$HXH = Z. \quad (18.88)$$

Podemos pues construir la MCZ aplicando puertas de Hadammard en el qúbit objetivo de la MCT antes y después de la misma.

Jupyter Notebook: [13. Algoritmo de Grover](#)

Ver la sección [13.1. Puerta multicontrolada \$Z\$ \(MCZ\)](#) del Notebook [13. Algoritmo de Grover](#).
El Notebook puede descargarse de [Github](#).

18.8.2. Difusor genérico.

Como ya comentamos, habitualmente en vez de implementar U_{Ψ_0} implementamos $-U_{\Psi_0}$. Vimos además que podemos hacer la implementación mediante transformadas de Walsh-Hadamard $H^{\otimes n}$, puertas $X^{\otimes n}$ y la puerta MCZ , es decir

$$-U_{\Psi_0} = S_{\Psi_0} = H^{\otimes n}S_0H^{\otimes n} = H^{\otimes n}X^{\otimes n}(MCZ)X^{\otimes n}H^{\otimes n} \quad (18.89)$$

Jupyter Notebook: [13. Algoritmo de Grover](#)

Ver la sección del [13.2. Difusor genérico](#) notebook [13. Algoritmo de Grover](#).

El Notebook puede descargarse de [Github](#).

18.8.3. Oráculo “trivial”.

Vamos a presentar en esta sección un código de qiskit para construir un oráculo que cambie el signo de los estados que nosotros le digamos. Este es uno de esos ejemplo típicos que se plantean cuando se habla de Grover, esos en los que sabemos con antelación los estados concretos que queremos buscar. De esta forma, el oráculo que construimos está hecho “ad hoc” para marcar ciertos estado.

(Precisamente elegí llamarle a este caso “trivial” porque no entraña ningún misterio, sino que como comento es un caso académico.)

Jupyter Notebook: [13. Algoritmo de Grover](#)

Ver la sección [13.3. Oráculo Trivial](#) del Notebook [13. Algoritmo de Grover](#).

El Notebook puede descargarse de [Github](#).

18.8.4. Oráculos que verifican condiciones: sudoku 2×2 .

En esta sección vamos a ver como se puede usar el algoritmo de Grover para buscar cadenas de bits que satisfacen unas ciertas condiciones. En concreto vamos a ver dos caso: como solucionar un **sudoku** 2×2

Jupyter Notebook: [13. Algoritmo de Grover](#)

Ver la sección [13.4. Oráculos que verifican condiciones: Sudoku \$2 \times 2\$](#) del Notebook [13. Algoritmo de Grover](#).

El Notebook puede descargarse de [Github](#).

Capítulo 19

Criptografía y Quantum Key Distribution (QKD)

19.1. Introducción

Desde los orígenes de la escritura ya se tenía claro que no existen canales de comunicación seguros. Por ejemplo, si se envía un jinete con un mensaje, este puede ser interceptado por el enemigo, asaltándolo y robándole el mensaje. La única solución viable es **encriptar el mensaje**, para que en caso de que se interceptado no puedan leerlo. Los orígenes de la criptografía se remontan a la antigüedad Babilónica donde el mensaje se enrollaba en un cilindro de un radio adecuado que permitía que las letras se alineasen de manera correcta. La **clave**, por tanto, era el radio del cilindro. Sólo el poseedor del cilindro correcto podía descifrar el mensaje. En el imperio romano se usaron las llamadas *reglas de sustitución*, en las que todas las letras se remplazaban por otras de acuerdo con una regla secreta (cifrado del Cesar). Por ejemplo la letra siguiente del abecedario, o a una cierta distancia fija. La regla de sustitución se denomina **clave criptográfica** y la sustitución directa en inversa se conocen como **cifrado y descifrado**. Las correlaciones que existen entre el mensaje original y el cifrado permiten casi siempre adivinar la regla de sustitución, lo que se denomina **romper un código**. Notar que éste método requiere **compartir una clave secreta**. A partir de ahí, los mensajes se pueden compartir por un **canal inseguro**.

Esto que parece tan lejano, en realidad muy común a día de hoy para nosotros con las telecomunicaciones: en Internet es imposible enviar un mensaje sin que este sea susceptible de ser interceptado por una tercera persona. Además, estos mensajes pueden contener información muy sensible, como por ejemplo las claves de acceso a la aplicación del banco. Es decir, con la aparición de Internet, la criptografía se convirtió en pan de cada día.

Cuando hablamos de **Quantum Key Distribution (QKD)** hablamos del hecho de usar la física cuántica para enviar las claves criptográficas entre los dos participantes de la conversación. La gracia aquí está en que las leyes de la física cuántica nos ayudan a saber cuando la clave ha sido interceptada, pudiendo así abortar el protocolo y probar otra vez a enviar las claves. Una vez que los dos participantes tienen las claves, ya pueden empezar a comunicarse por un canal inseguro.

Cuando hablamos de criptografía, siempre se suelen usar tres personajes ficticios: Alice, Bob y Eve. Alice quiere enviarle un mensaje a Bob, mientras que Eve intenta leer el mensaje.

Este capítulo se basa en gran medida en el artículo recopilatorio [35] de 2009, así como en la sección 12.6.3 del Nielsen-Chuang [9].

19.1.1. Criptografía

La **criptografía** es un campo de aplicaciones que proporciona privacidad, autenticación y confidencialidad a los usuarios. Un subcampo importante es el de la **comunicación segura**, cuyo objetivo es permitir la comunicación confidencial entre distintas partes de forma que ninguna parte no autorizada tenga acceso al contenido de los mensajes. Este campo tiene una larga historia de éxitos y fracasos, ya que a lo largo de los siglos surgieron muchos métodos para codificar mensajes, y los códigos siempre se rompían tiempo después.

Sin embargo, la historia no tiene por qué repetirse eternamente. En 1917, Vernam inventó el llamado cifrado *one time pad*, que utiliza una clave secreta simétrica y aleatoria compartida entre emisor y receptor [36]. En principio, este esquema no puede romperse, siempre que las partes no reutilicen su clave. Tres décadas más tarde, Shannon demostró que el esquema Vernam es óptimo: no existe ningún método de cifrado que requiera menos clave [37]. Esto significa que la clave se agota en el proceso. Para emplear este esquema, por tanto, las partes comunicantes deben disponer de un método seguro para compartir una clave que sea tan larga como el texto que se desea cifrar. Debido a esta limitación, que se agrava cuando hay que transmitir grandes cantidades de información de forma segura, la mayoría de las aplicaciones criptográficas actuales se basan en otros esquemas, cuya seguridad no puede demostrarse en principio, sino que se basa más bien en nuestra experiencia de que algunos problemas son difíciles de resolver. En otras palabras, estos esquemas pueden romperse, pero con una cantidad sustancial de potencia de cálculo.

19.1.2. Criptografía de clave privada

Hasta la invención de la **criptografía de clave pública** en los años 70, todos los criptosistemas funcionaban según un principio diferente, conocido ahora como **criptografía de clave privada**. En un criptosistema de clave privada, si Alice desea enviar mensajes a Bob, debe tener una **clave de codificación**, que le permita cifrar su mensaje, y Bob debe tener una **clave de descodificación** equivalente, que le permita descifrar el mensaje cifrado. Un criptosistema de clave privada sencillo pero muy eficaz es el **cifrado de Vernam**, a veces llamado *one time pad*. Alice y Bob comienzan con cadenas de claves secretas de n bits, que son idénticas. Alice codifica su mensaje de n bits sumando el mensaje y la clave, y Bob descodifica restando para invertir la codificación.

La gran característica de este sistema es que, mientras las cadenas de claves sean realmente secretas, es **demostrablemente seguro**. Es decir, cuando el protocolo utilizado por Alice y Bob tiene éxito, lo hace con una probabilidad arbitrariamente alta. Y para cualquier estrategia de escucha empleada por Eve, Alice y Bob pueden garantizar que la información mutua de Eve con su mensaje no codificado puede hacerse tan pequeña como se desee. Por el contrario, la seguridad de la criptografía de clave pública se basa en *suposiciones matemáticas no demostradas sobre la dificultad de resolver ciertos problemas* como la factorización (¡con ordenadores clásicos!), a pesar de ser ampliamente utilizada y más conveniente.

La principal dificultad de los criptosistemas de clave privada es la **distribución segura de los bits de clave**. En concreto, el cifrado de Vernam sólo es seguro si el número de bits de clave es al menos tan grande como el tamaño del mensaje que se codifica, y los bits de clave no pueden reutilizarse. Por tanto, la gran cantidad de bits de clave necesarios hace que estos sistemas sean poco prácticos para el uso general. Además, los bits clave deben entregarse con antelación, custodiarse asiduamente hasta su uso y destruirse después; de lo contrario, en principio, esa información clásica puede copiarse sin alterar los originales, lo que compromete la seguridad de todo el protocolo. A pesar de estos inconvenientes, los criptosistemas de clave privada como el cifrado de Vernam siguen utilizándose por su seguridad demostrable, con material clave entregado mediante reuniones clandestinas, mensajeros de confianza o enlaces de comunicación privados y seguros.

19.1.2.1. Cifrado de Vernam (One time pad)

El cifrado de Vernam es un protocolo en tres pasos

1. El mensaje se codifica en forma de un número binario $s_i = 0100101110$.
2. La clave secreta es una secuencia binaria aleatoria de longitud al menos tan grande como el mensaje $c_i = 0101111001$. La codificación consiste en la suma módulo dos de ambas secuencias: $\tilde{s}_i = s_i \oplus c_i = 0001010111$.
3. La de-codificación consiste en la suma módulo dos de nuevo de la clave compartida con el mensaje cifrado $s_i = \tilde{s}_i \oplus c_i = s_i \oplus 2c_i = 0001010111$.

Por un lado, el texto cifrado es indescifrable porque la clave es aleatoria. Por tanto no transmite ninguna correlación entre los dígitos del mensaje original.

19.1.2.2. Distribución secreta de claves

El (cifrado de Vernam) fue inventado por el matemático Gilbert Vernam en 1917 [36], y es el primer ejemplo de código irrompible. La prueba de su inviolabilidad se debe a Claude Shannon treinta años más tarde [37], y depende crucialmente de tres detalles: 1) que la clave sea aleatoria, 2) que la clave sea secreta (sólo conocida por Bob y Alice), y 3) que la clave sólo se utilice una vez (one-time pad).

Sólo cuando existe la seguridad absoluta de que la clave no ha sido interceptada por un tercer agente (Eve), Alice y Bob pueden confiar en la imposibilidad de nadie para descifrar su mensaje. Desde un punto de vista práctico, Alice y Bob poseen un banco de claves secretas que han intercambiado en algún momento del pasado, y van gastándolas a razón de una por cada mensaje. En este sentido, el cifrado de Vernam, cambia el centro de gravedad del problema de la transmisión segura, a la **distribución segura de claves**. Este es precisamente el punto donde entran los algoritmos de QKD.

19.1.3. Criptografía de clave pública

El criptosistema de clave pública requiere una función tipo puerta trampa, f , fácil pero que tenga una inversa, f^{-1} , difícil de calcular. Fácil y difícil se refieren al nivel de complejidad computacional requerido. En general diremos que un problema es fácil si el tiempo de cálculo escala polinómicamente (clase P), como n^a con el número n de letras del mensaje. Un problema difícil lo hace de manera no-polinómica (clase NP). Como ya se ha comentado, la seguridad de los protocolos de clave pública no se puede demostrar, sino que simplemente se basa en la dificultad a la hora de invertir la función.

En el sistema de clave pública Alice y Bob no intercambian ninguna clave. Bob tiene dos claves, una para encriptar y otra para desencriptar. Publica la primera, y Alice con ella encripta su mensaje. Una vez recibido, Bob usa la clave de desencriptar para averiguar el contenido. Quien no posea la clave privada de Bob, aún puede intentar averiguar la operación inversa a la encriptación que efectuó Alice con la clave pública. Este proceso debe ser muy costoso en términos de tiempo para que la clave sea considerada segura (es decir, lo que comentamos de calcular f^{-1})

19.1.3.1. El protocolo RSA

El protocolo RSA es el criptosistema más utilizado en transacciones comerciales, o para la verificación de identidad. Inventado en 1977 por Rivest, Shamir y Adleman. Funciona con el siguiente conjunto de pasos

1. Bob escoge dos numeros primos p y q y calcula $N = pq$ y $L = (p - 1)(q - 1)$.
2. Escoge e (de encriptación), un número impar pequeño, coprimo con L .
3. Calcula d (de desencriptación), el inverso módulo L de e . Es decir

$$e \cdot d \bmod L = 1 \tag{19.1}$$

Esta tarea es fácil, conociendo e y L .

4.
 - La clave pública es (e, N) y cualquiera puede usarla para mandar mensajes a Bob.
 - La clave privada es (d, N) y, a priori, sólo es conocida por Bob.
5. Para encriptar, Alice divide su mensaje en bloques, $i = 1, 2, \dots$, de manera que cada bloque pueda ser identificado por un número $m_i < N$. La clave permite una *regla de sustitución*

$$m_i \rightarrow \tilde{m}_i = (m_i)^e \pmod{N} \quad (19.2)$$

es decir el resto mod(N) de exponentiar m_i a la potencia e .

6. Para desencriptar, Bob realiza la operación inversa sobre \tilde{m}_i . Se puede probar que $(\tilde{m}_i)^{ed} \pmod{N} = m_i$. Por tanto la operación que realiza Bob es simplemente

$$\tilde{m}_i = (\tilde{m}_i)^d \pmod{N} \quad (19.3)$$

Las dos ventajas principales del sistema criptográfico público son

- No necesitan de una distribución secreta de clave.
- Se puede reutilizar la misma clave repetidamente.

La fortaleza de la clave RSA crece con la magnitud del número N . El código puede romperse si alguien obtiene los factores primos $pq = N$. Esto permite hallar $L = (p-1)(q-1)$ y, como e es público, obtener d fácilmente. La fiabilidad del sistema reside en la inexistencia de métodos para hallar p y q tarea en un tiempo polinómico en $n = \log_2 N$. El más eficiente, método de criba, y requiere un tiempo que crece como $\exp(n^{1/3}(\log n)^{2/3})$. El algoritmo de Shor permite resolver el problema en un tiempo polinómico $\mathcal{O}(n^c)$.

19.2. Fundamentos de QKD

El panorama ha cambiado en las últimas décadas, gracias a las inesperadas aportaciones de la física cuántica. A principios de los años 80, Bennett y Brassard propusieron una solución al problema de la distribución de claves basada en la física cuántica [38]; esta idea, redescubierta independientemente por Ekert unos años más tarde [39], fue el comienzo de la **Quantum Key Distribution (QKD)**.

En un intrigante avance independiente, diez años después de la aparición de la QKD, Peter Shor descubrió que, en principio, los números grandes pueden factorizarse eficientemente si se pueden realizar manipulaciones coherentes en muchos sistemas cuánticos (ver capítulo 16). La factorización de grandes números es un ejemplo de tarea matemática considerada clásicamente difícil de resolver y, por este motivo, está relacionada con una clase de esquemas criptográficos muy utilizados en la actualidad (el protocolo RSA visto anteriormente). Aunque todavía no se han construido ordenadores cuánticos con suficientes qubits como para factorizar claves del tamaño de las que se usan en el protocolo RSA, la computación cuántica representa una amenaza para este tipo de criptografía.

La QKD es un protocolo de seguridad demostrable que permite crear bits de clave privada entre dos partes a través de un canal público. Los bits de clave pueden utilizarse para aplicar un criptosistema clásico de clave privada que permita a las partes comunicarse de forma segura. El único requisito del protocolo QKD es que los qubits puedan comunicarse a través del canal público con una tasa de error inferior a un determinado umbral. La seguridad de la clave resultante está garantizada por las propiedades de la información cuántica y, por tanto, sólo está condicionada a que las leyes fundamentales de la física sean correctas.

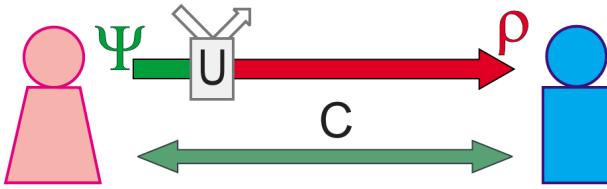


Figura 19.1: El escenario de QKD: Alice y Bob están conectados por un canal cuántico, en el que Eve puede intervenir sin más restricción que las leyes de la física; y por un canal clásico autenticado, en el que Eve sólo puede escuchar.

19.2.1. Un poco de historia

La QKD se desarrolló con la presentación del primer protocolo completo por parte de Bennett y Brassard en 1984 [38], el **BB84**, basado en ideas anteriores de Wiesner [40]. En el protocolo BB84, los bits se codifican en dos bases complementarias de un sistema de dos niveles (qúbit); este qúbit es enviado por Alice a Bob, que lo mide. El teorema de no clonación se menciona explícitamente como la razón de la seguridad. Estos trabajos se publicaron en actas de congresos y fueron prácticamente desconocidos para la comunidad de físicos. No fue hasta 1991, cuando Artur Ekert, independientemente de los desarrollos anteriores, publicó un artículo sobre distribuciones cuánticas de claves (el algoritmo **E91**) que el campo ganó popularidad rápidamente [39]. El argumento de Ekert a favor de la seguridad tenía un sabor diferente: un fisgón introduce “elementos de realidad” en las correlaciones compartidas por Alice y Bob; así, si observan correlaciones que violan una desigualdad de Bell, la comunicación no puede haber sido completamente rota por Eve. Poco después, Bennett, Brassard y Mermin argumentaron que los protocolos basados en el entrelazamiento, como el E91, son equivalentes a los protocolos de preparación y medida, como el protocolo BB84 [41].

El año 1992 fue testigo de otros dos hitos: la invención del protocolo B92 [42] y la primera demostración experimental [43]. Se puede concluir razonablemente el periodo fundacional de la QKD con la definición de la amplificación de la privacidad, el postprocesamiento clásico necesario para borrar la información de Eve de la clave bruta [44].

Con el paso de los años el campo de la QKD práctica ha ido creciendo en amplitud y madurez. Se han propuesto nuevas familias de protocolos, en particular **protocolos de variable continua** [45–50] y los más recientes **protocolos de referencia de fase distribuida** (*distributed-phase-reference protocols*) [51, 52].

19.2.2. Conceptos generales

La configuración genérica de QKD se muestra en la Fig. 1. Los dos interlocutores autorizados, aquellos que desean establecer una clave secreta a distancia, se denominan tradicionalmente Alice y Bob. Necesitan estar conectados por dos canales: un canal cuántico, que les permite compartir señales cuánticas, y un canal clásico, por el que pueden enviar mensajes clásicos de ida y vuelta.

El canal clásico necesita autenticación: esto significa que Alice y Bob se identifican; una tercera persona puede escuchar la conversación pero no participar en ella. El canal cuántico, sin embargo, está abierto a cualquier posible manipulación por parte de un tercero. En concreto, la tarea de Alice y Bob es garantizar la seguridad contra un espía adversario, normalmente llamado Eve, que se cuela en el canal cuántico y escucha los intercambios en el canal clásico.

Por garantía de seguridad entendemos que nunca se utiliza una clave no secreta: o bien los socios autorizados pueden crear una clave secreta (una lista común de bits secretos que sólo ellos conocen) o bien abortan el protocolo. Por lo tanto, tras la transmisión de una secuencia de símbolos, Alice y

Bob deben estimar cuánta información sobre sus listas de bits se ha filtrado a Eve. Esta estimación es obviamente imposible en la comunicación clásica: si alguien está interviniendo una línea telefónica, la comunicación continúa sin modificaciones. Aquí es donde entra en juego la física cuántica: en un canal cuántico, la fuga de información está cuantitativamente relacionada con la degradación de la comunicación. A continuación profundizamos un poco más en las razones físicas de esta afirmación.

19.2.3. El origen de la seguridad

El origen de la seguridad de la QKD se remonta a algunos principios fundamentales de la física cuántica. Se puede argumentar, por ejemplo, que cualquier acción mediante la cual Eve extrae alguna información de los estados cuánticos es una forma generalizada de **medición**; y un principio bien conocido de la física cuántica dice que la medición en general modifica el estado del sistema medido. Alternativamente, se puede pensar que el objetivo de Eve es tener una copia perfecta del estado que Alice envía a Bob; esto, sin embargo, está prohibido por el teorema de no clonación (ver sección 8.5), que afirma que no se puede duplicar un estado cuántico desconocido manteniendo intacto el original. Ambos argumentos ya aparecían en el artículo seminal de Bennett y Brassard [38].

Para los algoritmos que se basan en pares entrelazados (como el de Ekert de 1991 [39]) se puede invocar un tercer argumento físico, que suele considerarse más un hecho que un principio, pero muy profundo: las correlaciones cuánticas obtenidas por mediciones separadas en miembros de pares entrelazados violan las desigualdades del seno de Bell y, por tanto, no pueden haber sido creadas por un acuerdo preestablecido. En otras palabras, *los resultados de las mediciones no existían antes de las mediciones*; pero entonces, en particular, Eve no podía conocerlos [39]. Este argumento supone que la QKD se implementa con estados entrelazados.

El hecho de que la seguridad pueda basarse en principios generales de la física sugiere la posibilidad de una **seguridad incondicional** (*unconditional security*, es decir, la posibilidad de garantizar la seguridad sin imponer ninguna restricción al poder del fisgón. De hecho, actualmente se ha demostrado la seguridad incondicional de varios protocolos QKD.

19.2.4. La elección de la luz

En general, el procesamiento cuántico de la información puede aplicarse con cualquier sistema y, de hecho, existen propuestas para aplicar la informática cuántica con iones, átomos, luz, espines, etc. En abstracto, este es también el caso de la QKD: se podría imaginar realizar un experimento de QKD con electrones, iones y moléculas; sin embargo, la luz es la única opción práctica.

Ahora bien, como es bien sabido, la luz no interactúa fácilmente con la materia; por tanto, los estados cuánticos de la luz pueden transmitirse a lugares distantes básicamente sin decoherencia, en el sentido de que se espera una pequeña perturbación en la definición del modo óptico. El problema de la luz es la **dispersión**, es decir, las pérdidas: muy a menudo, los fotones simplemente no llegan. En primer lugar, las pérdidas imponen límites a la tasa de claves secretas (que no puede escalar con la distancia mejor que la transmisividad de la línea y la distancia alcanzable) cuando las pérdidas son tan grandes que la señal se pierde en eventos espurios, los recuentos oscuros". En segundo lugar, las pérdidas pueden filtrar información al fisgón, según la naturaleza de la señal cuántica: en el caso de los pulsos coherentes, sin duda; en el de los fotones individuales, no; el caso de los haces entrelazados es más sutil. Una tercera diferencia básica viene determinada por el esquema de detección. Las implementaciones que utilizan contadores de fotones se basan en la postselección: si un fotón no llega, el detector no hace clic y el evento simplemente se descarta. Por el contrario, las implementaciones que utilizan **detección homodina** siempre dan una señal; por lo tanto, las pérdidas se traducen como ruido adicional.

En resumen, la QKD siempre se implementa con luz y no hay razón para creer que las cosas vayan a cambiar en el futuro. En consecuencia, el canal cuántico es cualquier medio que propague la luz con

pérdidas razonables: normalmente, una fibra óptica o simplemente el espacio libre siempre que Alice y Bob tengan una línea de visión.

19.2.5. Tratamiento cuántico de la información: marcos P&M y EB

19.2.5.1. Marco P&M

El primer paso de un protocolo QKD es el intercambio y la medición de señales en el canal cuántico. El papel de Alice es codificar: el protocolo debe especificar qué estado cuántico $|\Psi(S_n)\rangle$ codifica para la secuencia de n símbolos $S_n = \{s_1, \dots, s_n\}$. En muchos protocolos, pero no en todos, el estado $|\Psi(S_n)\rangle$ tiene la forma de un producto tensorial $|\psi(s_1)\rangle \otimes \dots \otimes |\psi(s_n)\rangle$. En todos los casos, es crucial que el protocolo utilice un conjunto de estados **no ortogonales**, ya que, de lo contrario, Eve podría descodificar la secuencia sin introducir errores midiendo en la base adecuada (en otras palabras, un conjunto de estados ortogonales puede clonarse perfectamente). La razón de usar estados no ortogonales viene dada por la siguiente proposición:

Proposición 1 (*La ganancia de información implica perturbaciones*) *En cualquier intento de distinguir entre dos estados cuánticos no ortogonales, la ganancia de información sólo es posible a costa de introducir perturbaciones en la señal.*

Demostración: Sean $|\psi\rangle$ y $|\varphi\rangle$ los estados cuánticos no ortogonales sobre los que Eve intenta obtener información. Podemos suponer sin pérdida de generalidad que el proceso que utiliza para obtener información es interactuar unitariamente el estado ($|\psi\rangle$ o $|\varphi\rangle$) con una ancilla preparada en un estado estándar $|u\rangle$. Suponiendo que este proceso no perturba los estados, en los dos casos se obtiene

$$\begin{aligned} |\psi\rangle|u\rangle &\rightarrow |\psi\rangle|v\rangle \\ |\varphi\rangle|u\rangle &\rightarrow |\varphi\rangle|v'\rangle \end{aligned} \quad (19.4)$$

A Eve le gustaría que $|v\rangle$ y $|v'\rangle$ fueran diferentes, de tal forma que pudiera adquirir información sobre la identidad de los estados. Sin embargo, como el producto interno se conserva respecto a las transformaciones unitarias, debe de seguirse que

$$\langle v|v'\rangle\langle\psi|\varphi\rangle = \langle u|u\rangle\langle\psi|\varphi\rangle \Rightarrow \langle v|v'\rangle = \langle u|u\rangle = 1 \quad (19.5)$$

lo que implica que $|v\rangle$ y $|v'\rangle$ deben de ser idénticos. Así, distinguir entre $|\psi\rangle$ y $|\varphi\rangle$ debe perturbar inevitablemente al menos uno de estos estados. ■

Aprovechamos esta idea transmitiendo estados de qubits no ortogonales entre Alice y Bob. Al comprobar si hay perturbaciones en sus estados transmitidos, establecen un límite superior para cualquier ruido o escucha que se produzca en su canal de comunicación. Por lo tanto, el papel de Bob es doble: sus mediciones permiten, por supuesto, descodificar la señal, pero también una estimación de la pérdida de coherencia cuántica y, por tanto, de la información de Eve. Para que esto sea posible, deben utilizarse mediciones no compatibles.

19.2.5.2. Marco EB

Hemos descrito la codificación cuántica de los protocolos QKD con el lenguaje de los esquemas de preparación y medida (*prepare-and-measure*, **P&M**): Alice elige activamente la secuencia S_n que quiere enviar, prepara el estado $|\Psi(S_n)\rangle$, y lo envía a Bob, que realiza alguna medida. Cualquier esquema de este tipo puede traducirse inmediatamente en un esquema basado en el entrelazamiento (*entanglement-based*, **EB**): Alice prepara el estado entrelazado

$$|\Phi^n\rangle = \frac{1}{\sqrt{d_n}} \sum_{S_n} |S_n\rangle_A \otimes |\Psi(S_n)\rangle_B \quad (19.6)$$

donde d_n es el número de posibles secuencias S_n y $|S_n\rangle_A$ forma una base ortogonal. Midiendo en esta base, Alice aprende un S_n y prepara el correspondiente $|\Psi(S_n)\rangle$ en el subsistema que se envía a Bob: desde el punto de vista de Bob, nada cambia. Esta traducción formal no significa que ambas realizaciones sean igualmente prácticas o incluso factibles con la tecnología actual. Sin embargo, implica que la prueba de seguridad para el protocolo EB se traduce inmediatamente al protocolo P&M correspondiente y viceversa.

19.3. Protocolos de QKD: Tres familias

El número de protocolos QKD explícitos es prácticamente infinito: después de todo, Bennett ha demostrado que se puede obtener seguridad cuando un bit se codifica en sólo dos estados cuánticos no ortogonales [41]. Pero, de hecho, esta posible variedad se ha cristalizado en tres familias principales: codificación de **variable discreta** (sección 19.3.1), codificación de **variable continua** (sección 19.3.2) y, más recientemente, codificación **distributed-phase-reference** (sección ??). La diferencia crucial es el esquema de detección: la codificación discreta-variable y la codificación distribuida-fase-referencia utilizan el **recuento de fotones** y postseleccinan los eventos en los que se ha producido efectivamente una detección, mientras que la codificación continua-variable se define por el uso de **detección homodina**.

La codificación variable discreta es la original. Su principal ventaja es que los protocolos pueden diseñarse de forma que, en ausencia de errores, Alice y Bob comparten inmediatamente una clave secreta perfecta. Siguen siendo los protocolos QKD más frecuentemente implementados. Los argumentos a favor de la codificación de variable continua se derivan de la observación de que los contadores de fotones presentan normalmente bajas eficiencias cuánticas, altas tasas de recuento oscuro y tiempos muertos bastante largos, mientras que estos inconvenientes pueden superarse utilizando la detección homodina. El precio que hay que pagar es que el protocolo proporciona a Alice y Bob una realización correlacionada pero bastante ruidosa de una variable aleatoria continua, porque las pérdidas se traducen en ruido: como consecuencia, hay que utilizar una cantidad significativa de corrección de errores. En cuanto a la codificación distributed-phase-reference, su origen se encuentra en los esfuerzos de algunos grupos experimentales hacia una implementación más práctica. Desde el punto de vista de la detección, estos protocolos producen un resultado de variable discreta; pero la naturaleza de las señales cuánticas es muy diferente a la de la codificación de variable discreta, y esto motiva un tratamiento aparte.

A pesar de las diferencias originadas por el uso de un dispositivo de detección distinto, existe una fuerte unidad conceptual subyacente en la QKD discreta y continua. Por poner sólo un ejemplo, en ambos casos la capacidad de distribuir una clave cuántica está estrechamente relacionada con la capacidad de distribuir entrelazamiento, independientemente del esquema de detección utilizado e incluso si no hay entrelazamiento real. Estas similitudes no son muy sorprendentes, ya que se sabe desde hace tiempo que las características cuánticas de la luz pueden revelarse mediante el recuento de fotones (por ejemplo, experimentos antibunching o anticorrelación) o mediante la detección homodina (por ejemplo, experimentos de squeezing). Dado que la QKD es una técnica que explota estas características cuánticas de la luz, no hay razón para que se restrinja al régimen de recuento de fotones. Sorprendentemente, al igual que el antibunching (o una fuente monofotónica) ni siquiera es necesario en la QKD basada en el recuento de fotones, el squeezing tampoco es necesario en la QKD basada en la detección homodina. La única característica cuántica que resulta necesaria es la no ortogonalidad de los estados de luz.

19.3.1. Protocolos de variable discreta

19.3.1.1. BB84-BBM

El protocolo BB84 es, no solo el más famoso de los protocolos de variable discreta, sino el más famoso

en general. Se trata también de la primera propuesta de un algoritmo de QKD y data de 1984 [38]. El correspondiente protocolo EB (entanglement-base) es conocido como BBM [41]: el protocolo E91 [39] es equivalente a él cuando se implementa con qubits. Veamos la filosofía de estos protocolos.

Supongamos que Alicia tiene una fuente de fotones individuales. Las propiedades espectrales de los fotones están claramente definidas, por lo que el único grado de libertad que queda es la polarización. Alice y Bob alinean sus polarizadores y acuerdan utilizar la base horizontal o vertical (+), o la base complementaria de polarizaciones lineales, es decir, +45/-45 (×). En concreto, la codificación de bits es

$ 0_+\rangle$,	codifica la polarización horizontal	
$ 1_+\rangle$,	codifica la polarización vertical	
$ 0_\times\rangle$,	codifica la polarización a +45°	
$ 1_\times\rangle$,	codifica la polarización a -45°	

(19.7)

Vemos que ambos valores de bit 0 y 1 se codifican de dos formas posibles, o más exactamente en estados no ortogonales porque

$$\begin{aligned} |0_\times\rangle &= \frac{1}{\sqrt{2}} (|0_+\rangle + |0_-\rangle) \\ |1_\times\rangle &= \frac{1}{\sqrt{2}} (|0_+\rangle - |0_-\rangle) \end{aligned} \quad (19.8)$$

Nota: caso con qubits

Estamos hablando de polarizaciones de la luz porque, como ya comentamos, se usan fotones para las transmisiones en el canal cuántico. Sin embargo, podríamos plantear el mismo protocolo tratando con qubit. Por ejemplo, con qubits podemos usar los siguientes cuatro estados:

$$\begin{aligned} |+x\rangle, |-x\rangle, &\quad \text{autoestados de } \sigma_x \\ |+y\rangle, |-y\rangle, &\quad \text{autoestados de } \sigma_y \end{aligned} \quad (19.9)$$

Dada esta codificación, el protocolo BB84 es el siguiente:

1. Alice prepara un fotón aleatoriamente en uno de los cuatro estados anteriores y lo envía a Bob por el canal cuántico.
2. Bob lo mide. Como Bob no conoce en qué base lo ha generado Alice, este elige aleatoriamente una de las dos bases: + o ×.
3. Los dos pasos anteriores se repiten $4N$ veces, de forma que tanto Alice como Bob tienen ahora una lista de $4N$ bits cada uno.
4. Alice y Bob se comunican por un canal clásico diciéndose qué base han usado para medir cada uno de los fotones.
5. Alice y Bob descartan aquellos bits en los cuales han medido en bases distintas. Este paso se le suele llamar **criba** (*sifting*). En promedio, habrán descartado $2N$ bits, así que les queda una lista con los otros $2N$ bits. A esto se llama la **clave bruta** (*raw key*).
6. Alice y Bob revelan ahora una muestra aleatoria de tamaño N de los bits de sus claves brutas y estiman la tasa de error en el canal cuántico, y por tanto a su vez la información de Eve, manteniendo secretos los otros N bits. En ausencia de errores, la clave bruta es idéntica para Alice y Bob, y Eve no tiene información: en este caso, los N bits secretos de la clave bruta ya son la clave secreta. Sin embargo, si hay errores, Alice y Bob tienen que corregirlos y borrar la información que Eve podría haber obtenido. Ambas tareas pueden realizarse mediante la comunicación en el canal clásico, por lo que esta parte del protocolo se denomina **postprocesamiento clásico**.

Al final de este procesamiento, Alice y Bob comparten una clave realmente secreta o nada en absoluto (si la información de Eve era demasiado grande).

Nota

Véase que las comunicaciones por el canal clásico no tiene porque estar encriptadas, pues aunque Eve conozca esa información no le servirá para romper el protocolo.

19.3.1.2. SAR04

El protocolo SARG04 [53, 54] utiliza los mismos cuatro estados Ec. (19.9) y las mismas medidas en el lado de Bob que BB84, pero el bit se codifica en la base en lugar de en el estado (la base X codifica para 0 y la base Y codifica para 1). Bob tiene que elegir sus bases con probabilidad 1/2.

La creación de la clave en bruto es ligeramente más complicada que en BB84. Supongamos por definición que Alice envía $|+x\rangle$: en ausencia de errores tenemos que:

- si Bob mide X , obtiene $s_b = +$
- si Bob mide Y , puede obtener ambos $s_b = +/ -$ con igual probabilidad

Alice sin embargo, como ha enviado ella el estado sabe que $s_a = +$. En la fase de criba, Bob revela s_b ; Alice le dice que acepte si había preparado un estado con $s_a \neq s_b$, en cuyo caso Bob acepta el bit correspondiente a la base que no ha utilizado. La razón está clara en el ejemplo anterior: en ausencia de errores, $s_b = -$ señala la base equivocada.

Nota: Ejemplo con $| -x \rangle$

Si Alice envía $| -x \rangle$ ella tiene que $s_a = -$. Sin embargo, Bob tiene los casos:

- si Bob mide X , obtiene $s_b = -$
- si Bob mide Y , puede obtener ambos $s_b = +/ -$ con igual probabilidad

En este caso, lo que señala que Bob ha medido en la base equivocada es que obtenga $s_b = +$.

SARG04 se inventó para implementaciones con fuentes láser atenuadas porque es más robusto que BB84 contra los **ataques PNS**. Se ha demostrado la seguridad incondicional de este algoritmo (ver [35]).

19.3.1.3. Otros protocolos de variable discreta

Se ha propuesto un gran número de otros protocolos de variables discretas; todos ellos tienen características que los hacen menos interesantes para la QKD práctica que BB84 o SARG04. El protocolo de seis estados [55, 56] sigue la misma estructura que BB84, al que añade la tercera base Z mutuamente inesgada definida por la matriz de Pauli σ_z . Su seguridad incondicional se demostró bastante pronto [57]. El interés de este protocolo reside en el hecho de que la estimación del canal se vuelve "tomográficamente completa", es decir, los parámetros medidos caracterizan completamente el canal. Como consecuencia, se puede tolerar más ruido que con los protocolos BB84 o SARG04. Sin embargo, el ruido es bastante bajo en las configuraciones ópticas, mientras que las pérdidas son más preocupantes. En este sentido, el protocolo de seis estados funciona peor porque requiere componentes ópticos con pérdidas adicionales. Consideraciones similares se aplican a la versión de seis estados de la codificación SARG04.

Por último, mencionamos el protocolo B92 [42], que utiliza sólo dos estados no ortogonales, cada uno de los cuales codifica un valor de bit. En términos de codificación, ésta es obviamente la posibilidad más económica. Desafortunadamente, B92 es un protocolo bastante sensible: como se observó en el artículo original, este protocolo es seguro sólo si alguna otra señal (por ejemplo, un pulso de referencia

fuerte) está presente junto con los dos estados que codifican el bit. Se ha demostrado la seguridad incondicional de las implementaciones monofotónicas [58, 59] y de algunas implementaciones con un pulso de referencia fuerte [60, 61].

19.3.2. Protocolos de variable continua

La codificación de variables discretas puede aplicarse con varias fuentes, pero requiere técnicas de recuento de fotones. Se ha sugerido un enfoque alternativo a la QKD, en el que los contadores de fotones se sustituyen por fotodiodos *p-i-n* estándar de telecomunicaciones, que son más rápidos (gigahercios en lugar de megahercios) y más eficientes (normalmente el 80 % en lugar del 10 %). Los esquemas de medición correspondientes se basan entonces en la **detección homodina** e implican la medición de datos que son amplitudes reales en lugar de eventos discretos; de ahí que estos esquemas se denominen QKD de variable continua (**CV**).

Las primeras propuestas que sugieren el uso de la detección homodina en QKD se deben a Ralph [45], Hillery [46] y Reid [62]. En particular, una versión de **estado estrangulado** (*squeezed-state*) de BB84 fue propuesta por [46], donde la elección de base de Alice consiste en seleccionar si el estado de luz enviado a Bob está estrangulado en la **cuadratura** $q = x \circ q = p$. A continuación, este q-squeezed state se desplaza en q en $+c$ o $-c$ dependiendo de un bit aleatorio elegido por Alice, donde c es una constante elegida apropiadamente. La base aleatoria elegida por Bob define si lo que se mide es la cuadratura x o p . La criba consiste simplemente en conservar sólo los casos en los que las cuadraturas elegidas por Alice y Bob coinciden. En este caso, el valor medido por Bob se distribuye según una distribución gaussiana centrada en el valor $(+c \circ -c)$ enviado por Alice. En cierto sentido, este protocolo puede considerarse "híbrido" porque los datos de Alice son binarios mientras que los de Bob son reales (con distribución gaussiana).

Estas primeras propuestas y su generalización directa se denominan **protocolos CV con modulación discreta**; al mismo tiempo, se propuso otra clase de protocolos CV que utilizan en su lugar una **modulación continua**, en particular una modulación gaussiana. Aunque los protocolos CV son mucho más recientes que los protocolos de variable discreta, sus pruebas de seguridad no han dejado de progresar en los últimos años y ahora están a punto de alcanzar un nivel comparable (ver [35]).

19.3.2.1. Squeezed-states y cuadraturas

En el contexto de la óptica cuántica y la mecánica cuántica, un "estado estrangulado" (*squeezed-states*) se refiere a un estado cuántico específico de la luz (normalmente radiación electromagnética, como la luz láser) con algunas propiedades únicas. Los squeezed-states se caracterizan por la reducción de la incertidumbre en una de las dos propiedades complementarias del campo luminoso, a menudo denominadas **cuadraturas**. Los conceptos clave son los siguientes:

1. Cuadraturas:

- En óptica cuántica, la luz puede describirse mediante dos propiedades u observables complementarias: la **posición** (x) y el **momento** (p). Se denominan variables conjugadas y representan direcciones ortogonales en el espacio de fases del sistema cuántico.
- El principio de incertidumbre, formulado por Heisenberg, establece que existe un límite fundamental en cuanto a la precisión con la que pueden conocerse simultáneamente la posición y el momento de una partícula cuántica (o, en este caso, un fotón en un campo luminoso). En otras palabras, no se pueden conocer con precisión ambas propiedades simultáneamente.

2. Squeezed-state:

- Un squeezed-state es un estado cuántico de la luz en el que la incertidumbre (desviación típica) en una de las cuadraturas (x o p) se reduce por debajo del límite de incertidumbre

de Heisenberg.

- Cuando un estado se “estruja” en una cuadratura concreta, significa que las mediciones de esa propiedad se hacen más precisas, pero a costa de aumentar la incertidumbre en la cuadratura complementaria.

Por ejemplo, si se tiene un squeezed-state en la cuadratura de posición (x), significa que se reduce la incertidumbre en la medición de la posición, lo que permite realizar mediciones de posición más precisas. Sin embargo, esto tiene el coste de aumentar la incertidumbre en la cuadratura del momento (p). A la inversa, si el estado está comprimido en la cuadratura de momento (p), permite mediciones de momento más precisas a expensas de la incertidumbre de posición.

19.3.2.2. Detección homodina

La QKD de variación continua se basa en la medición de los componentes de cuadratura de la luz. Esto puede hacerse cómodamente mediante detección óptica homodina. Este esquema de detección utiliza dos haces de la misma frecuencia: la señal y el llamado oscilador local (LO), mucho más fuerte y, por tanto, a menudo tratado como clásico. Los haces se superponen en un divisor de haz equilibrado. La intensidad de la luz en cada uno de los modos de salida se mide con detectores proporcionales y se registra la diferencia entre las photocorrientes resultantes. Si la amplitud y la fase del oscilador local son estables, la corriente diferencial transporta información sobre un componente en cuadratura de la señal de entrada.

Las intensidades se miden mediante diodos $p-i-n$, que proporcionan una alta eficiencia de detección (normalmente el 80 %) y un ruido relativamente bajo. Por tanto, la detección homodina podría funcionar en principio a velocidades de repetición de gigahercios, a diferencia de los contadores de fotones basados en APD, cuya velocidad de detección está limitada por el tiempo muerto del detector.

Desgraciadamente, el uso de una técnica de detección homodina de tan alta velocidad tiene un precio. Debido al principio de incertidumbre, la medición de cuadraturas complementarias es intrínsecamente ruidosa. El ruido de vacío (o ruido intrínseco) es el ruido que se obtiene cuando hay vacío en el puerto de señal (sólo está presente el oscilador local). Ahora bien, las inevitables pérdidas de transmisión en la línea óptica, que simplemente causan “clics perdidos” en los esquemas basados en el recuento de fotones, provocan una disminución de la relación señal/ruido en los esquemas basados en la detección homodina. El ruido del vacío es responsable de un ruido añadido bastante significativo en la QKD de variación continua, que debe corregirse durante la etapa clásica de postprocesamiento: un esfuerzo informático adicional en la QKD de variación continua.

19.3.3. Protocolos Distributed-phase-reference

Tanto los protocolos de variable discreta como los de variable continua fueron inventados por teóricos. Algunos grupos experimentales, en sus desarrollos hacia sistemas QKD prácticos, han concebido nuevos protocolos, que no encajan en las categorías anteriores. En ellos, al igual que en los protocolos de variable discreta, las claves brutas están formadas por realizaciones de una variable discreta (un bit) y ya están perfectamente correlacionadas en ausencia de errores. Sin embargo, el canal cuántico se controla utilizando las propiedades de los estados coherentes, más concretamente, observando la coherencia de fase de los impulsos subsiguientes; de ahí el nombre de protocolos distributed-phase-reference.

El primer protocolo de este tipo se ha denominado desplazamiento de fase diferencial (DPS) [51, 63]. Alice produce una secuencia de estados coherentes de la misma intensidad,

$$|\Psi(S_n)\rangle = \cdots |e^{i\varphi_{k-1}}\sqrt{\mu}\rangle |e^{i\varphi_k}\sqrt{\mu}\rangle |e^{i\varphi_{k+1}}\sqrt{\mu}\rangle \cdots \quad (19.10)$$

donde cada fase puede fijarse en $\varphi = 0$ o π (ver Fig. 19.2). Los bits se codifican en la diferencia entre

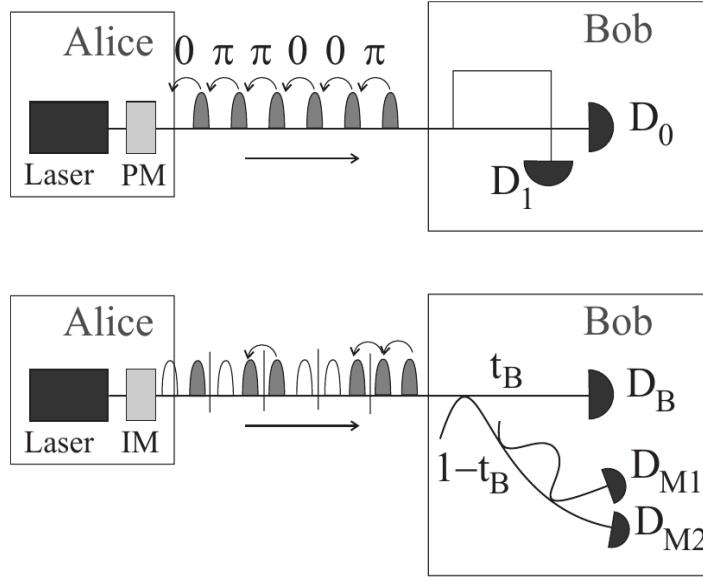


Figura 19.2: Dos formas de implementar el protocolo distributed-phase-reference: desplazamiento de fase diferencial (DPS, arriba) y coherente unidireccional (COW, abajo). Leyenda: PM, modulador de fase; IM, modulador de intensidad. Figura tomada de [35]

dos fases sucesivas: $b_k = 0$ si $e^{i\varphi_k} = e^{i\varphi_{k+1}}$ y $b_k = 1$ en caso contrario. Esto puede discriminarse inequívocamente utilizando un interferómetro desequilibrado. La complejidad en el análisis de este protocolo radica en que el k -ésimo pulso contribuye tanto al k -ésimo como al $(k + 1)$ -ésimo bit. El protocolo DPS ya ha sido objeto de varios experimentos [64–66]

En el protocolo denominado coherent one way (COW) [52, 67], cada bit se codifica en una secuencia de un impulso no vacío y otro vacío,

$$|0\rangle_k = |\sqrt{\mu}\rangle_{2k+1} |0\rangle_{2k}, \quad |1\rangle_k = |0\rangle_{2k+1} |\sqrt{\mu}\rangle_{2k}. \quad (19.11)$$

Estos dos estados pueden discriminarse inequívocamente de forma óptima midiendo el tiempo de llegada (ver Fig. 19.2).

Tanto DPS como COW son esquemas P&M, adaptados para fuentes láser. Aún no ha sido posible derivar un límite para la seguridad incondicional porque las técnicas existentes sólo se aplican cuando $|\Psi(S_n)\rangle$ puede descomponerse en señales independientes. (Esta afirmación es a fecha del 2009, pues el paper en que se basa esta sección [35] data de esa fecha.)

Bibliografía

- [1] Javier Mas Sole, “Curso de formación interna del proyecto de quantumspain,” 2022.
- [2] C. Cohen-Tannoudji and B. Diu, *Quantum Mechanics, Vol 1.* Wiley-VCH.
- [3] M. Le Bellac, *Quantum Physics.* Cambridge University Press, 2006.
- [4] S. Khatri and M. M. Wilde, “Principles of quantum communication theory: A modern approach (notes in progress,” 2020.
- [5] Wikipedia contributors, “Gram–schmidt process — Wikipedia, the free encyclopedia,” 2023. [Online; accessed 12-November-2023].
- [6] M. Fayngold and V. Fayngold, *Quantum mechanics and quantum information.* Wiley-VCH, 1982.
- [7] D. J. Griffiths and D. F. Schroeter, *Introduction to Quantum Mechanics.* Cambridge University Press, 3 ed., 2018.
- [8] Wikipedia contributors, “Kronecker product — Wikipedia, the free encyclopedia,” 2023. [Online; accessed 15-June-2023].
- [9] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition.* Cambridge University Press, 2010.
- [10] J. S. Bell, “On the einstein podolsky rosen paradox,” *Physics Physique Fizika*, vol. 1, pp. 195–200, Nov 1964.
- [11] A. Aspect, J. Dalibard, and G. Roger, “Experimental test of bell’s inequalities using time-varying analyzers,” *Phys. Rev. Lett.*, vol. 49, pp. 1804–1807, Dec 1982.
- [12] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt, “Proposed experiment to test local hidden-variable theories,” *Phys. Rev. Lett.*, vol. 23, pp. 880–884, Oct 1969.
- [13] W. K. Wootters and W. H. Zurek, “A single quantum cannot be cloned,” *Nature*, vol. 299, no. 5886, pp. 802–803, 1982.
- [14] D. Dieks, “Communication by epr devices,” *Physics Letters A*, vol. 92, no. 6, pp. 271–272, 1982.
- [15] K. S. Krane, *Introductory nuclear physics.* New York, NY: Wiley, 1988.
- [16] L. M. K. Vandersypen and I. L. Chuang, “Nmr techniques for quantum control and computation,” *Rev. Mod. Phys.*, vol. 76, pp. 1037–1069, Jan 2005.
- [17] D. W. Leung, I. L. Chuang, F. Yamaguchi, and Y. Yamamoto, “Efficient implementation of coupled logic gates for quantum computation,” *Phys. Rev. A*, vol. 61, p. 042310, Mar 2000.
- [18] J. Jones and E. Knill, “Efficient refocusing of one-spin and two-spin interactions for nmr quantum

- computation,” *Journal of Magnetic Resonance*, vol. 141, no. 2, pp. 322–325, 1999.
- [19] X. HU, R. D. SOUSA, and S. D. SARMA, “DECOHERENCE AND DEPHASING IN SPIN-BASED SOLID STATE QUANTUM COMPUTERS,” in *Foundations of Quantum Mechanics in the Light of New Technology*, WORLD SCIENTIFIC, oct 2002.
- [20] S. Aaronson and D. Gottesman, “Improved simulation of stabilizer circuits,” *Phys. Rev. A*, vol. 70, p. 052328, Nov 2004.
- [21] IONQ, “Getting started with native gates,” <https://ionq.com/docs/getting-started-with-native-gates>, 6 de abril de 2023.
- [22] D. Coppersmith, “An approximate fourier transform useful in quantum factoring,” 2002.
- [23] M. A. Nielsen and I. Chuang, “Quantum computation and quantum information,” 2002.
- [24] “Wikipedia: Euclidean algorithm.” https://en.wikipedia.org/wiki/Euclidean_algorithm.
- [25] “Wikipedia: Continued fraction.” https://en.wikipedia.org/wiki/Continued_fraction.
- [26] S. Beauregard, “Circuit for shor’s algorithm using $2n+3$ qubits,” *arXiv preprint quant-ph/0205095*, 2002.
- [27] T. G. Draper, “Addition on a quantum computer,” 2000.
- [28] M. Mosca and A. Ekert, “The hidden subgroup problem and eigenvalue estimation on a quantum computer,” 1999.
- [29] C. Zalka, “Fast versions of shor’s quantum factoring algorithm,” 1998.
- [30] S. Parker and M. B. Plenio, “Efficient factorization with a single pure qubit and $\log N$ mixed qubits,” *Phys. Rev. Lett.*, vol. 85, pp. 3049–3052, Oct 2000.
- [31] Textbook de Qiskit (IBM), “Grover’s algorithm,” 2021.
- [32] G. Brassard, P. HØyer, and A. Tapp, “Quantum counting,” in *Automata, Languages and Programming*, pp. 820–831, Springer Berlin Heidelberg, 1998.
- [33] A. Y. Kitaev, “Quantum measurements and the abelian stabilizer problem,” 1995.
- [34] L. K. Grover, “Quantum computers can search rapidly by using almost any transformation,” *Phys. Rev. Lett.*, vol. 80, pp. 4329–4332, May 1998.
- [35] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, “The security of practical quantum key distribution,” *Rev. Mod. Phys.*, vol. 81, pp. 1301–1350, Sep 2009.
- [36] G. S. Vernam, “Cipher printing telegraph systems: For secret wire and radio telegraphic communications,” *Journal of the A.I.E.E.*, vol. 45, no. 2, pp. 109–115, 1926.
- [37] C. E. Shannon, “Communication theory of secrecy systems,” *The Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, 1949.
- [38] G. Brassard and C. H. Bennett, “Quantum cryptography: Public key distribution and coin tossing,” in *International conference on computers, systems and signal processing*, pp. 175–179, 1984.
- [39] A. K. Ekert, “Quantum cryptography based on bell’s theorem,” *Phys. Rev. Lett.*, vol. 67, pp. 661–663, Aug 1991.

- [40] S. Wiesner, “Conjugate coding,” *SIGACT News*, vol. 15, p. 78–88, jan 1983.
- [41] C. H. Bennett, G. Brassard, and N. D. Mermin, “Quantum cryptography without bell’s theorem,” *Phys. Rev. Lett.*, vol. 68, pp. 557–559, Feb 1992.
- [42] C. H. Bennett, “Quantum cryptography using any two nonorthogonal states,” *Phys. Rev. Lett.*, vol. 68, pp. 3121–3124, May 1992.
- [43] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. A. Smolin, “Experimental quantum cryptography,” *J. Cryptology*, vol. 5, pp. 3–28, 1992.
- [44] C. Bennett, G. Brassard, C. Crepeau, and U. Maurer, “Generalized privacy amplification,” *IEEE Transactions on Information Theory*, vol. 41, no. 6, pp. 1915–1923, 1995.
- [45] T. C. Ralph, “Continuous variable quantum cryptography,” *Phys. Rev. A*, vol. 61, p. 010303, Dec 1999.
- [46] M. Hillery, “Quantum cryptography with squeezed states,” *Phys. Rev. A*, vol. 61, p. 022309, Jan 2000.
- [47] N. J. Cerf, M. Lévy, and G. V. Assche, “Quantum distribution of gaussian keys using squeezed states,” *Phys. Rev. A*, vol. 63, p. 052311, Apr 2001.
- [48] D. Gottesman and J. Preskill, “Secure quantum key distribution using squeezed states,” *Phys. Rev. A*, vol. 63, p. 022309, Jan 2001.
- [49] F. Grosshans and P. Grangier, “Continuous variable quantum cryptography using coherent states,” *Phys. Rev. Lett.*, vol. 88, p. 057902, Jan 2002.
- [50] C. Silberhorn, T. C. Ralph, N. Lütkenhaus, and G. Leuchs, “Continuous variable quantum cryptography: Beating the 3 db loss limit,” *Phys. Rev. Lett.*, vol. 89, p. 167901, Sep 2002.
- [51] K. Inoue, E. Waks, and Y. Yamamoto, “Differential phase shift quantum key distribution,” *Phys. Rev. Lett.*, vol. 89, p. 037902, Jun 2002.
- [52] D. Stucki, N. Brunner, N. Gisin, V. Scarani, and H. Zbinden, “Fast and simple one-way quantum key distribution,” *Applied Physics Letters*, vol. 87, p. 194108, 11 2005.
- [53] A. Acín, N. Gisin, and V. Scarani, “Coherent-pulse implementations of quantum cryptography protocols resistant to photon-number-splitting attacks,” *Phys. Rev. A*, vol. 69, p. 012309, Jan 2004.
- [54] V. Scarani, A. Acín, G. Ribordy, and N. Gisin, “Quantum cryptography protocols robust against photon number splitting attacks for weak laser pulse implementations,” *Phys. Rev. Lett.*, vol. 92, p. 057901, Feb 2004.
- [55] D. Bruß, “Optimal eavesdropping in quantum cryptography with six states,” *Phys. Rev. Lett.*, vol. 81, pp. 3018–3021, Oct 1998.
- [56] H. Bechmann-Pasquinucci and N. Gisin, “Incoherent and coherent eavesdropping in the six-state protocol of quantum cryptography,” *Phys. Rev. A*, vol. 59, pp. 4238–4248, Jun 1999.
- [57] H.-K. Lo, “Proof of unconditional security of six-state quantum key distribution scheme,” *arXiv preprint quant-ph/0102138*, 2001.
- [58] K. Tamaki, M. Koashi, and N. Imoto, “Unconditionally secure key distribution based on two nonorthogonal states,” *Phys. Rev. Lett.*, vol. 90, p. 167904, Apr 2003.

- [59] K. Tamaki and N. Lütkenhaus, “Unconditional security of the bennett 1992 quantum key-distribution protocol over a lossy and noisy channel,” *Phys. Rev. A*, vol. 69, p. 032316, Mar 2004.
- [60] M. Koashi, “Unconditional security of coherent-state quantum key distribution with a strong phase-reference pulse,” *Phys. Rev. Lett.*, vol. 93, p. 120501, Sep 2004.
- [61] K. Tamaki, N. Lütkenhaus, M. Koashi, and J. Batuwantudawe, “Unconditional security of the bennett 1992 quantum-key-distribution scheme with a strong reference pulse,” *Physical Review A*, vol. 80, sep 2009.
- [62] M. D. Reid, “Quantum cryptography with a predetermined key, using continuous-variable einstein-podolsky-rosen correlations,” *Phys. Rev. A*, vol. 62, p. 062308, Nov 2000.
- [63] K. Inoue, E. Waks, and Y. Yamamoto, “Differential-phase-shift quantum key distribution using coherent light,” *Phys. Rev. A*, vol. 68, p. 022317, Aug 2003.
- [64] H. Takesue, E. Diamanti, T. Honjo, C. Langrock, M. M. Fejer, K. Inoue, and Y. Yamamoto, “Differential phase shift quantum key distribution experiment over 105 km fibre,” *New Journal of Physics*, vol. 7, pp. 232–232, nov 2005.
- [65] H. Takesue, S. W. Nam, Q. Zhang, R. H. Hadfield, T. Honjo, K. Tamaki, and Y. Yamamoto, “Quantum key distribution over a 40-dB channel loss using superconducting single-photon detectors,” *Nature Photonics*, vol. 1, pp. 343–348, jun 2007.
- [66] E. Diamanti, H. Takesue, C. Langrock, M. M. Fejer, and Y. Yamamoto, “100 km differential phase shift quantum key distribution experiment with low jitter up-conversion detectors,” *Opt. Express*, vol. 14, pp. 13073–13082, Dec 2006.
- [67] N. Gisin, G. Ribordy, H. Zbinden, D. Stucki, N. Brunner, and V. Scarani, “Towards practical and fast quantum cryptography,” 2004.
- [68] “Textbook ibm: Quantum phase estimation.” <https://learn.qiskit.org/course/ch-algorithms/quantum-phase-estimation>.
- [69] “Textbook ibm: Quantum fourier transform.” <https://learn.qiskit.org/course/ch-algorithms/quantum-fourier-transform>.
- [70] “Textbook ibm: Shor’s algorithm.” <https://learn.qiskit.org/course/ch-algorithms/shors-algorithm>.
- [71] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, “Tight bounds on quantum searching,” *Fortschritte der Physik*, vol. 46, pp. 493–505, jun 1998.