FACULTAD DE CIENCIAS

TRABAJO FIN DE GRADO

Grado en Matemáticas

Corrección de errores cuánticos mediante códigos topológicos.

Autora: Silvia Muñoz Nogales
Tutor: Diego Ruano
Año 2025

ÍNDICE GENERAL

INTRODUCCIÓN

En las últimas décadas, la computación cuántica ha emergido como una de las fronteras más prometedoras y desafiantes de la ciencia y la tecnología. Basada en los principios de la mecánica cuántica, esta nueva forma de computación ofrece la posibilidad de resolver ciertos problemas de una manera mucho más eficiente que los ordenadores clásicos, abriendo la puerta a avances en criptografía, simulación de sistemas complejos, optimización y muchos otros campos. Sin embargo, la computación cuántica enfrenta un reto fundamental: la fragilidad de los cúbits frente al ruido y los errores. Las operaciones cuánticas son extremadamente sensibles a las perturbaciones externas y al mismo proceso de medición, lo que limita la fiabilidad de los cálculos cuánticos a gran escala.

Para superar esta limitación, la corrección de errores cuánticos se ha convertido en un campo de estudio esencial. Los códigos correctores cuánticos buscan proteger la información almacenada en cúbits mediante técnicas de redundancia y codificación, análogas en ciertos aspectos a los códigos correctores clásicos, pero adaptadas a las propiedades únicas de los sistemas cuánticos, como la superposición y el entrelazamiento. Entre los diferentes métodos de corrección de errores cuánticos, los códigos topológicos han ganado especial relevancia debido a su potencial para lograr una corrección de errores eficiente y escalable. Este tipo de códigos se basa en una codificación que distribuye la información en una estructura espacial, lo que permite detectar y corregir errores de manera robusta sin necesidad de una gran cantidad de cúbits físicos adicionales.

Dentro de los códigos topológicos, el código tórico de Kitaev se destaca como uno de los modelos teóricos más influyentes. Propuesto por el físico Alexei Kitaev, este código representa la información cuántica en una red bidimensional, lo que permite que los errores se identifiquen y corrijan mediante operaciones locales en la red. La estructura toroidal del código es clave, ya que facilita la implementación de cúbits lógicos, que son una abstracción de cúbits resistentes a errores. Además, el código tórico es uno de los modelos fundamentales para entender la computación cuántica topológica, un enfoque que promete realizar operaciones cuánticas sin necesidad de medidas repetidas ni de corrección activa, aprovechando la robustez natural de los sistemas topológicos.

Este trabajo se centra en los fundamentos de la computación cuántica y la corrección de errores, avanzando desde los conceptos básicos hasta los códigos correctores cuánticos, con especial énfasis en los códigos estabilizadores, CSS, topológicos y el código tórico de Kitaev. En el primer capítulo, se introducen los conceptos esenciales de la computación cuántica, como el cúbit, las puertas lógicas cuánticas, la medición y la representación de estados en la esfera de Bloch. Posteriormente, se abordan la matriz de densidad y los superoperadores, que son herramientas matemáticas fundamentales para describir y manipular sistemas cuánticos en presencia de ruido. A continuación, se exploran los códigos correctores clásicos y cuánticos, haciendo hincapié en la necesidad de corregir errores en sistemas cuánticos y en cómo las propiedades cuánticas demandan técnicas únicas de corrección.

Se introducen los códigos cuánticos primero con los ejemplos más sencillos y se continúa con la construcción de códigos mediante estabilizadores y los códigos de Calderbank-Shor-Steane (CSS)

Finalmente, el trabajo culmina en el estudio de los códigos topológicos, donde se introduce el código tórico de Kitaev como un ejemplo de la vanguardia en corrección de errores cuánticos y computación cuántica robusta. Este código no solo representa una estrategia avanzada para la corrección de errores, sino que también abre la puerta a la exploración de la computación cuántica topológica, una de las propuestas más prometedoras para construir ordenadores cuánticos estables y escalables.

1. CONCEPTOS GENERALES DE COMPUTACIÓN CUÁNTICA

En este cápitulo presentamos definiciones y resultados sobre los aspectos básicos de la computación cuántica que nos permitirán comprender los resultados que se muestren a continuación. Sin embargo, debido a la extensión del tema no se darán algunas demostraciones. Las referencias usadas han sido el libro *Quantum computing and Quantum information* [1], en especial para este capítulo el capítulo 4; y el material introductorio enfocado a la programación de circuitos cuánticos de la empresa canadiense dedicada a la computación cuántica Xanadu [2].

1.1. Definiciones básicas

Para entender la computación cuántica es necesario empezar por los cimientos, es decir, a partir de los cuales, se va construyendo conceptos más complejos.

1.1.1. Introducción al cúbit

Un cúbit es la unidad mínima de información en computación cuántica, lo que conceptualmente se correspondería con un bit en la computación clásica. Independientemente de la plataforma física en la que se encuentre, un cúbit es un objeto matemático que se rige por las leyes de la mecánica cuántica y tiene dos estados propios, $|0\rangle$ y $|1\rangle$.
En esencia, el marco matemático de la computación cuántica es el álgebra lineal lo que nos permite representar y manipular dichos estados de forma sencilla y comprensible sin entrar en detalles físicos.

Definición 1. Cúbit.
Un cúbit es definido de forma algebraica mediante un vector dentro de un espacio de Hilbert complejo de dimensión 2. Sus estados básicos son :

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ y } \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Debido a que trabajar con vectores escritos de esta forma es algo tedioso a la hora de manipular estados, se utiliza la siguiente notación:

Notación 1. Notación de Dirac o Bra-ket.

El vector que representa el estado de un cúbit se denomina “ket” y su notación es $|\cdot\rangle$, escribiendo entre medias el nombre correspondiente al estado, de modo que los dos estados básicos del cúbit que mencionamos en la definición 1 se escribirían como sigue:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ y } |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Es común ver la notación genérica $|\psi\rangle \in C^2$ para designar un estado cualquiera.
Por otra parte, cada “ket” tiene asociado un “bra” que se corresponde con el adjunto de dicho vector, es decir, el conjugado transpuesto. El “bra” se representa como sigue:

$$\langle 0| = \begin{pmatrix} 1 & 0 \end{pmatrix} \text{ y } \langle 1| = \begin{pmatrix} 0 & 1 \end{pmatrix}$$

1.1.2. Superposición

Como hemos mencionado, los estados $|0\rangle$ y $|1\rangle$ son los estados básicos ó fundamentales en los que puede encontrarse un cúbit, lo que nos lleva a la siguiente pregunta de forma natural: ¿Existen más estados a parte de estos?
La respuesta es que sí existen más estados, de hecho existe un número infinito de estos que se encuentran entre el $|0\rangle$ y el $|1\rangle$. Lo que nos lleva a la siguiente pregunta lógica: ¿Cómo se definen dichos estados?

Definición 2. Estados de superposición. Los estados de superposición se forman a partir de combinaciones lineales de los estados fundamentales $|0\rangle$ y $|1\rangle$. Sea $|\psi\rangle$ un estado de superposición, entonces $|\psi\rangle$ es de la forma:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \quad \alpha, \beta \in \mathbb{C} \text{ tal que } |\alpha|^2 + |\beta|^2 = 1$$

Estos α y β son llamados amplitudes ó amplitudes de probabilidad.

El principio fundamental de la mecánica cuántica es el principio de superposición que establece que un sistema cuántico puede existir en múltiples estados al mismo tiempo, en lugar de estar en un estado definido. Estos estados se combinan de manera lineal, y solo cuando se mide el sistema se colapsa en uno de los estados posibles, con probabilidades asociadas a cada estado. Este principio es fundamental para entender fenómenos como la naturaleza probabilística de la mecánica cuántica.
Es por esto que α y β reciben el nombre de amplitudes de probabilidad, ya que representan las probabilidades asociadas a los estados $|0\rangle$ y $|1\rangle$ respectivamente. Así mismo, esto también nos hace entender el por qué de la condición $|\alpha|^2 + |\beta|^2 = 1$ dada en la definición 2, ya que la probabilidad de ambos estados debe sumar 1, y se tiene que:

$$\text{Probabilidad de medir } 0 = |\alpha|^2$$

$$\text{Probabilidad de medir } 1 = |\beta|^2$$

1.1.3. Producto interno

Dada la definición algebraica que hemos dado de un cúbit como vector en un espacio complejo de Hilbert de dimensión 2 (que denotaremos desde ahora como V), es necesario definir también su producto interno.

Definición 3. Producto interno. El producto interno de dos cúbits se corresponde con una función $\langle\cdot,\cdot\rangle : V \times V \longrightarrow \mathbb{C}$. Dicha función conocida como el producto escalar usual viene definida de la siguiente forma para dos estados genéricos dados $|v\rangle = \alpha_1 |0\rangle + \beta_1 |1\rangle$ y $|w\rangle = \alpha_2 |0\rangle + \beta_2 |1\rangle$ (Donde α^* indica el conjugado de α):

$$\langle |v\rangle , |w\rangle\rangle := \begin{pmatrix} \alpha_1^* & \beta_1^* \end{pmatrix} \begin{pmatrix} \alpha_2 \\ \beta_2 \end{pmatrix} = \langle v| |w\rangle$$

Ejemplo 1. El producto interno de $|0\rangle$ y $|1\rangle$:

$$\langle 0| |1\rangle = \begin{pmatrix} 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = 0$$

Notación 2. Notación del producto interno La forma usual de escribir este producto es la siguiente:

$$\langle 0|1\rangle = \langle 0| |1\rangle$$

1.1.4. Fase relativa y fase global.

Sea $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ un estado genérico, si escribimos los números complejos α y β de forma polar obtendremos:

$$\alpha = ae^{i\theta} \text{ y } \beta = be^{i\varphi} \quad , \text{con } a, b \in \mathbb{R}$$

De modo que podemos reescribir el estado como:

$$|\psi\rangle = ae^{i\theta} |0\rangle + be^{i\varphi} |1\rangle = e^{i\theta}(a |0\rangle + be^{i(\varphi-\theta)} |1\rangle) = e^{i\theta}(a |0\rangle + be^{i\phi} |1\rangle)$$

Con $\phi = \varphi - \theta$. De esta manera podemos observar cómo el término $e^{i\theta}$ no afecta al cálculo de las amplitudes de probabilidad que hemos definido anteriormente, y por tanto sin pérdida de generalidad se puede ignorar esta fase global y describir dicho estado de la forma siguiente:

$$|\psi\rangle = a |0\rangle + be^{i\phi} |1\rangle$$

Donde $e^{i\phi}$ es conocido como la fase relativa.

1.1.5. Base computacional

Lo que diferencia a un cúbit de un bit clásico, es que como hemos visto antes, un cúbit se puede encontrar en superposición de dos estados base, $|0\rangle$ y $|1\rangle$. Por lo tanto, el estado genérico $|\psi\rangle$ se corresponde a un elemento perteneciente a un espacio vectorial complejo de dimensión 2. Como bien sabemos, los vectores $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ y $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ constituyen una base ortogonal del espacio vectorial complejo de dimensión 2, análogamente en la computación cuántica encontramos el término de base computacional.

Proposición 1. Base computacional. Los estados fundamentales de un cúbit, $|0\rangle$ y $|1\rangle$ constituyen una base ortonormal.

1.2. Esfera de Bloch

La esfera de Bloch es una herramienta importante en la teoría cuántica que simplifica la representación y comprensión de estados cuánticos de un cúbit.

1.2.1. Descripción de un estado en ángulos

Como hemos visto en la subsección 1.1.4, el estado de un cúbit (sin tener en cuenta la fase global) se puede escribir como sigue:

$$|\psi\rangle = a\,|0\rangle + be^{i\phi}\,|1\rangle$$

Donde a y b son números reales que además cumplen la siguiente condición:

$$a = |\alpha| \text{ y } b = |\beta|$$

Por lo tanto, si aplicamos la ecuación vista en la definición 2 que dice que:

$$|\alpha|^2 + |\beta|^2 = 1$$

Podemos deducir por tanto que $a^2 + b^2 = 1$.
De este modo, sabemos que el estado de un cúbit se puede escribir como:

$$|\psi\rangle = a\,|0\rangle + be^{i\phi}\,|1\rangle \text{ donde se cumple que } a^2 + b^2 = 1$$

Y podemos por tanto hacer una asociación natural con dos funciones trigonométricas que guardan la misma relación:

$$a = \cos\frac{\theta}{2} \text{ y } b = \sin\frac{\theta}{2}$$

Con lo que obtendríamos:

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

1.2.2. Representación del estado de un cúbit

Ahora que ya tenemos esta segunda forma de describir el estado de un cúbit, podemos dar una forma única y muy visual de represantarlo haciendo uso de las coordenadas esféricas, aunque hay que recordar que la fase global no se representa.

Definición 4. Esfera de Bloch. La esfera de Bloch es una representación geométrica de todos los posibles estados cuánticos de un cúbit. Matemáticamente, se basa en la proyección estereográfica de un espacio de Hilbert complejo bidimensional en una esfera unitaria en el espacio tridimensional. Cada punto en la esfera de Bloch corresponde a un estado cuántico único y se relaciona con un vector unitario en el espacio de Hilbert.

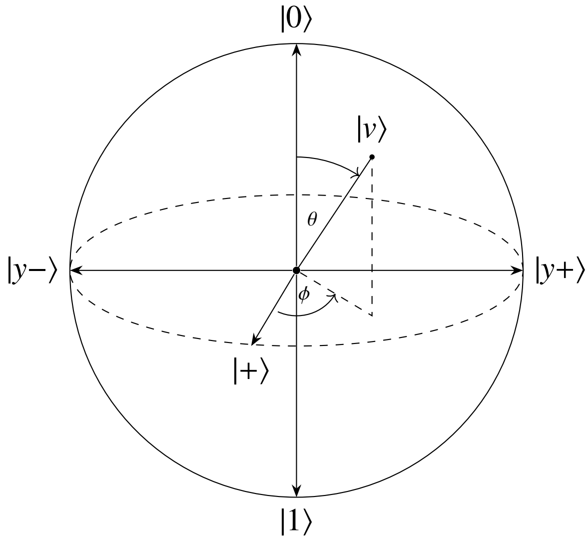


Figura 1.1: Esfera de Bloch

Posteriormente según vayamos avanzando en esta introducción nos irán apareciendo los estados que se sitúan en los ejes y que no son los estados básicos, es decir: $|+\rangle, |-\rangle, |y+\rangle, |y-\rangle$.

1.3. Operaciones con cúbits

Una vez hemos definido los conceptos más básicos, podemos empezar a construir sobre estos, el propósito de esta sección es comprender cómo se trabaja con los cúbits, es decir, la clase de operaciones que se pueden realizar sobre ellos y la manera en la que estas se llevan a cabo y perturban el estado de dichos cúbits.

1.3.1. Puertas lógicas cuánticas

Las conocidas como puertas lógicas presentes en la computación clásica, tienen su análogo en el mundo cuántico, pero antes de dar la definición de lo qué es formalmente una puerta lógica cuántica, es preciso recordar la definición de matriz unitaria:

Definición 5. Matriz unitaria. Una matriz unitaria es una matriz compleja U de tamaño $[n, n]$ que cumple que

$$UU^* = I_n$$

Siendo U^* la matriz traspuesta conjugada de U e I_n la matriz identidad de tamaño $[n, n]$. Por lo tanto, que una matriz sea unitaria implica que $U^{-1} = U^*$. Además, otras propiedades interesantes son que las matrices unitarias tienen norma 1 y preservan el producto escalar.

Definición 6. Puerta lógica. Una puerta lógica cuántica es una aplicación lineal unitaria que actúa sobre el espacio de Hilbert asociado a uno o varios cúbits. Matemáticamente, se representa como una matriz unitaria U que multiplica al vector de estado cuántico $|\psi\rangle$ para obtener un nuevo vector de estado $|\psi'\rangle$:

$$|\psi'\rangle = U\,|\psi\rangle$$

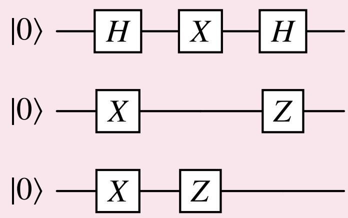
Donde $|\psi\rangle$ y $|\psi'\rangle$ son vectores de estado en el espacio de Hilbert, y U es una matriz unitaria. Esta operación lineal unitaria permite realizar transformaciones controladas y reversibles en la información cuántica representada por el vector de estado.

1.3.2. Circuitos cuánticos

Hasta ahora, hemos estado hablando de un único cúbit pero lo normal es trabajar con más y emplear la representación de circuitos cuánticos para entender mejor las operaciones que se realizan a estos, su orden y dependencia.

Definición 7. Circuito cuántico. Un circuito cuántico, desde un punto de vista matemático, es una secuencia de operadores unitarios representados por matrices unitarias que actúan sobre n cúbits, los cuales son vectores en un espacio de Hilbert complejo de dimensión $2n$. Estas matrices unitarias representan puertas cuánticas que aplican transformaciones lineales en los estados de los cúbits. La secuencia de operadores se combina mediante multiplicación de matrices para calcular el estado final del sistema cuántico, que describe la evolución de los cúbits a medida que atraviesan el circuito.

Ejemplo 2. Un circuito cuántico tiene el siguiente aspecto:



Notación 3. Notación de un circuito En un circuito los cúbits se representan como "cables" que se dibujan en paralelo. Dichos cables pueden empezar describiendo el estado inicial del cúbit ó en caso de no hacerlo se entiende que empiezan en $|0\rangle$. La forma de nombrar los cúbits es de arriba a abajo empezando por 0, es decir, si pintamos n cables tenemos los cúbits de 0 a $n-1$.
Además, también se representan las operaciones que se llevan a cabo en los mismos a través de "cajas" que contienen el nombre por el que se describe la puerta. Por último, cabe observar que las operaciones que se llevan a cabo sobre distintos cúbits pueden escribirse de forma paralela (ó no) sin que esto afecte a las operaciones. Por ejemplo en el ejemplo anterior 2, a los cúbits 1 y 2 les estamos aplicando las mismas puertas y sin embargo en el cúbit 1 no están escritas de forma paralela y en el 2 sí pero ambos cúbits tienen el mismo estado final.

1.4. Principales puertas lógicas cuánticas que actúan sobre 1 cúbit

Una vez hemos visto qué es una puerta lógia cuántica y un circuito, vamos a ver cúales son las principales puertas lógicas que nos podemos encontrar, su matriz unitaria y su representación en un circuito.

1.4.1. Pauli$X$

La primera puerta que vamos a tratar es la puerta de Pauli$X$, también conocida como la puerta NOT debido a que es muy similar a la operación lógia de negación.

Definición 8. Puerta lógica cuántica Pauli$X$. La puerta lógica cuántica Pauli$X$ se aplica a un único cúbit y se representa mediante la siguiente matriz:

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

Proposición 2. Los autovectores normalizados de la puerta Pauli$X$ son

$$|v_1\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$|v_2\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Sus autovalores respectivamente son $\lambda_1 = -1$ y $\lambda_2 = 1$.
Demostración. Calculamos el polinomio característico haciendo el determinante de

$$\begin{pmatrix}0 & 1\\1 & 0\end{pmatrix} - xI$$

y obtenemos $p(x) = x^2 - 1$. Ahora igualamos a 0 y despejamos para obtener los autovalores: $p(x) = 0 \longrightarrow x = -1, x = 1$.
Una vez los tenemos, calculamos el núcleo de las matrices asociadas a estos, y con ello los autovectores:

En el caso de $\lambda_1 = -1$:

$$\begin{pmatrix}-1 & 1\\1 & -1\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix} \longrightarrow x = y \longrightarrow v_1 = \begin{pmatrix}1\\1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$$

En el caso de $\lambda_2 = 1$:

$$\begin{pmatrix}1 & 1\\1 & 1\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix} \longrightarrow x = -y \longrightarrow v_2 = \begin{pmatrix}1\\-1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$$

Con lo que queda demostrado.

Notación 4. Los vectores $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ y $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ son muy usados en el mundo de la computación cuántica y tienen una notación especial:

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Como podemos comprobar en la figura 1.1 los estados anteriores $|+\rangle$ y $|-\rangle$ aparecen situados en el eje X y en la sección 1.4.4 entenderemos por qué.

Notación 5. Puerta de Pauli$X$. La puerta de Pauli$X$ se representa en un circuito mediante los símbolos:

$$-\boxed{X}-$$

$$-\oplus-$$

1.4.2. Hadamard

La puerta lógica cuántica de Hadamard es esencial en la computación cuántica debido a su capacidad para crear superposiciones, esta propiedad es fundamental en la resolución de problemas cuánticos y permite que los algoritmos cuánticos exploren soluciones de manera paralela, lo que a menudo conduce a una aceleración significativa en comparación con los algoritmos clásicos.

Definición 9. Puerta lógica cuántica de Hadamard. La puerta lógica cuántica Hadamard se aplica a un único cúbit y se representa mediante la siguiente matriz:

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

Como hemos dicho anteriormente, esta puerta es muy importante por crear estados de superposición, en concreto, crea un estado de superposición uniforme (con amplitudes de probabilidad iguales) cuando se aplica sobre los estados $|0\rangle$ y $|1\rangle$:

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

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Proposición 3. Base de Hadamard. Los vectores $|+\rangle$ y $|-\rangle$ constituyen una base ortogonal de los estados de un cúbit .

Proposición 4. $H^{-1} = H$. La matriz inversa de H es ella misma.
Demostración. Facilmente se puede comprobar que $H \cdot H = I$ y por lo tanto,
$H = H^{-1} = H^*$ (donde la última igualdad se debe a que H es una matriz unitaria).

Notación 6. Puerta de Hadamard. La puerta de Hadamard se representa en un circuito mediante el símbolo:

$$-\boxed{H}-$$

1.4.3. Rotaciones Z

La puerta RZ ó rotaciones de Z son otro tipo de rotaciones alrededor del eje Z.

Definición 10. Rotaciones Z. Sea $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ un estado genérico, entonces una rotación Z de ángulo ω expresado en radianes se define como:

$$RZ(\omega) = \alpha |0\rangle + \beta e^{i\omega} |1\rangle$$

La matriz que representa esta rotación es:

$$\begin{pmatrix} e^{-i\frac{\omega}{2}} & 0 \\ 0 & e^{i\frac{\omega}{2}} \end{pmatrix}$$

Debido a que la matriz anterior es diagonal, podemos observar sin esfuerzo que los autovectores normalizados de las Rotaciones Z son: $|0\rangle$ y $|1\rangle$ y sus autovalores son $e^{-i\frac{\omega}{2}}$ y $e^{i\frac{\omega}{2}}$ respectivamente y esto se puede apreciar también en la figura 1.1 ya que aparecen los estados $|0\rangle$ y $|1\rangle$ situados en el eje Z

Dado que la puerta RZ es una rotación, es conveniente observar que la inversa de una rotación de ángulo ω es la misma rotación de ángulo $-\omega$, en particular, se tiene que:

$$RZ^{-1}(\omega) = RZ(-\omega) \text{ y por ser una matriz unitaria: } RZ^{*}(\omega) = RZ(-\omega)$$

Hay 3 casos particulares de rotaciones Z que destacan por encima del resto y que se corresponden con las siguientes puertas lógicas:

- **Puerta Pauli$Z$:** Este caso se corresponde con $RZ(\pi)$
- **Puerta S:** Este caso se corresponde con $RZ(\frac{\pi}{2})$
- **Puerta T:** Este caso se corresponde con $RZ(\frac{\pi}{4})$

Notación 7. Los símbolos correspondientes a las puertas anteriores son:

Rotación Z

$$-\boxed{R_Z(\omega)}-$$

Puerta de Pauli$Z$

$$-\boxed{Z}-$$

Puerta S

$$-\boxed{S}-$$

Puerta T

$$-\boxed{T}-$$

1.4.4. Rotaciones X

La puerta RX ó rotaciones de X son otro tipo de rotaciones alrededor del eje X.

Definición 11. Rotaciones X. Una rotación X de ángulo θ expresado en radianes se define de forma matricial como:

$$\begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$$

La puerta Pauli$X$ es también una rotación X, concretamente se tiene que: Pauli$X$ = $-iRX(\pi)$.
En el caso de esta rotación, sus autovectores son $|+\rangle$ y $|-\rangle$ como podemos observar en la figura 1.1.

Notación 8. La puerta $RX(\theta)$ se representa en un circuito de la siguiente manera:

$$-\boxed{R_Z(\theta)}-$$

1.4.5. Rotaciones Y

La puerta RY ó rotaciones de Y son otro tipo de rotaciones alrededor del eje Y.

Definición 12. Rotaciones Y. Una rotación Y de ángulo θ expresado en radianes se define de forma matricial como:

$$\begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$$

En el caso de esta rotación, sus autovectores son:

$$|y+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$$

$$|y-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$$

como podemos observar en la figura 1.1.

Dentro de este de conjunto de rotaciones destaca el caso particular de $\theta = \pi$, esta puerta es conocida como Pauli$Y$.

Notación 9. Los símbolos correspondiestes a las puertas anteriores son los siguientes:

Rotación Y

$$-\boxed{R_Y(\omega)}-$$

Puerta de Pauli$Y$

$$-\boxed{Y}-$$

1.5. Mediciones y observables

En esta seción se explora con algo más de detalle el concepto, muy importante, de realizar mediciones en un sistema cuántico.

1.5.1. Mediciones

En la sección anterior se ha mencionado el proceso de medir un cúbit en el contexto de la base computacional (medir 0 o medir 1), este es un caso muy particular de las mediciones en general.

Definición 13. Medición. Una medición cuántica viene dada por un conjunto de operadores, M_m tal que cada operador representa un posible valor medible con probabilidad

$$p(m) = \langle\psi| M_m^* M_m |\psi\rangle$$

donde $|\psi\rangle$ es el estado sobre el que se aplica la medición.

La medición cumple la ecuación de completitud:

$$I = \sum_m M_m^* M_m$$

Esta ecuación implica que la suma de las probabilidades sea 1.

$$1 = \sum_m p(m) = \sum_m \langle\psi| M_m^* M_m |\psi\rangle$$

Ejemplo 3. Medir en la base computacional se corresponde a la medición con operadores

$$M_0 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = |0\rangle\langle 0| \quad M_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = |1\rangle\langle 1|$$

Sea $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ entonces

$$p(0) = \begin{pmatrix} \alpha^* & \beta^* \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

1.5.2. Mediciones proyectivas

En esta sección mostraremos un caso particular de medición, pero antes es necesario de dar una definición previa. La descomposición espectral es un teorema de representación de operadores normales que es de gran utilidad en el mundo de la cuántica.

Teorema 1. Descomposición espectral. Sea M un operador en un espacio vectorial V, M es un operador normal (aquél que conmuta con su adjunto $MM^* = M^*M$, como por ejemplo un operador unitario ó hermítico $M = M^*$) si y sólo si es diagonalizable.

Definición 14. Mediciones proyectivas. Las mediciones proyectivas son descritas por un observable M, que es un operador hermítico definido en el espacio del sistema observado. Su descomposición espectral es la siguiente:

$$M = \sum_m m P_m$$

Siendo P_m la proyección en el autoespacio de M asociado al autovalor m. Por lo tanto, al medir un estado $|\psi\rangle$, la probabilidad de obtener m viene dada por:

$$p(m) = \langle\psi| P_m |\psi\rangle$$

En caso de obtener dicho valor m, el estado del sistema cuántico tras la medición sería:

$$\frac{P_m|\psi\rangle}{\sqrt{p(m)}} = \frac{P_m|\psi\rangle}{||P_m|\psi\rangle||}$$

Como hemos mencionado anteriormente, las mediciones proyectivas son un caso particular de mediciones que toma como conjunto de operadores de medida el conjunto de las proyecciones $\{P_m\}$, y que tiene como posibles resultados de la medición los correspondientes autovalores m del observable.

Ejemplo 4. Como ejemplo de medición, vamos a comenzar con la descomposición espectral de un operador sencillo:

$$M = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = 1 \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} + 1 \begin{pmatrix} \frac{-1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{-1}{2} \end{pmatrix} = m_1 \cdot P_1 + m_2 \cdot P_2$$

La descomposición espectral nos indica unos operadores de medida que cumplen la condición de completitud. en particular son proyecciones en los autoespacios del operador que observamos son los espacios generados por $|+\rangle$ y $|-\rangle$.

Realizamos una medición para $|\psi\rangle = |0\rangle$ entonces la probabilidad de medir $|+\rangle$ viene dada por

$$\langle\psi| P_1 |\psi\rangle = \frac{1}{2}$$

e igualmente para $|-\rangle$.

Si midiesemos $|+\rangle$ el estado después de la medición sería

$$\frac{P_1 |\psi\rangle}{||P_1 |\psi\rangle||} = \frac{(\frac{1}{2}, \frac{1}{2})}{\sqrt{\frac{1}{2}}} = |+\rangle$$

como es de esperar.

En vez de partir del operador M, podemos decir directamente que medimos sobre una base, en este caso la base $\{|+\rangle, |-\rangle\}$.

1.6. Sistemas multicúbit

Como hemos mencionado al comienzo, concretamente en la definición 1, los cúbits se definen como vectores dentro de un espacio de Hilbert complejo de dimensión 2, y para abordar los sistemas multicúbit es necesario comenzar explicando cómo se componen estos espacios de Hilbert en los que se encuentran.

1.6.1. Producto tensorial

En general, el producto tensorial es una operación entre dos espacios vectoriales que nos permite obtener un espacio de dimensión más grande. En concreto, si V, W son espacios vectoriales tales que $\dim(V) = n$, $\dim(W) = m$, entonces $\dim(V \otimes W) = nm$.

Usamos la referencia [1] para la siguiente definición.

Definición 15. Producto tensorial de espacios vectoriales.

Sean V, W espacios vectoriales sobre un mismo cuerpo de escalares. Definimos el producto tensorial $V \otimes W$ como el espacio cuyos elementos son combinaciones lineales de pares de elementos de los espacios originales $v \otimes w$ ($v \in V, w \in W$), y las operaciones de suma y multiplicación por un escalar se derivan a partir de las operaciones en V y W mediante las siguientes propiedades:

- Sean z un escalar, $v \in V, w \in W$ entonces

$$z(v \otimes w) = (zv) \otimes w = v \otimes (zw)$$

- Sean $v_1, v_2 \in V$, $w \in W$ entonces

$$(v_1 + v_2) \otimes w = v_1 \otimes w + v_2 \otimes w$$

- Sean $v \in V$, $w_1, w_2 \in W$ entonces

$$v \otimes (w_1 + w_2) = v \otimes w_1 + v \otimes w_2$$

Por otra parte podemos definir operadores lineales en $V \otimes W$ a partir de operadores en los espacios V, W.

Sean $A : V \mapsto V$ y $B : W \mapsto W$, lineales, definimos el operador $A \otimes B$ de la siguiente manera:

$$(A \otimes B)(v \otimes w) = Av \otimes Bw$$

Esta ecuación se extiende mediante linearidad a todos los elementos de $V \otimes W$.

Esto nos resultará útil para representar operaciones en sistemas multicúbits dónde es común aplicar una puerta a ciertos cúbits y otra a otros, permitiéndonos descomponerlas.

También podemos derivar un producto interno para $V \otimes W$ a partir de los productos internos de V y W si los tuvieran. El producto interno entre dos elementos arbitrarios de $V \otimes W$ se define de la siguiente manera.

$$\left(\sum_i a_i(v_i \otimes w_i)\right) \cdot \left(\sum_j b_j(v'_j \otimes w'_j)\right) = \sum_{i,j} a_i^* b_j (v_i \cdot v'_j)(w_i \cdot w'_j)$$

Se puede comprobar que esto cumple las propiedades de un producto interno y que el producto tensorial de dos espacios de Hilbert es de Hilbert.

Otra característica importante del producto tensorial es que podemos construir una base a partir de bases de los espacios originales. Si $\{e_i\}_i$ es una base (ortogonal) de V y $\{d_j\}_j$ es una base (ortogonal) de W, entonces $\{e_i \otimes d_j\}_{i,j}$ es una base (ortogonal) de $V \otimes W$.

Se puede ver el producto tensorial de manera más concreta mediante la siguiente representación matricial.

$$\begin{pmatrix} a & \dots & b \\ \vdots & \ddots & \vdots \\ c & \dots & d \end{pmatrix} \otimes \begin{pmatrix} \alpha & \dots & \beta \\ \vdots & \ddots & \vdots \\ \gamma & \dots & \delta \end{pmatrix} = \begin{pmatrix} a\begin{pmatrix} \alpha & \dots & \beta \\ \vdots & \ddots & \vdots \\ \gamma & \dots & \delta \end{pmatrix} & \dots & b\begin{pmatrix} \alpha & \dots & \beta \\ \vdots & \ddots & \vdots \\ \gamma & \dots & \delta \end{pmatrix} \\ \vdots & \ddots & \vdots \\ c\begin{pmatrix} \alpha & \dots & \beta \\ \vdots & \ddots & \vdots \\ \gamma & \dots & \delta \end{pmatrix} & \dots & d\begin{pmatrix} \alpha & \dots & \beta \\ \vdots & \ddots & \vdots \\ \gamma & \dots & \delta \end{pmatrix} \end{pmatrix}$$

Ejemplo 5.

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

Notación 10. Para simplificar la forma de escribir dicho producto tensorial, se emplea la siguiente notación:

$$|0\rangle \otimes |0\rangle \otimes \ldots \otimes |1\rangle = |00\ldots 1\rangle$$

En el caso concreto de la cuántica se aplica el producto tensorial entre los estados de los sistemas cuánticos y sobre los distintos operadores unitarios.

Un sistema multicúbit de n cúbits va a ser el producto tensorial n veces del espacio de 1 cúbit que hemos definido.

Por lo tanto un estado de un sistema multicúbit se representará como un vector en un espacio de Hilbert de dimensión 2^n.

Proposición 5. Descomposición espectral del producto tensorial. Sean A y B operadores normales con las siguientes descoposiciones espectrales respectivamente:

$$A = \lambda_1 P_1 + \lambda_2 P_2 + \cdots + \lambda_m P_m$$

$$B = \eta_1 Q_1 + \eta_2 Q_2 + \cdots + \eta_n Q_n$$

Entonces la descomposición espectral de $A \otimes B$ es la siguiente:

$$A \otimes B = \sum_{i=1}^{m} \sum_{j=1}^{n} \lambda_i \eta_j P_i \otimes Q_j$$

Siendo $\lambda_i \eta_j$, $i = 1...m$, $j = 1...n$ los autovalores de $A \otimes B$

1.6.2. Base computacional de un sistema de n cúbits

Según lo expuesto de manera general en la sección sobre el producto tensorial 1.6.1, se puede definir una base computacional para un sistema de n cúbits a partir de la base para el espacio de 1 cúbit. Como ya me hemos explicado representaremos los estados de un sistema de n cúbits como vectores de un espacio de Hilbert de dimensión 2^n sobre los complejos y por lo tanto dichas bases constarán de 2^n elementos.

Ejemplo 6. La base computacional de un sistema de 2 cúbits es:

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

$$|0\rangle \otimes |1\rangle = |01\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$|1\rangle \otimes |0\rangle = |10\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$|1\rangle \otimes |1\rangle = |11\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Como podemos observar en el ejemplo anterior, si tenemos la base computacional de un sistema de n cúbits, cada elemento de la base va a tener un 1 en una posición del vector y 0 en el resto. La posición en la que va estar dicho 1 se puede obtener fácilmente sin necesidad de hacer el producto, por ejemplo, si tomamos el elemento $|10\rangle$ del ejemplo anterior, lo interpretamos cómo si fuera binario y lo pasamos a decimal, obtendríamos el número 2 de modo que el vector tendrá el 1 en la posición 2 (comenzando a indexar dicho vector en 0).

1.6.3. Estados entrelazados y separables

Una vez hemos visto el producto tensorial, es necesario introducir un par de conceptos relacionados con el mismo.

Definición 16. Estados entrelazados y separables. Se dice que el estado de un sistema multicúbit es entrelazado si no se puede descomponer como producto tensorial de los estados individuales de los cúbits que constituyen el sistema, en caso contrario se dice que el estado es separable.

Ejemplo 7. Supongamos que tenemos un sistema de dos cúbits cuyo estado es:

$$|\psi\rangle = \frac{1}{\sqrt{2}} |00\rangle + |11\rangle$$

y los estados de los cúbits que constituyen el sistema son de la forma:

$$|\psi_1\rangle = \alpha |0\rangle + \beta |1\rangle \quad |\psi_2\rangle = \gamma |0\rangle + \delta |1\rangle$$

Al hacer el producto tensorial de $|\psi_1\rangle \otimes |\psi_2\rangle$ tendriamos que:

$$|\psi_1\rangle \otimes |\psi_2\rangle = \alpha\gamma |00\rangle + \alpha\delta |01\rangle + \beta\gamma |10\rangle + \beta\delta |11\rangle$$

Y para que el resultado de dicho producto fuese el estado $|\psi\rangle$ tendríamos que:

$$\alpha\gamma = \beta\delta = \frac{1}{\sqrt{2}} \qquad \alpha\delta = \beta\gamma = 0$$

Y como podemos ver esto no es posible por lo que el estado es entrelazado.

En el ejemplo anterior hemos visto un estado entrelazado, concretamente un estado totalmente entrelazado pero también existen otro tipo de entrelazamientos en los que se puede hacer una descomposición parcial en productos tensoriales. Por ejemplo, en el caso de tener un sistema de 3 cúbits y poder dividirlo en un producto de 2 estados (en lugar de 3) tendríamos un entrelazamiento parcial.

1.6.4. Teorema de no clonación.

Es posible que trás lo visto en este capítulo surja preguntarse si existe la posibilidad de crear una copia idéntica de un estado cuántico desconocido arbitrario, sorprendentemente la respuesta es que no, expliquemos por qué.

Teorema 2. Teorema de no clonación. No existe un operador unitario U tal que dado un estado cuántico arbitrario $|\psi\rangle$,

$$U(|\psi\rangle \otimes |\phi\rangle) = |\psi\rangle \otimes |\psi\rangle$$

siendo $|\phi\rangle$ un estado inicial cualquiera del cúbit.
Demostración. Supongamos que existe dicho operador unitario tal que

$$U((|\psi_1\rangle + |\psi_2\rangle) \otimes |\psi\rangle) = (|\psi_1\rangle + |\psi_2\rangle) \otimes (|\psi_1\rangle + |\psi_2\rangle)$$

Dado que los operadores unitarios son lineares, aplicando esta propiedad al lado izquierdo de la igualdad tendríamos que

$$U((|\psi_1\rangle + |\psi_2\rangle) \otimes |\psi\rangle) = U(|\psi_1\rangle \otimes |\psi\rangle) + U(|\psi_2\rangle \otimes |\psi\rangle) = |\psi_1\rangle \otimes |\psi_1\rangle + |\psi_2\rangle \otimes |\psi_2\rangle$$

Por lo tanto se tendría que

$$(|\psi_1\rangle + |\psi_2\rangle) \otimes (|\psi_1\rangle + |\psi_2\rangle) = (|\psi_1\rangle \otimes |\psi_1\rangle) + (|\psi_2\rangle \otimes |\psi_2\rangle)$$

Lo que es absurdo.

La importancia del teorema de no clonación, en el contexto de los códigos correctores, es que descarta la posibilidad de copiar información, anædiendo directamete redundancia, para corregir errores.

1.7. Principales puertas lógicas cuánticas que actúan sobre sistemas de más de 1 cúbit

Una vez que ya hemos introducido los sistemas multicúbits, vamos a ver cúales son las principales puertas lógicas que nos podemos encontrar, su matriz unitaria y su representación en un circuito.

1.7.1. Puerta $CNOT$

La puerta cuántica $CNOT$ (Controlled-NOT) es una operación fundamental en la computación cuántica que realiza una inversión condicional en un cúbit (denominado objetivo) basado en el estado de otro cúbit denominado (de control), y desempeña un papel esencial debido a su propiedad de crear estados entrelazados.

Definición 17. Puerta $CNOT$. La puerta cuántica $CNOT$ se aplica a 2 cúbits, es una manera controlada de aplicar la puerta NOT ó Pauli$X$ a un cúbit que denominaremos "objetivo" en base al estado de otro cúbit que denominaremos "de control", concretamente realiza una operación XOR: si tenemos un estado $|ab\rangle$, entonces suponiendo que el primer bit es el bit de control, $CNOT_{ab}\,|ab\rangle = |a(b \oplus a)\rangle$, en caso de que sea el segundo se tendría que $CNOT_{ba}\,|ab\rangle = |(a \oplus b)b\rangle$. La matriz que representa esta puerta para el caso en el que el primer bit sea el de control es:

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

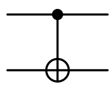
Y en caso de que el bit de control sea el segundo:

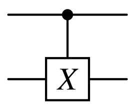
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Ejemplo 8. Puerta $CNOT_{ba}$ aplicada a la base computacional

$\|ab\rangle$	$CNOT_{ba}\,\|ab\rangle$
$\|00\rangle$	$\|00\rangle$
$\|01\rangle$	$\|11\rangle$
$\|10\rangle$	$\|10\rangle$
$\|11\rangle$	$\|01\rangle$

Notación 11. Puerta $CNOT$. La puerta $CNOT$ se representa en un circuito de las siguientes maneras



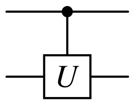


Donde el punto relleno se sitúa sobre el cúbit de control y la puerta NOT sobre el objetivo.

1.7.2. Puertas controladas

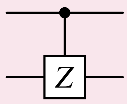
La puerta $CNOT$ 1.7.1 es la puerta controlada más importante pero a parte de esta, cualquier puerta vista en la sección 1.4 puede convertirse en una puerta controlada, esto se hace de la misma forma que para la puerta $CNOT$, se define un cúbit de control y otro objetivo al que aplicar la puerta de modo que cuando el de control sea 1 se aplica la puerta y en caso contrario no se aplica.

Notación 12. Notación general para puertas controladas. Sea U una matriz unitaria que define una operación sobre un cúbit, si queremos controlar dicha puerta se representaría de la siguiente manera:



Donde el punto relleno se sitúa sobre el cúbit de control y la operación U sobre el objetivo.

Ejemplo 9. Puerta controlada Z



1.7.3. Puerta SWAP

La puerta cuántica SWAP como su propio nombre indica es una operación cuántica que permite intercambiar los estados de dos cúbits. A través de esta operación, los valores cuánticos de dos cúbits se permutan, lo que resulta crucial en la manipulación de información y en la reorganización de datos en circuitos cuánticos, lo que la convierte en una puerta importante.

Definición 18. Puerta SWAP. La puerta SWAP intercambia los estados de dos cúbits y se puede definir mediante e producto tensorial de la siguiente manera:

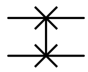
$$SWAP(|\psi\rangle \otimes |\phi\rangle) = |\phi\rangle \otimes |\psi\rangle$$

La podemos describir también mediante la siguiente matriz:

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

Claramente podemos observar en la matriz anterior como los estados $|00\rangle$ y $|11\rangle$ no cambian mientras que $|01\rangle$ y $|10\rangle$ se invierten.

Notación 13. Puerta SWAP. La puerta SWAP se representa en un circuito como:



1.7.4. Puerta Toffoli

La puerta cuántica Toffoli, también conocida como la puerta CCNOT (Controlled-Controlled-NOT), es una operación clave en la computación cuántica que extiende la funcionalidad de la puerta CNOT. La puerta Toffoli realiza una operación condicional en tres cúbits, donde dos cúbits de control determinan si se invierte el estado de un tercer cúbit, el objetivo.

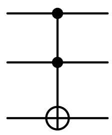
Definición 19. Puerta Toffoli. La puerta Toffoli al igual que la puerta $CNOT$ puede representarse mediante la operación XOR de la siguiente manera en el caso en el que los dos primeros bits sean los bits de control: controlled-$CNOT_{abc} \left|abc\right\rangle = \left|ab(c \oplus ab\right\rangle$ de modo que se suma el producto de los bits de control al bit objetivo y se haría de manera análoga en el caso de cambiar los bits de control. La matriz asociada a dicha operación (con primer y segundo bit de control) es la siguiente:

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

Ejemplo 10. Puerta controlled-$CNOT_{abc}$ aplicada a la base computacional

$\|abc\rangle$	$CCNOT_{abc}\|abc\rangle$
$\|000\rangle$	$\|000\rangle$
$\|001\rangle$	$\|001\rangle$
$\|010\rangle$	$\|010\rangle$
$\|011\rangle$	$\|011\rangle$
$\|100\rangle$	$\|100\rangle$
$\|101\rangle$	$\|101\rangle$
$\|110\rangle$	$\|111\rangle$
$\|111\rangle$	$\|110\rangle$

Notación 14. Puerta Toffoli. La puerta Toffoli se representa en un circiuto como:



Para poder completar nuestra introducción autocontenida a la computación cuántica se han introducido las principales puertas lógicas, la mayoría de las cuales serán de gran importancia en los siguientes capítulos.

2. MATRIZ DE DENSIDAD Y SUPEROPERADORES

En este capítulo se utiliza principalmente como referencia *Quantum computation and quantum information* [1]

2.1. Matriz de densidad

En este capítulo vamos a presentar un enfoque distinto para la computación cuántica, en contraposición con los vectores que hemos utilizado para representar el estado de un sistema cuántico: matrices de densidad.

Como formulación matemática de la computación cuántica va a resultar ser igual de expresiva que los vectores que representan un estado, pero va a ser más útil a la hora de abordar ciertos problemas.

Esta perspectiva nos va a ser especialmente útil en el contexto de los códigos de corrección de errores cuánticos para representar que un sistema se encuentra en varios posibles estados con cierta probabilidad.

Definición 20. Conjunto de estados cuánticos puros. Llamamos Conjunto de estados cuánticos puros a un conjunto de pares

$$\{(|\psi_i\rangle, p_i) : i \in 1 \dots n\} \quad \text{con } \sum_i p_i = 1$$

De manera que el sistema se encuentra en el estado $|\psi_i\rangle$ con probabilidad p_i.

Observemos la diferencia entre lo expresado por la superposición y los conjuntos de estados cuánticos puros. El concepto clave de esto es que un estado en superposición no deja de ser un único estado conocido, aunque hablemos de amplitudes de probabilidad en el contexto de la superposición esas probabilidades solo se manifiestan a la hora de realizar mediciones. Un conjunto de estados cuánticos puros representan varios posibles estados para un sistema, cada uno de ellos puede ser a su vez una superposición.

Ejemplo 11. Un conjunto de estados es $\left\{\left(|0\rangle, \frac{1}{2}\right), \left(|1\rangle, \frac{1}{2}\right)\right\}$, que es distinto a un estado de superposición con amplitudes de probabilidad $\frac{1}{2}$ como $|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$, ya que como decíamos este se trata de un único estado definido. A efectos de la medición en la base computacional serían indistinguibles pero consideremos ahora que medimos en la base $\{|+\rangle, |-\rangle\}$. En el primer caso ambos estados son equiprobables mientras que en el segundo caso $|+\rangle$ se mide con probabilidad 1.

Definición 21. Matriz de densidad. La Matriz de densidad (u *Operador de densidad*) para un conjunto de estados cuánticos puros $\{(|\psi_i\rangle, p_i) : i \in 1 \ldots n\}$ es

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$$

Ejemplo 12. Siguiendo con el ejemplo anterior, el conjunto con los dos estados equiprobables tendrá matriz de densidad

$$\frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1| = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

mientras que por otra parte el estado puro tendrá matriz de densidad

$$|\psi\rangle\langle\psi| = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

(Observamos de nuevo que son distintas)

2.2. Propiedades de la matriz de densidad

Veamos como se expresan las transformaciones en terminos de la matriz de densidad.

Proposición 6. La transformación del sistema descrito por una matriz de densidad ρ, mediante un operador unitario U va a ser dada por

$$U\rho U^*$$

Es directo a partir de la definición si consideramos que la matriz de densidad es la de un conjunto de estados $\{(|\psi_i\rangle, p_i)\}$, tras la aplicación de U tendremos el conjunto de estados

$\{(U|\psi_i\rangle, p_i)\}$, y su matriz de densidad puede ser computada por

$$\sum_i p_i U|\psi_i\rangle\langle\psi_i|U^* = U\rho U^*$$

Las combinaciones de matrices de densidad se comportan según lo esperable:

Proposición 7. Si tenemos un conjunto de conjuntos de estados dados por sus matrices de densidad ρ_i cada uno de ellos con probabilidad p_i respectivamente tenemos que la matriz de densidad del estado es

$$\rho = \sum_i p_i \rho_i$$

Proposición 8. El estado conjunto de dos estados con matrices de densidad ρ, ρ' tiene matriz de densidad

$$\rho \otimes \rho'$$

Ambas proposiciones son de nuevo consecuencia directa de la definición, de la relación entre expresar el estado como vector y como matriz de densidad.

Para el siguiente teorema es conveniente hacer uso de los vectores $|\tilde{\psi}_i\rangle$, los cuales pueden no estar normalizados a longitud unitaria. Decimos que el conjunto $\{|\tilde{\psi}_i\rangle\}$ genera el operador $\rho \equiv \sum_i |\tilde{\psi}_i\rangle\langle\tilde{\psi}_i|$, y así la conexión con la representación usual de los operadores de densidad en un conjunto queda expresada por la ecuación $\tilde{\psi}_i = \sqrt{p_i}|\psi_i\rangle$. ¿Cuándo dos conjuntos de vectores, $\{|\tilde{\psi}_i\rangle\}$ y $\{|\tilde{\varphi}_j\rangle\}$, generan el mismo operador ρ? La solución a este problema nos permitirá responder la pregunta de qué conjuntos dan lugar a una matriz de densidad dada.

Teorema 3. Libertad unitaria en el conjunto de matrices densidad. Los conjuntos $\{|\tilde{\psi}_i\rangle\}$ y $\{|\tilde{\varphi}_j\rangle\}$ generan la misma matriz densidad si y solo si

$$|\tilde{\psi}_i\rangle = \sum_j u_{ij}|\tilde{\varphi}_j\rangle, \tag{2.1}$$

donde u_{ij} es una matriz unitaria de números complejos, con índices i y j, y rellenamos el conjunto de vectores $|\tilde{\psi}_i\rangle$ o $|\tilde{\varphi}_j\rangle$, el que sea menor, con vectores adicionales 0 para que ambos conjuntos tengan el mismo número de elementos.

Como consecuencia del teorema, observamos que $\rho = \sum_i p_i|\psi_i\rangle\langle\psi_i| = \sum_j q_j|\varphi_j\rangle\langle\varphi_j|$ para estados normalizados $|\psi_i\rangle, |\varphi_j\rangle$ y distribuciones de probabilidad p_i y q_j si y solo si

$$\sqrt{p_i}|\psi_i\rangle = \sum_j u_{ij}\sqrt{q_j}|\varphi_j\rangle, \tag{2.2}$$

para alguna matriz unitaria u_{ij}, y podemos rellenar el conjunto más pequeño con elementos cuya probabilidad sea cero para hacer que ambos conjuntos tengan el mismo tamaño. Así, este teorema caracteriza la libertad en los conjuntos $\{p_i, |\psi_i\rangle\}$ que generan una dada matriz densidad ρ. De hecho, el caso de las matrices de densidad con dos descomposiciones diferentes, surge como un caso especial de este resultado general.

Demostración:

Supongamos que $|\tilde{\psi}_i\rangle = \sum_j u_{ij}|\tilde{\varphi}_j\rangle$ para alguna matriz unitaria u_{ij}. Entonces,

$$\sum_i |\tilde{\psi}_i\rangle\langle\tilde{\psi}_i| = \sum_{ijk} u_{ij}u_{ik}^*|\tilde{\varphi}_j\rangle\langle\tilde{\varphi}_k|, \tag{2.3}$$

$$= \sum_{jk}(\sum_i u_{ij}u_{ik}^*)|\tilde{\varphi}_j\rangle\langle\tilde{\varphi}_k|, \tag{2.4}$$

$$= \sum_{jk} \delta_{kj}|\tilde{\varphi}_j\rangle\langle\tilde{\varphi}_k|, \tag{2.5}$$

$$= \sum_j |\tilde{\varphi}_j\rangle\langle\tilde{\varphi}_j|, \tag{2.6}$$

lo que muestra que los conjuntos $|\tilde{\psi}_i\rangle$ y $|\tilde{\varphi}_j\rangle$ generan el mismo operador.

Recíprocamente, supongamos que

$$A = \sum_i |\tilde{\psi}_i\rangle\langle\tilde{\psi}_i| = \sum_j |\tilde{\varphi}_j\rangle\langle\tilde{\varphi}_j|. \tag{2.7}$$

Sea $A = \sum_k \lambda_k|k\rangle\langle k|$ una descomposición de A tal que los estados $|k\rangle$ son ortonormales y los λ_k son estrictamente positivos. Nuestra estrategia es relacionar los estados $|\tilde{\psi}_i\rangle$ con los estados $|\tilde{k}\rangle \equiv \sqrt{\lambda_k}|k\rangle$, y de manera similar relacionar los estados $|\tilde{\varphi}_j\rangle$ con los estados $|\tilde{k}\rangle$. Combinando ambas relaciones obtendremos el resultado. Sea $|\psi\rangle$ cualquier vector ortonormal al espacio generado por los $|\tilde{k}\rangle$, de modo que $\langle\psi|\tilde{k}\rangle\langle\tilde{k}|\psi\rangle = 0$ para todo k, y así observamos que

$$0 = \langle\psi|A|\psi\rangle = \sum_i \langle\psi|\tilde{\psi}_i\rangle\langle\tilde{\psi}_i|\psi\rangle = \sum_i |\langle\psi|\tilde{\psi}_i\rangle|^2. \tag{2.8}$$

Así, $\langle\psi|\tilde{\psi}_i\rangle = 0$ para todo i y para todo $|\psi\rangle$ ortonormal al espacio generado por los $|\tilde{k}\rangle$. Se deduce que cada $|\tilde{\psi}_i\rangle$ puede expresarse como una combinación lineal de los $|\tilde{k}\rangle$, $|\tilde{\psi}_i\rangle = \sum_k c_{ik}|\tilde{k}\rangle$. Como $A = \sum_k |\tilde{k}\rangle\langle\tilde{k}| = \sum_i |\tilde{\psi}_i\rangle\langle\tilde{\psi}_i|$, observamos que

$$\sum_k |\tilde{k}\rangle\langle\tilde{k}| = \sum_{kl}(\sum_i c_{ik}c_{il}^*)|\tilde{k}\rangle\langle\tilde{l}|. \tag{2.9}$$

Los operadores $|\tilde{k}\rangle\langle\tilde{l}|$ son fácilmente vistos como linealmente independientes, y por lo tanto debe ser que $\sum_i c_{ik}c_{il}^* = \delta_{kl}$. Esto asegura que podemos agregar columnas adicionales a c para obtener una matriz unitaria v tal que $|\tilde{\psi}_i\rangle = \sum_k v_{ik}|\tilde{k}\rangle$, donde hemos agregado vectores cero a la lista de $|\tilde{k}\rangle$. De manera similar, podemos encontrar una matriz unitaria w tal que $|\tilde{\varphi}_j\rangle = \sum_k w_{jk}|\tilde{k}\rangle$. Así,

$$|\tilde{\psi}_i\rangle = \sum_j u_{ij}|\tilde{\varphi}_j\rangle, \tag{2.10}$$

donde $u = vw^\dagger$ es unitaria.

2.3. Mediciones

Por otra parte vamos a ver como calcular las mediciones, pero primero necesitamos hablar de la traza.

Como es bien sabido la traza de una matriz se define como la suma de los elementos de su diagonal principal, es decir

$$tr(A) = \sum_i A_{ii}$$

La traza tiene propiedades interesantes, para empezar es sencillo ver que es lineal y además tenemos

$$tr(AB) = \sum_i \sum_k A_{ik} B_{ki} = \sum_i \sum_k B_{ik} A_{ki} = tr(BA)$$

(Sin más que renombrar los índices).

Sabiendo esto podemos ver que la traza es invariante por transformaciones UAU^* donde U es una matriz unitaria

$$tr(UAU^*) = tr(UU^*A) = tr(A)$$

Este tipo de transformaciones son cambios de base, lo cual hace que la traza sea una propiedad intrínseca del operador independiente de la base.

Vamos a ver finalmente, a partir de esta última, una propiedad útil para tratar con la matriz de densidad:

Proposición 9. Dado un estado cuántico $|\psi\rangle$ y una matriz A de dimensiones adecuadas, entonces

$$tr(A|\psi\rangle\langle\psi|) = \langle\psi|A|\psi\rangle$$

Como hemos visto la traza es independiente de la base, luego podemos construir una base ortonormal (Gram-Schimdt) tal que el estado $|\psi\rangle$ forme parte de ella. Si llamamos $|i\rangle$ al i-ésimo elemento de esta base entonces es claro que

$$A_{ii} = \langle i|A|i\rangle$$

para la matriz A escrita en esa base. Por lo tanto

$$tr(A|\psi\rangle\langle\psi|) = \sum_i \langle i|A|\psi\rangle\langle\psi||i\rangle$$

por ortogonalidad todos los sumandos se anulan excepto aquel para el que $|i\rangle = |\psi\rangle$ con lo que obtenemos el resultado.

Armados con estas propiedades de la traza veamos como calcular las probabilidades de obtener un resultado particular en una medición a partir de la matriz de densidad. Recordamos que dado un conjunto de operadores de medición $\{M_m\}$, para uno de los estados $|\psi_i\rangle$ tenemos que la probabilidad de medir m es

$$p(m|i) = \langle\psi_i|M_m^* M_m|\psi_i\rangle = tr(M_m^* M_m |\psi_i\rangle \langle\psi_i|)$$

Donde la notación $p(m|i)$ tiene el significado usual en probabilidad: Es la probabilidad de medir m condicionada a que el sistema se encuentre en el estado i.

La probabilidad de medir m sobre el conjunto de estados será entonces

$$p(m) = \sum_i p_i p(m|i) = \sum_i p_i tr(M_m^* M_m |\psi_i\rangle \langle\psi_i| = tr(M_m^* M_m \rho)$$

Donde en la última igualdad hemos aplicado la linealidad de la traza.

Podemos calcular también la matriz de densidad del sistema en su estado posterior a la medición, como sabemos tomando como estado inicial $|\psi_i\rangle$ y habiendo medido m el estado resultante tras la medición será

$$|\psi_i^m\rangle = \frac{M_m |\psi_i\rangle}{\sqrt{\langle\psi_i| M_m^* M_m |\psi_i\rangle}}$$

El conjunto de estados resultante consistirá de nuevo en n estados, uno por cada estado inicial, pero sus probabilidades no serán las mismas, estarán condicionadas a la información nueva de la que disponemos después de la medición. Es decir, necesitamos considerar las probabilidades

$$p(i|m) = \frac{p(m|i)p_i}{p_m} \quad \text{(Bayes)}$$

Por las igualdades anteriores para $p(m)$ y $p(m|i)$

$$p(i|m) = p_i \frac{tr(M_m^* M_m |\psi_i\rangle \langle\psi_i|)}{tr(M_m^* M_m \rho)}$$

Sabiendo esto el sistema tendrá matriz de densidad

$$\rho_m = \sum_i p(i|m) |\psi_i^m\rangle \langle\psi_i^m| = \sum_i p(i|m) \frac{M_m |\psi_i\rangle \langle\psi_i| M_m^*}{\langle\psi_i| M_m^* M_m |\psi_i\rangle}$$

Vemos que el denominador de los sumandos lo podemos escribir también en términos de la traza $tr(M_m^* M_m |\psi_i\rangle \langle\psi_i|)$, término que desaparece al sustituir la expresión que tenemos para $p(i|m)$ resultando en

$$\rho_m = \sum_i p_i \frac{M_m |\psi_i\rangle \langle\psi_i| M_m^*}{tr(M_m^* M_m \rho)} = \frac{M_m \rho M_m^*}{tr(M_m^* M_m \rho)}$$

Resultando en una expresión limpia y función de la matriz de densidad y el operador de medición correspondiente, independiente de los estados puros.

Veamos ahora la medición mediante un ejemplo:

Ejemplo 13. Vamos a considerar de nuevo los sistemas con matrices de densidad

$$\rho_1 = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \rho_2 = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Vamos a medir primero sobre la base computacional, los operadores serán

$$M_0 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad M_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

para el primer sistema tenemos

$$p(0) = tr(M_0^* M_0 \rho_1) = tr\left(\begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 0 \end{pmatrix}\right) = \frac{1}{2}$$

$$p(1) = tr(M_1^* M_1 \rho_1) = tr\left(\begin{pmatrix} 0 & 0 \\ 0 & \frac{1}{2} \end{pmatrix}\right) = \frac{1}{2}$$

para el segundo sistema

$$p(0) = tr(M_0^* M_0 \rho_2) = tr\left(\begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 0 \end{pmatrix}\right) = \frac{1}{2}$$

$$p(1) = tr(M_1^* M_1 \rho_2) = tr\left(\begin{pmatrix} 0 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}\right) = \frac{1}{2}$$

Como era esperable ambos sistemas son equivalentes a efectos de una medición sobre la base computacional, con resultados equiprobables ambos.

Supongamos ahora que en ambos casos se ha medido $|0\rangle$. ¿Son iguales también las matrices de densidad después de medir?

$$\rho_{1,0} = \frac{M_0 \rho M_0^*}{tr(M_0^* M_0 \rho_1)} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\rho_{2,0} = \frac{M_0 \rho M_0^*}{tr(M_0^* M_0 \rho_2)} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

Sí lo son, ya que conocemos en ambos casos sin duda que el sistema se encuentra en el estado $|0\rangle$.

2.4. Caracterización de la matriz de densidad

Hasta ahora hemos utilizado la definición de matriz de densidad dada en términos de los conjuntos de estados, pero podemos ver que las matrices de densidad son aquellas que cumplen unas propiedades

Teorema 4. Caracterización de la matriz de densidad. Un operador ρ es el operador de densidad para algún conjunto de estados si y solo si cumple

1. $tr(\rho) = 1$
2. ρ es positivo. ($\langle\psi|\rho|\psi\rangle \geq 0 \quad \forall\psi$; En particular es real)

Comenzamos suponiendo que partimos de un operador de densidad correspondiente a un conjunto de estados $\{(p_i, |\psi_i\rangle)\}$, entonces

$$tr(|\psi_i\rangle\langle\psi_i|) = 1$$

Ya que podemos tener una base ortonormal que incluya al estado $|\psi_i\rangle$, por lo tanto

$$tr(\rho) = tr(\sum_i p_i |\psi_i\rangle\langle\psi_i|) = \sum_i p_i = 1$$

con lo que se satisface la primera condición. Por otra parte para un estado arbitrario $|\phi\rangle$ tenemos

$$\langle\phi|\rho|\phi\rangle = \sum_i p_i \langle\phi| |\psi_i\rangle\langle\psi_i| |\phi\rangle = \sum_i p_i(\langle\phi| |\psi_i\rangle)(\langle\phi| |\psi_i\rangle)^* = \sum_i p_i |\langle\phi| |\psi_i\rangle|^2 \geq 0$$

con lo que se satisface la segunda condición.

Consideremos ahora un operador ρ satisfaciendo ambas propiedades, podemos diagonalizar ρ (al ser positivo es en particular hermítico luego diagonalizable) y obtener una representación

$$\sum_i \lambda_i |i\rangle\langle i|$$

donde además los autovalores λ_i son todos positivos por ser ρ positivo y de la condición de la traza obtenemos que $\sum_i \lambda_i = 1$, con lo que el conjunto de estados $\{(\lambda_i, |i\rangle)\}$ tendría como matriz de densidad asociada ρ.

Cabe observar que este conjunto de estados no es único para una matriz de densidad.

Ejemplo 14. Consideremos la matriz de densidad que habíamos obtenido con los estados de la base computacional cada uno de ellos con probabilidad $\frac{1}{2}$

$$\rho_1 = \frac{1}{2}I$$

Tiene un autoespacio de dimensión 2, el espacio completo, con autovalor $\frac{1}{2}$, Para obtener un conjunto de estados equivalente podemos obtener una base ortonormal distinta de los autoespacios, en este caso cualquier base del espacio completo nos sirve, por ejemplo la base de Hadammard $\{|+\rangle, |-\rangle\}$:

$$\frac{1}{2}|+\rangle\langle+| + \frac{1}{2}|-\rangle\langle-| = \frac{1}{2}\begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} + \frac{1}{2}\begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{pmatrix} = \frac{1}{2}I = \rho_1$$

con lo que también sería la matriz de densidad para el conjunto $\{(\frac{1}{2}, |+\rangle), (\frac{1}{2}, |-\rangle)\}$

Nos va a ser también útil poder caracterizar a partir de la matriz de densidad, si el sistema se encuentra en un estado puro, es decir $\rho = |\psi\rangle\langle\psi|$ o no.

Proposición 10. Caracterización de estados puros. Sea ρ una matriz de densidad entonces

1. $tr(\rho^2) \leq 1$
2. $tr(\rho^2) = 1$ si y solo si $\rho = |\psi\rangle\langle\psi|$ para algún estado $|\psi\rangle$.

Supongamos primero que $\rho = |\psi\rangle\langle\psi|$, en tal caso tenemos

$$\rho^2 = |\psi\rangle\langle\psi||\psi\rangle\langle\psi| = \rho$$

ya que $\langle\psi||\psi\rangle = 1$, entonces $tr(\rho^2) = tr(\rho) = 1$.

Por otra parte supongamos ahora que $\rho = \sum_{i=0}^{n} p_i |\psi_i\rangle\langle\psi_i|$ con $n > 1$, siendo sin pérdida de generalidad los estados $|\psi_i\rangle$ ortonormales.

Observamos que necesariamente $p_i < 1\ \forall i$ lo que implica, obviamente que $p_i^2 < p_i \quad \forall i$. Por la ortonormalidad tenemos

$$\rho^2 = \sum_i p_i^2 |\psi_i\rangle\langle\psi_i|$$

Como la traza es independiente de la base y la matriz es diagonal para la base ortonormal $\{|\psi_i\rangle\}$:

$$tr(\rho^2) = \sum_i p_i^2 < \sum_i p_i = 1$$

2.5. Traza parcial. Operador de densidad reducido

En ocasiones es interesante estudiar el estado de un subsistema, parte de un sistema más grande (sistema compuesto).

Partiendo de la matriz de densidad del sistema conjunto, podemos obtener una matriz de densidad para un subsistema usando la traza parcial

Definición 22. Traza parcial. Dados dos sistemas A y B, con $|a_1\rangle, |a_2\rangle$ estados posibles de A, y con $|b_1\rangle, |b_2\rangle$ estados posibles de B, la traza parcial sobre B, tr_B es un operador lineal cumpliendo

$$tr_B(|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|) = |a_1\rangle\langle a_2| \, tr(|b_1\rangle\langle b_2|)$$

Definición 23. Operador de densidad reducido. Dados dos sistemas A y B con un estado conjunto dado por el operador de densidad ρ^{AB}, el operador de densidad reducido para el sistema A es

$$\rho^A = tr_B(\rho^{AB})$$

Vamos a justificar con un caso particular sencillo por qué esta definición tiene sentido.

Ejemplo 15. Supongamos dos sistemas A, B, con estados no entrelazados, el sistema compuesto tendrá una matriz de densidad $\rho \otimes \sigma$, donde ρ es la matriz de densidad del sistema A y σ para el B. Entonces

$$\rho^A = tr_B(\rho \otimes \sigma) = \rho tr(\sigma) = \rho$$

La última igualdad es clara ya que la traza de una matriz de densidad es 1.

Primero veamos que en el caso en que $\rho = |\psi_1\rangle \langle\psi_1|$ y $\sigma = |\psi_2\rangle \langle\psi_2|$, es decir ambos sistemas se encuentran en un estado puro, la ecuación anterior es una aplicación directa de la definición.

Ahora supongamos primero que ρ no corresponde a un estado puro, entonces

$$\rho = \sum_i p_i |i\rangle \langle i|$$

Como hemos definido el operador tr_B como lineal entonces tenemos que

$$tr_B(\rho \otimes \sigma) = tr_B((\sum_i p_i |i\rangle \langle i|) \otimes |\psi_2\rangle \langle\psi_2|) =$$
$$tr_B((\sum_i (p_i |i\rangle \langle i|) \otimes |\psi_2\rangle \langle\psi_2|)) =$$
$$\sum_i tr_B(p_i |i\rangle \langle i| \otimes |\psi_2\rangle \langle\psi_2|) = \sum_i p_i |i\rangle \langle i| = \rho$$

Donde hemos aplicado el caso anterior. Tenemos que la ecuación que nos interesa es cierta para cualquiera matriz de densidad ρ y para una matriz de densidad σ correspondiente a un estado puro. Podemos volver a utilizar la linealidad de manera análoga a este caso para obtener finalmente el resultado general.

Vemos que la matriz reducida coindice con la matriz de densidad del sistema A cuando los estados de los dos sistemas son separables.

Veamos ahora como actúa sobre un estádo cuántico entrelazado.

Ejemplo 16. Consideremos el estado cuántico

$$|\psi\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$$

Que tiene matriz de densidad

$$\rho = |\psi\rangle\langle\psi| = \frac{|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 00|}{2}$$

Computamos la traza parcial sobre el primer cúbit:

Observamos que $|ii\rangle\langle jj| = (|i\rangle\langle j|) \otimes (|i\rangle\langle j|)$, luego podemos aplicar la propiedad de la definición tras hacer uso de la linealidad de la traza parcial para obtener:

$$\rho^1 = tr_2(\rho) = \frac{tr_2((|0\rangle\langle 0|) \otimes (|0\rangle\langle 0|))}{2} + \frac{tr_2((|1\rangle\langle 0|) \otimes (|1\rangle\langle 0|))}{2} +$$
$$\frac{tr_2((|0\rangle\langle 1|) \otimes (|0\rangle\langle 1|))}{2} + \frac{tr_2((|1\rangle\langle 1|) \otimes (|1\rangle\langle 1|))}{2} =$$
$$\frac{(|0\rangle\langle 0|)\langle 0||0\rangle)}{2} + \frac{(|1\rangle\langle 0|)\langle 1||0\rangle)}{2} + \frac{(|0\rangle\langle 1|)\langle 0||1\rangle)}{2} + \frac{(|1\rangle\langle 1|)\langle 1||1\rangle)}{2} =$$
$$\frac{|0\rangle\langle 0| + |1\rangle\langle 1|}{2} = \frac{I}{2}$$

Observamos que no se trata de un estado puro ($tr(\rho^2) = \frac{1}{2} < 1$), a pesar de partir de uno que en el sistema grande si lo era. Se trata del estado que hemos considerado en otros ejemplos.

2.6. Descomposición de Schmidt y purificación

Vamos a finalizar la sección dando un par de resultados útiles para trabajar con matrices de densidad: descomposición de Schmidt y purificación.

Teorema 5. Descomposición de Schmidt. Si $|\psi\rangle$ es un estado (puro) de un sistema compuesto AB, entonces existen estados ortonormales $|i_A\rangle$, $|i_B\rangle$ para los sistemas A y B respectivamente tales que

$$|\psi\rangle = \sum_i \lambda_i |i_A\rangle \otimes |i_B\rangle$$

con $\lambda_i > 0$ y $\sum_i \lambda_i^2 = 1$. A los coeficientes λ_i se les llama coeficientes de Schmidt.

Sean $|j\rangle$, $|k\rangle$ bases ortonormales de los sistemas A y B, con n y m elementos respectivamente entonces

$$|\psi\rangle = \sum_{jk} a_{jk} |j\rangle \otimes |k\rangle$$

para ciertos coeficientes a_{jk} ya que los elementos $|j\rangle \otimes |k\rangle$ formarán una base del sistema compuesto. Podemos pensar en estos coeficientes como una matriz a a la que podemos aplicar la descomposicion en valores singulares reducida, es decir tomando aquella parte correspondiente únicamente a los valores singulares no nulos. Si r de los valores singulares son no nulos entonces esto resultará en

$$a = udv^t$$

siendo d una matriz diagonal $r \times r$ y u, v matrices ortogonales de dimensiones $n \times r$ y $m \times r$ respectivamente, luego sus columnas seran un conjunto ortonormal de vectores.

Tenemos por lo tanto que

$$|\psi\rangle = \sum_{ijk} u_{ji} d_{ii} v_{ik} |j\rangle \otimes |k\rangle$$

y poniendo $|i_A\rangle = \sum_j u_{ji} |j\rangle$, $|i_B\rangle = \sum_k v_{ik} |k\rangle$ y $d_{ii} = \lambda_i$ concluimos observando que los conjuntos (de r estados) $|i_A\rangle$, $|i_B\rangle$ son ortonormales:

Si $|i_A\rangle$ y $|i'_a\rangle$ son dos estados distintos, por ejemplo del primero de estos conjuntos entonces por la ortonormalidad tanto de $|j\rangle$ como de los vectores de las columnas de u:

$$\langle i_A| |i'_A\rangle = \sum_j u_{ji} u_{ji'} \langle j| |j\rangle = \sum_j u_{ji} u_{ji'} = u_i \cdot u_{i'} = 0$$

Igualmente si $i = i'$ el producto es 1.

Una consecuencia importante de este teorema se aplica a las matrices de densidad reducidas ya que es facil observar que

$$\rho^A = \sum_i \lambda_i^2 |i_A\rangle \langle i_A|$$

$$\rho^B = \sum_i \lambda_i^2 |i_B\rangle \langle i_B|$$

Sea $\rho = |\psi\rangle \langle\psi| = \sum_i \lambda_i^2 (|i_A\rangle \otimes |i_B\rangle)(\langle i_A| \otimes \langle i_B|)$ entonces su traza parcial es

$$tr_B(\rho) = \sum_i \lambda_i^2 |i_A\rangle \langle i_A| \, tr(|i_B\rangle \langle i_B|)$$

El resultado se tiene de que $tr(|\psi\rangle\langle\psi|) = 1$, para cualquier estado $|\psi\rangle$. (Podemos ver esto utilizando de nuevo el hecho de que la traza es independiente de la base en que se represente la matriz y usando una base ortonormal que incluya al estado $|\psi\rangle$).

Ejemplo 17. Volviendo al ejemplo anterior observamos que

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle \otimes |0\rangle + \frac{1}{\sqrt{2}}|1\rangle \otimes |1\rangle$$

donde queda en la forma de la descomposición de Schmidt. Podemos evitarnos ahora los cálculos del ejemplo anterior y concluir directamente que la matriz de densidad reducida para cualquiera de los subsistemas (primer o segundo cúbit) es

$$\sum_i \lambda_i^2 |i_A a\rangle \otimes |i_B\rangle = \frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|) = \frac{I}{2}$$

La descomposición de Schmidt es importante para el siguiente concepto: Purificación. La idea es dado un sistema en un estado dado por un conjunto de estados puros, introducir más dimensiones, observarlo como parte de un sistema más grande, ficticio, de manera que en este sistema el estado sí sea puro. Más formalmente:

Definición 24. Purificación. Dado un sistema A en un estado representado por una matriz de densidad ρ_A, un estado puro $|\psi\rangle \in A \otimes R$ es una purificación de ρ_A si

$$tr_B(|\psi\rangle\langle\psi|) = \rho_A$$

En este contexto llamamos a R *sistema de referencia*.

Teorema 6. Existe una purificación para cualquier estado.

Sea ρ_A la matriz de densidad del estado para el que queremos encontrar una purificación, sabemos que entonces

$$\rho_A = \sum_{i=1}^{n} p_i |i_A\rangle$$

Tomamos entonces un conjunto ortonormal cualquiera también con n estados en el sistema de referencia y definimos

$$|AR\rangle = \sum_{i=1}^{n} \sqrt{p_i}|i_A\rangle|i_R\rangle$$

Observando la forma de descomposición de Schmidt de este estado es claro que es una purificación.

2.7. Superoperadores

En esta subsección vamos a tratar un conjunto muy general de operaciones sobre sistemas cuánticos. Hasta ahora la operación más común sobre un sistema se daba en forma de un operador unitario pero también modifica un estado, por ejemplo, una operación de medición; vamos a unificar estas y otras transformaciones bajo el concepto de superoperadores.

Definición 25. Superoperador. Un superoperador es una aplicación entre espacios de operadores de densidad. Es decir lleva una matriz de densidad a otra matriz de densidad.

Como ya hemos visto anteriormente tenemos para operaciones unitarias lo siguiente:

Proposición 11. Una transformación unitaria U define un superoperador

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

Por otra parte podemos definir transformaciones que se dan con una cierta probabilidad, esto nos será útil para definir un canal ruidoso.

Sean U_i operadores unitarios y p_i tales que $\sum_i p_i = 1$, entonces

$$\mathcal{E}(\rho) = \sum_i p_i U_i \rho U_i^*$$

es un superoperador

Ejemplo 18. Consideremos el superoperador

$$\mathcal{E}(\rho) = (1 - p)\rho + pX\rho X$$

Con probabilidad p al estado se le aplica la puerta X y con probabilidad $1 - p$ permanece igual. Esto por ejemplo nos es útil para modelar un posible error.

2.7.1. Entornos y operaciones cuánticas

La dinámica de un sistema cuántico cerrado (sistema físico que no intercambia energía ni información con su entorno) se describe mediante una transformación unitaria. Conceptualmente, podemos pensar en la transformación unitaria como una caja en la que entra el estado de entrada y de la cual sale el estado de salida. Para nuestros propósitos, el funcionamiento interno de la caja no nos concierne; podría ser implementado por un circuito cuántico, o cualquier otra cosa.

Una forma natural de describir la dinámica de un sistema cuántico abierto es considerarla como resultante de una interacción entre el sistema de interés, al que llamaremos el *sistema principal*, y un *entorno*, que juntos forman un sistema cuántico cerrado, como se ilustra en el lado derecho de la figura 2.1. En otras palabras, supongamos que tenemos un sistema en el estado ρ, que se envía a una caja acoplada a un entorno. En general, el estado final del sistema, $\mathcal{E}(\rho)$, puede no estar relacionado mediante una transformación unitaria con el estado inicial ρ. Suponemos (por ahora) que el estado de entrada del sistema-entorno es un estado producto, $\rho \otimes \rho_{\text{env}}$. Después de la transformación U de la caja, el sistema ya no interactúa con el entorno, y por lo tanto realizamos una traza parcial sobre el entorno para obtener el estado reducido del sistema solo:

$$\mathcal{E}(\rho) = \text{tr}_{\text{env}} \left[U \left(\rho \otimes \rho_{\text{env}} \right) U^* \right]. \tag{2.11}$$

Por supuesto, si U no involucra ninguna interacción con el entorno, entonces $\mathcal{E}(\rho) = \tilde{U}\rho\tilde{U}^*$, donde $\tilde{U}$ es la parte de U que actúa solo sobre el sistema.

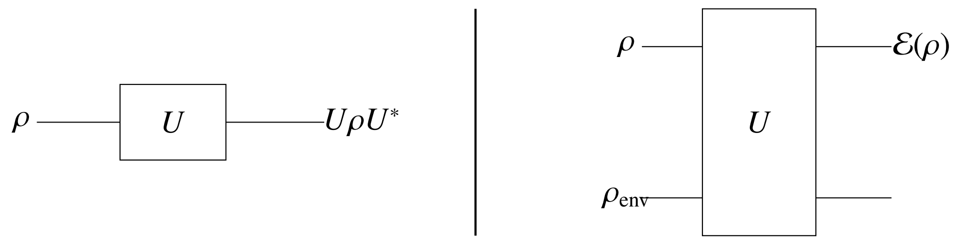
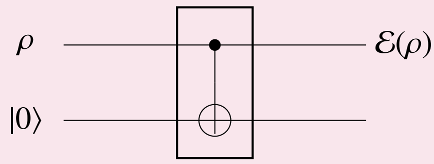


Figura 2.1: Transformación de un sistema cuántico cerrado (izquierda) y abierto (derecha).

Se hace una suposición importante en esta definición: asumimos que el sistema y el entorno comienzan en un estado producto. En general, por supuesto, esto no es cierto. Los sistemas cuánticos interactúan constantemente con sus entornos, acumulando correlaciones. Una forma en que esto se manifiesta es a través del intercambio de calor entre el sistema y su entorno. Si se deja a sí mismo, un sistema cuántico se relajará hasta alcanzar la misma temperatura que su entorno, lo que provoca la existencia de correlaciones entre ambos. Sin embargo, en muchos casos de interés práctico es razonable asumir que el sistema y su entorno comienzan en un estado producto. Cuando un experimentalista prepara un sistema cuántico en un estado específico, deshace todas las correlaciones entre ese sistema y el entorno. Idealmente, las correlaciones serán completamente destruidas, dejando al sistema en un estado puro. Incluso si este no es el caso, veremos más adelante que el formalismo de operaciones cuánticas puede describir incluso la dinámica cuántica cuando el sistema y el entorno no comienzan en un estado producto.

Otro problema que podría plantearse es: ¿cómo puede especificarse U si el entorno tiene casi infinitos grados de libertad? Resulta, muy interesante, que para que este modelo describa correctamente cualquier posible transformación $\rho \to \mathcal{E}(\rho)$, si el sistema principal tiene un espacio de Hilbert de dimensión d, entonces basta con modelar el entorno como estando en un espacio de Hilbert de no más de d^2 dimensiones. También resulta que no es necesario que el entorno comience en un estado mixto; un estado puro será suficiente.

Ejemplo 19. Como un ejemplo explícito del uso de la ecuación 2.11, consideremos el circuito cuántico de dos cúbits mostrado en la figura siguiente, en el cual U es una puerta controlada, con el sistema principal como el cúbit de control, y el entorno inicialmente en el estado $\rho_{\text{env}} = |0\rangle\langle 0|$ como el cúbit objetivo.



Insertando esto en la ecuación 2.11, se ve fácilmente que

$$\mathcal{E}(\rho) = P_0 \rho P_0 + P_1 \rho P_1, \tag{2.12}$$

donde $P_0 = |0\rangle\langle 0|$ y $P_1 = |1\rangle\langle 1|$ son operadores de proyección. Intuitivamente, esta dinámica ocurre porque el entorno permanece en el estado $|0\rangle$ solo cuando el sistema está en $|0\rangle$; de lo contrario, el entorno se invierte al estado $|1\rangle$.

2.7.2. Representación en suma de operadores

Las operaciones cuánticas se pueden representar en una forma elegante conocida como la *representación en suma de operadores*, que es esencialmente una reformulación de la ecuación 2.11 explícitamente en términos de operadores en el espacio de Hilbert del sistema principal. El resultado principal se motiva con el siguiente cálculo simple. Sea $|e_k\rangle$ una base ortonormal para el espacio de estados (de dimensión finita) del entorno, y sea $\rho_{\text{env}} = |e_0\rangle\langle e_0|$ el estado inicial del entorno. No hay pérdida de generalidad al suponer que el entorno comienza en un estado puro, ya que si comienza en un estado mixto, somos libres de introducir un sistema extra que purifique el entorno. Aunque este sistema extra es "ficticio", no afecta la dinámica experimentada por el sistema principal, y por lo tanto puede utilizarse como un paso intermedio en los cálculos. Así, la ecuación 2.11 puede reescribirse como:

$$\mathcal{E}(\rho) = \sum_k \langle e_k | U \left(\rho \otimes |e_0\rangle\langle e_0|\right) U^* | e_k \rangle \tag{2.13}$$

lo que se simplifica a:

$$\mathcal{E}(\rho) = \sum_k E_k \rho E_k^*, \tag{2.14}$$

donde $E_k \equiv \langle e_k | U | e_0 \rangle$ es un operador en el espacio de estados del sistema principal. La ecuación 2.14 es conocida como la *representación en suma de operadores* de $\mathcal{E}$. Los operadores $\{E_k\}$ son conocidos como *elementos de operación* para el superoperador $\mathcal{E}$.

Los elementos de operación satisfacen una restricción importante conocida como la relación de completitud. La relación de completitud surge de la necesidad de que la traza de $\mathcal{E}(\rho)$ sea igual a uno,

$$\begin{aligned} 1 &= \mathrm{tr}(\mathcal{E}(\rho)) \\ &= \mathrm{tr}\left(\sum_k E_k \rho E_k^*\right) \\ &= \mathrm{tr}\left(\sum_k E_k^* E_k \rho\right). \end{aligned}$$

Dado que esta relación es cierta para todo ρ, se sigue que debemos tener

$$\sum_k E_k^* E_k = I.$$

Esta ecuación es satisfecha por las operaciones cuánticas que preservan la traza. También existen operaciones cuánticas que no preservan la traza, para las cuales

$$\sum_k E_k^* E_k \leq I,$$

pero estas describen procesos en los que se obtiene información adicional sobre lo que ocurrió en el proceso mediante medición. Las aplicaciones $\mathcal{E}$ de la forma de la ecuación 2.14, para las cuales $\sum_k E_k^* E_k \leq I$, proporcionan nuestra segunda definición de un superoperador. Mostraremos a continuación que esta definición es esencialmente equivalente a la primera, la ecuación 2.11, y de hecho es ligeramente más general, ya que permite operaciones que no preservan la traza. A menudo tendremos la ocasión de movernos de un lado a otro entre estas dos definiciones; debe quedar claro por el contexto qué definición estamos utilizando en un momento dado.

La representación en términos de suma de operadores es importante porque nos proporciona un medio intrínseco de caracterizar la dinámica del sistema principal. La representación en términos de suma de operadores describe la dinámica del sistema principal sin tener que considerar explícitamente las propiedades del entorno; toda la información que necesitamos está encapsulada en los operadores E_k, los cuales actúan solo sobre el sistema principal. Esto simplifica los cálculos y a menudo proporciona un conocimiento teórico considerable. Además, muchas interacciones diferentes con el entorno pueden dar lugar a la misma dinámica en el sistema principal. Si solo interesa la dinámica del sistema principal, entonces tiene sentido elegir una representación de la dinámica que no incluya información irrelevante sobre otros sistemas.

Ejemplo 20. Supongamos que tenemos un sistema principal de un solo cúbit, interactuando con un entorno de un solo cúbit a través de la transformación

$$U = P_0 \otimes I + P_1 \otimes X,$$

donde X es la matriz de Pauli usual (actuando sobre el entorno), y $P_0 \equiv |0\rangle\langle 0|$, $P_1 \equiv |1\rangle\langle 1|$ son proyectores (actuando sobre el sistema). Vamos a dar el superoperador para este proceso, en la representación de suma de operadores, asumiendo que el entorno comienza en el estado puro $|0\rangle$. Es deci, el estado del sistema completo será

$$\rho_{\text{total}} = \rho \otimes |0\rangle\langle 0|,$$

donde ρ es el estado inicial del sistema principal.

Aplicamos la transformación unitaria U sobre el estado conjunto:

$$U(\rho \otimes |0\rangle\langle 0|)U^* = (P_0 \otimes I + P_1 \otimes X)(\rho \otimes |0\rangle\langle 0|)(P_0 \otimes I + P_1 \otimes X)^*.$$

$$U(\rho \otimes |0\rangle\langle 0|)U^* = P_0\rho P_0 \otimes |0\rangle\langle 0| + P_1\rho P_1 \otimes |1\rangle\langle 1|.$$

Para obtener el superoperador que actúa solo sobre el sistema principal, realizamos la traza parcial sobre el entorno:

$$\mathcal{E}(\rho) = \text{tr}_{\text{env}}\left[U(\rho \otimes |0\rangle\langle 0|)U^*\right].$$

Dado que $\text{tr}(|0\rangle\langle 0|) = 1$ y $\text{tr}(|1\rangle\langle 1|) = 1$, la traza parcial da como resultado:

$$\mathcal{E}(\rho) = P_0\rho P_0 + P_1\rho P_1.$$

Ahora identificamos los elementos de la operación en la representación de suma de operadores. Los operadores E_k son simplemente P_0 y P_1. Para ilustrar el significado de esta ecuación supongamos que ρ es un estado puro, es decir que $\rho = |\psi\rangle\langle\psi|$ siendo $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. En dicho caso tendríamos que

$$\rho = \alpha^2|0\rangle\langle 0| + \beta^2|1\rangle\langle 1| + \alpha\beta|0\rangle\langle 1| + \alpha\beta|1\rangle\langle 0|$$

Con lo que

$$\mathcal{E}(\rho) = \alpha^2|0\rangle\langle 0| + \beta^2|1\rangle\langle 1|$$

Es decir, se encuentra en el estado $|0\rangle$ con probabilidad α^2 y en el estado $|1\rangle$ con probabilidad β^2, por lo tanto esto sería equivalente a realizar una medición en la base computacional.

Teorema 7. Supongamos que $\{E_1, \ldots, E_m\}$ y $\{F_1, \ldots, F_n\}$ son elementos de operación que dan lugar a las operaciones cuánticas $\mathcal{E}$ y $\mathcal{F}$, respectivamente. Al agregar operadores nulos a la lista más corta de elementos de operación, podemos asegurar que $m = n$. Entonces, $\mathcal{E} = \mathcal{F}$ si y solo si existen números complejos u_{ij} tales que $E_i = \sum_j u_{ij} F_j$, y u_{ij} es una matriz unitaria de tamaño $m \times m$.

Demostración
La clave de la prueba es el teorema 3. Recordemos que este resultado nos dice que dos conjuntos de vectores $|\psi_i\rangle$ y $|\phi_j\rangle$ generan el mismo operador si y solo si

$$|\psi_i\rangle = \sum_j u_{ij} |\phi_j\rangle ,$$

donde u_{ij} es una matriz unitaria de números complejos, y completamos con estados adicionales 0 el conjunto de estados $|\psi_i\rangle$ o $|\phi_j\rangle$ más pequeño, de modo que ambos conjuntos tengan el mismo número de elementos. Este resultado nos permite caracterizar la libertad en las representaciones de suma de operadores. Supongamos que $\{E_i\}$ y $\{F_j\}$ son dos conjuntos de elementos de operación para la misma operación cuántica,

$$\sum_i E_i \rho E_i^* = \sum_j F_j \rho F_j^* \quad \text{para todo } \rho.$$

Llamemos Q al sistema original e intruduzcamos un sistema auxiliar R de la misma dimensión. Definimos

$$|e_i\rangle \equiv \sum_k |k_R\rangle E_i |k_Q\rangle , \quad |f_j\rangle \equiv \sum_k |k_R\rangle F_j |k_Q\rangle .$$

Y

$$\sigma = (\mathcal{I}_R \otimes \mathcal{E})$$

Es decir σ aplica $\mathcal{E}$ en el sistema Q y deja R intacto.

De esto sigue que $\sigma = \sum_i |e_i\rangle \langle e_i| = \sum_j |f_j\rangle \langle f_j|$, y por lo tanto existe una matriz unitaria u_{ij} tal que

$$|e_i\rangle = \sum_j u_{ij} |f_j\rangle .$$

Pero para cualquier $|\psi\rangle$ tenemos

$$E_i |\psi\rangle = \langle \tilde{\psi}| |e_i\rangle = \sum_j u_{ij} \langle \tilde{\psi}| |f_j\rangle = \sum_j u_{ij} F_j |\psi\rangle .$$

Por lo tanto,

$$E_i = \sum_j u_{ij} F_j.$$

Inversamente, suponiendo que E_i y F_j están relacionados mediante una transformación unitaria de la forma $E_i = \sum_j u_{ij} F_j$, unos cálculos sencillos muestran que la operación cuántica con elementos de operación $\{E_i\}$ es la misma que la operación cuántica con elementos $\{F_j\}$.

3. CÓDIGOS CORRECTORES CLÁSICOS

Para entender los códigos correctores cuánticos es necesario pasar por ciertos conceptos básicos de la teoría clásica de códigos correctores. En este capítulo se da una resumida introducción, sin entrar en detalles. En este capítulo utilizamos como referencia el libro *A Course In Error-Correcting Codes* **justesen2017course**, que trata en profundidad el tema. El capítulo es reducido en extensión dado que es un tema incluido en el grado pero se busca que la memoria sea autocontenida en lo posible.

3.1. ¿Qué son códigos correctores?

Un código corrector es un conjunto de reglas o procedimientos utilizados para detectar y corregir errores en los datos transmitidos o almacenados. En un sistema de comunicación, los datos pueden corromperse debido a diversos factores como el ruido o las interferencias. Los códigos correctores permiten identificar y corregir estos errores para recuperar los datos originales de manera confiable.

Dentro de los códigos correctores, los códigos lineales son los más comunes debido a su estructura algebraica sencilla y eficiencia en la corrección de errores. Además dada la importacia de codificar información digital, se trabaja casi siempre en el contexto de un cuerpo finito (que por ser discretos se ajustan más a la tarea de representar este tipo de información).

Definición 26. Código corrector lineal. Un código corrector lineal es un subespacio vectorial de un espacio vectorial sobre un campo finito.

La idea básica detrás de los códigos correctores es restringir el conjunto de palabras válidas (palabras código) para detectar aquellas que no lo son. Al limitar el conjunto de palabras código a un subespacio específico, se puede identificar y corregir errores al comparar la palabra recibida con las palabras código válidas más cercanas.

La noción de distancia que utilizaremos en este contexto es la distancia de Hamming.

Definición 27. Distancia de Hamming. La distancia de Hamming entre dos vectores es el número de posiciones en las que difieren. Es decir dados dos vectores v, u de dimensión n su distancia viene dada por

$$d(v, u) = |\{i : v_i \neq u_i, i \in 1 \dots n\}|$$

3.2. Parámetros del código

Para evaluar la eficacia y características específicas de un código corrector, es crucial entender ciertos parámetros clave que lo describen. Estos parámetros incluyen la dimensión, la longitud y la distancia del código. La dimensión se refiere a la cantidad de información original que puede ser codificada, la longitud al tamaño del vector codificado, y la distancia a la capacidad del código para detectar y corregir errores. Estos parámetros determinan no solo cómo se estructura el código, sino también su rendimiento en la detección y corrección de errores. A continuación, se definen estos parámetros de manera más formal.

Definición 28. Dimensión de un código. Llamamos dimensión del código, k, a la dimensión del código como espacio vectorial. Un código lineal de dimensión k puede codificar mensajes de longitud k.

Definición 29. Longitud de un código. Llamamos n a la dimensión del espacio vectorial que contiene al código, esto se corresponde con la longitud del vector codificado. Un código lineal de longitud n transforma un mensaje de k bits en un vector de n bits.

En general se tiene una función de codificación que transforma un mensaje de longitud k en una de las palabras del código n. La decodificación sin embargo consiste en encontrar primero la palabra del código más cercana a la recibida, normalmente con métodos adaptados para cada clase de códigos distinta que permiten hacer esta búsqueda de manera más efectiva, antes de revertir la operación de codificación y obtener el mensaje original.

Esta operación de buscar la palabra del código más cercana a la recibida está muy influenciada por la distancia del código:

Definición 30. Distancia de un código. Llamamos d a la mínima distancia de Hamming entre dos palabras código distintas en un código lineal.

La distancia de un código determina su capacidad para detectar y corregir errores.

Proposición 12. Capacidad correctora de un código. Un código con distancia mínima d puede detectar hasta $d - 1$ errores y puede corregir hasta $\left\lfloor \frac{d-1}{2} \right\rfloor$ errores.

Existe una limitación entre la cantidad de información que transmite una palabra (dada por k) y la capacidad de corrección. Si queremos tener más tolerancia a errores necesitaremos transmitir menos densidad de información, es decir más información redundante. Esta limitación está cuantificada por la cota de Singleton.

Proposición 13. Cota de Singleton. Para un código lineal con longitud n, dimensión k y distancia d, se cumple la siguiente desigualdad conocida como la cota de Singleton:

$$d \leq n - k + 1$$

Esta cota establece un límite superior para la distancia de un código dado su longitud y dimensión.

3.3. Matrices Generadoras y de Paridad

En la teoría de códigos correctores lineales, las matrices generadoras y de paridad son herramientas fundamentales que facilitan la codificación y decodificación de los mensajes.

3.3.1. Matriz Generadora

Una matriz generadora es una matriz que permite codificar los mensajes. Para un código lineal con dimensión k y longitud n, una matriz generadora G es una matriz de tamaño $k \times n$ que transforma un vector de mensaje $\mathbf{m} \in \mathbb{F}_q^k$ en una palabra código $\mathbf{c} \in \mathbb{F}_q^n$ mediante la multiplicación matricial:

$$\mathbf{c} = \mathbf{m}G$$

La matriz generadora tiene la propiedad de que las filas de G forman una base del subespacio vectorial correspondiente al código. Esto asegura que cualquier combinación lineal de los mensajes generará una palabra código válida.

3.3.2. Matriz de Paridad

La matriz de paridad, o matriz de control, es otra herramienta esencial utilizada principalmente en la decodificación. Para un código lineal, una matriz de paridad H de tamaño $(n-k) \times n$ satisface la siguiente relación para cualquier palabra código $\mathbf{c}$:

$$\mathbf{c}H^T = \mathbf{0}$$

Esta relación implica que cada palabra código debe ser ortogonal a las filas de H. La matriz de paridad define las restricciones adicionales que las palabras código deben satisfacer para ser consideradas válidas.

3.3.3. Relación entre las matrices generadora y de paridad

Para un código lineal con longitud n y dimensión k, las matrices generadora G y de paridad H están relacionadas de la siguiente manera. Si G es de tamaño $k \times n$ y H es de tamaño $(n - k) \times n$, entonces:

$$GH^T = \mathbf{0}$$

Esta relación asegura que las palabras código generadas por G cumplen las restricciones impuestas por H.

3.3.4. Síndrome

El síndrome es una herramienta que permite identificar si una palabra recibida contiene errores y, en caso afirmativo, proporciona información sobre la naturaleza de esos errores.

Dada una palabra recibida $\mathbf{r} \in \mathbb{F}_q^n$, el síndrome $\mathbf{s}$ se calcula mediante la matriz de paridad H:

$$\mathbf{s} = \mathbf{r}H^T$$

El síndrome $\mathbf{s}$ es un vector en $\mathbb{F}_q^{n-k}$ que contiene información sobre los errores presentes en la palabra recibida. Si $\mathbf{s} = \mathbf{0}$, entonces $\mathbf{r}$ es una palabra código válida y no contiene errores. Si $\mathbf{s} \neq \mathbf{0}$, entonces $\mathbf{r}$ contiene errores, y el valor del síndrome ayuda a identificar la posición y el tipo de errores.

3.4. Ejemplo. Código de repetición

En esta sección vamos a describir un código corrector muy simple pero que nos valdrá para ilustrar los conceptos anteriores y que además usaremos también para comenzar a estudiar los códigos correctores cuánticos.

Definición 31. Código de n-repetición. Un código de n-repetición es un código que repite cada símbolo del mensaje original n veces. Formalmente, dado un mensaje de un solo símbolo $m \in \mathbb{F}_q$, la palabra código correspondiente en un código de n-repetición es $\mathbf{c} = (m, m, \ldots, m) \in \mathbb{F}_q^n$.

Proposición 14. Los parámetros de un código de n-repetición son los siguientes:

- Longitud del código (n): n
- Dimensión del código (k): 1

- Distancia mínima del código (d): n

La capacidad correctora de un código de n-repetición es $\left\lfloor \frac{n-1}{2} \right\rfloor$, es decir, puede corregir hasta $\left\lfloor \frac{n-1}{2} \right\rfloor$ errores.

Proposición 15. Las matrices generadora y de control de paridad para un código de n-repetición son:

- Matriz generadora (G): Es una matriz de $1 \times n$ donde todos los elementos son 1.

$$G = \begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix}$$

- Matriz de control de paridad (H): Es una matriz de $(n-1) \times n$ que satisface $GH^T = \mathbf{0}$, y puede ser escrita como:

$$H = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -1 \end{pmatrix}$$

Ejemplo 21. Consideremos un código de 3-repetición sobre $\mathbb{F}_2$.

La matriz generadora y la matriz de paridad para este código son:

$$G = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$$

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

Supongamos que queremos codificar el mensaje $m = 1$.

La palabra código correspondiente es:

$$\mathbf{c} = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$$

Ahora, consideremos que durante la transmisión se introduce un error y la palabra recibida es $\mathbf{r} = \begin{pmatrix} 1 & 0 & 1 \end{pmatrix}$. Calculamos el síndrome $\mathbf{s}$:

$$\mathbf{s} = \mathbf{r}H^T = \begin{pmatrix} 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

El síndrome $\mathbf{s} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ indica que se ha producido un error. El primer 1 del sindrome indica que la primera y segunda posición son distintas, es decir hay un error en la primera o segunda posición. El segundo 1 del síndrome indica lo mismo para la segunda y tercera posición. Concluimos que el error está entonces en la segunda posición:

$$\mathbf{c} = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$$

De este modo, hemos ilustrado cómo funciona la codificación, la detección de errores usando el síndrome y la corrección de errores en un código de repetición sobre $\mathbb{F}_2$.

4. CÓDIGOS CORRECTORES CUÁNTICOS

Los códigos correctores cuánticos se sitúan en la vanguardia de la investigación en la confluencia de la teoría de la información cuántica y la teoría de códigos de corrección de errores. Estos códigos, diseñados para preservar la integridad de la información almacenada en cúbits, representan una herramienta crucial para contrarrestar los efectos adversos del ruido y otros errores inherentes en los sistemas cuánticos. En este capítulo, trataremos los fundamentos teóricos de los códigos correctores cuánticos.

Para proteger estados cuánticos contra los efectos del ruido, deseamos desarrollar códigos cuánticos correctores de errores basados en principios similares a los de la computación clásica. Sin embargo, existen importantes diferencias entre la información clásica y la cuántica que requieren la introducción de nuevas ideas para hacer posibles dichos códigos cuánticos correctores de errores. En particular, a primera vista nos enfrentamos a tres grandes dificultades:

De nuevo en este capítulo se utiliza principalmente como referencia *Quantum computation and quantum information* [1]

1. **No clonación:** La implementación del código de repetición a nivel cuántico mediante la duplicación del estado cuántico tres o más veces está prohibida por el teorema de no clonación 1.6.4. Incluso si la clonación fuera posible, sería imposible medir y comparar los tres estados cuánticos de salida del canal.

2. **Los errores son continuos:** Un continuo de errores diferentes puede ocurrir en un único cúbit. Determinar qué error ocurrió para corregirlo parece requerir una precisión infinita y, por lo tanto, recursos infinitos.

3. **La medición destruye la información cuántica:** En la corrección de errores clásica, observamos la salida del canal y decidimos qué procedimiento de decodificación adoptar. La observación en mecánica cuántica generalmente altera el estado cuántico en observación, lo que hace imposible su recuperación.

En resumen, la naturaleza cuántica plantea desafíos únicos que deben abordarse de manera innovadora para desarrollar efectivos códigos cuánticos correctores de errores.

4.1. Código de cambio de bit

El código de cambio de bit de es fundamental en la corrección de errores cuánticos. Diseñado para proteger un único cúbit contra errores de cambio de bit, este código se basa en la redundancia cuántica y el entrelazamiento para garantizar la integridad de la información en sistemas cuánticos.

Codificación del código de cambio de bit de 3 cúbits
Supongamos que enviamos cúbits a través de un canal que deja los cúbits intactos con una probabilidad de $(1 - p)$ y cambia los cúbits con una probabilidad de p aplicando la puerta X. Es decir, con probabilidad p, el estado $|\psi\rangle$ se transforma en el estado $X|\psi\rangle$, donde X es el operador de Pauli usual, ó el operador de cambio de bit. Este canal se llama canal de cambio de bit.

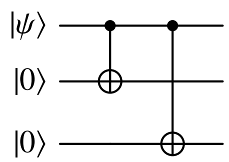
Supongamos que codificamos el estado de un único cúbit $a|0\rangle + b|1\rangle$ en tres cúbits como $a|000\rangle + b|111\rangle$. Una manera conveniente de escribir esta codificación es:

$$|0\rangle \rightarrow |0_L\rangle \equiv |000\rangle$$

$$|1\rangle \rightarrow |1_L\rangle \equiv |111\rangle$$

donde por L nos referimos a que es un cúbit lógico, más adelante se verá en más detalle pero básicamente es la correspondencia en el código de un estado (físico).

Se entiende que las superposiciones de estados de base se llevan a superposiciones correspondientes de estados codificados. Este circuito se representa como:



Decodificación del código de cambio de bit de 3 cúbits
Existe una forma de realizar la medición del síndrome que es útil en la generalización del código de tres cúbits. Esta consiste en realizar dos mediciones de manera independiente, la primera del observable Z_1Z_2 (es decir, $Z \otimes Z \otimes I$), y la segunda del observable Z_2Z_3(es decir, $I \otimes Z \otimes Z$). La primera medición, de Z_1Z_2, puede entenderse como la comparación entre el primer y segundo cúbits para ver si son iguales. Para comprender por qué es así, veamos que Z_1Z_2 tiene una descomposición espectral:

$$Z_1Z_2 = (|00\rangle\langle 00| + |11\rangle\langle 11|) \otimes I - (|01\rangle\langle 01| + |10\rangle\langle 10|) \otimes I,$$

que corresponde a una medición proyectiva con proyecciones $(|00\rangle\langle 00| + |11\rangle\langle 11|) \otimes I$ y $(|01\rangle\langle 01| + |10\rangle\langle 10|) \otimes I$. Por lo tanto, medir Z_1Z_2 puede entenderse como la comparación de los valores del primer y segundo cúbits, dando +1 si son iguales y -1 si son diferentes. De manera similar, medir Z_2Z_3 compara los valores del segundo y tercer cúbits, dando +1 si son iguales y -1 si son diferentes.
Lo crucial para el éxito de estas mediciones es que ninguna de ellas proporciona

información sobre las amplitudes α y β del estado cuántico codificado, y por lo tanto, ninguna de las mediciones destruye las superposiciones de estados cuánticos que deseamos preservar utilizando el código.
Usando estas medidas que acabamos de describir, podemos sacar los síndromes (que es lo que caracteriza al error) correspondientes a las distintas combinaciones de los resultados:

Z_1Z_2	Z_2Z_3	Síndrome
$+1$	$+1$	$I \otimes I \otimes I$
$+1$	-1	$I \otimes I \otimes X$
-1	$+1$	$X \otimes I \otimes I$
-1	-1	$I \otimes X \otimes I$

Otra manera equivalente de calcular los síndromes es sustituyendo las proyecciones Z_1Z_2 y Z_2Z_3 que comprueban los cambio de bit dos a dos por estas otras proyecciones que comprueban cada bit de manera independiente:

$P_0 \equiv |000\rangle\langle 000| + |111\rangle\langle 111|$ sin error
$P_1 \equiv |100\rangle\langle 100| + |011\rangle\langle 011|$ cambio de bit en el cúbit uno
$P_2 \equiv |010\rangle\langle 010| + |101\rangle\langle 101|$ cambio de bit en el cúbit dos
$P_3 \equiv |001\rangle\langle 001| + |110\rangle\langle 110|$ cambio de bit en el cúbit tres.

Observemos que $\langle\psi|P_i|\psi\rangle = 1, i = 1, 2, 3$ en el caso en que el bit i se haya cambiado (0 en caso contrario) y $\langle\psi|P_0|\psi\rangle = 1$ en el caso en el que no se haya producido ningún error (0 en caso contrario).
La tabla de Síndromes correspondiente a los resultados obtenidos de las proyecciones sería:

P_0	P_1	P_2	P_3	Síndrome
$+1$	0	0	0	$I \otimes I \otimes I$
0	$+1$	0	0	$X \otimes I \otimes I$
0	0	$+1$	0	$I \otimes X \otimes I$
0	0	0	$+1$	$I \otimes I \otimes X$

Una vez tenemos el síndrome calculado, de cualquiera de las dos formas anteriores, lo aplicamos al estado del sistema y obtendremos como resultado $a|000\rangle + b|111\rangle$ y para volver al estado inicial y conseguir decodificar el mensaje realizamos el mismo proceso que hemos hecho para codificar el estado $|\psi\rangle$ aplicando las puertas controladas $CNOT$ y recuperamos de nuevo el estado separable $|\psi\rangle \otimes |0\rangle \otimes |0\rangle$.

Ejemplo 22. Supongamos que Alice quiere enviar a Bob el siguiente estado $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ a través de un canal que deja los cúbits intactos con una probabilidad de $(1-p)$ y con una probabilidad de p, el estado $|\psi\rangle$ se transforma en el estado $X|\psi\rangle$. Entonces, Alice codifica el mensaje usando el código de cambio de bit de 3 cúbits y envía $|\psi_c\rangle = \alpha|000\rangle + \beta|111\rangle$ a Bob a través de dicho canal.
La palabra recibida por Bob es sin embargo $|\psi_r\rangle = \alpha|100\rangle + \beta|011\rangle$, de esta manera, si calculamos los síndromes usando las proyecciones Z_1Z_2 y Z_2Z_3 obtenemos

$$Z_1Z_2(|\psi_r\rangle) = -1$$

$$Z_2Z_3(|\psi_r\rangle) = 1$$

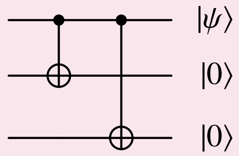
Si vamos a la tabla de síndromes obtenemos que el correspondiente a dicho error es:

$$X \otimes I \otimes I$$

y se lo aplicamos a la palabra recibida de manera que

$$X \otimes I \otimes I(|\psi_r\rangle) = \alpha|000\rangle + \beta|111\rangle = |\psi_c\rangle$$

Y ahora realizamos el proceso inverso de descodificación:



y obtenemos $|\psi_r\rangle = \alpha|000\rangle + \beta|111\rangle = |\psi\rangle \otimes |0\rangle \otimes |0\rangle$

4.2. Código de cambio de fase

El código de cambio de bit es interesante, pero no parece ser una innovación tan significativa en comparación con los códigos clásicos de corrección de errores, y deja muchos problemas abiertos, como son los tipos de errores distintos al cambio de bit pueden ocurrir en cúbits. Un canal cuántico ruidoso más interesante es el modelo de error de cambio de fase para un solo cúbit.

Codificación del código de cambio de fase de 3 cúbits
En este modelo de error, el cúbit se deja intacto con probabilidad $1-p$, y con probabilidad p se invierte la fase relativa de los estados $|0\rangle$ y $|1\rangle$. Más precisamente, el operador de cambio de fase Z se aplica al cúbit con probabilidad $p > 0$, por lo que el estado $a|0\rangle + b|1\rangle$

se transforma en el estado $a|0\rangle - b|1\rangle$ bajo el cambio de fase.
No existe un equivalente clásico para el canal de cambio de fase, ya que los canales clásicos no tienen ninguna propiedad equivalente a la fase. Sin embargo, hay una manera fácil de convertir el canal de cambio de fase en un canal de cambio de bit.
Supongamos que trabajamos en la base de cúbits: $|+\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $|-\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Con respecto a esta base, el operador Z lleva:

$$|+\rangle \to |-\rangle$$

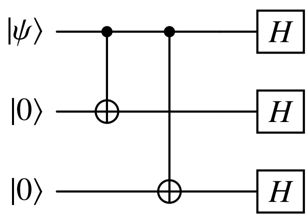
$$|-\rangle \to |+\rangle$$

Es decir, actúa exactamente como un cambio de bit con respecto a las etiquetas $+$ y $-$. Esto sugiere usar los estados:

$$|0_L\rangle \equiv |+++\rangle$$

$$|1_L\rangle \equiv |---\rangle$$

como estados lógicos cero y uno para protegerse contra errores de cambio de fase, de manera análoga al código de cambio de bit. Todas las operaciones necesarias para la corrección de errores, codificación, detección de errores y recuperación, se realizan de la misma manera que para el canal de cambio de bit, pero con respecto a la base $|+\rangle, |-\rangle$ en lugar de la base $|0\rangle, |1\rangle$. Para lograr este cambio se aplica el siguiente circuito:



Decodificación del código de cambio de fase de 3 cúbits

Se realiza de manera análoga a la decodificación realizada en el código de cambio de bit de 3 cúbits 4.1. La única diferencia es que hay que realizar una composición de las proyecciones con las puertas de Hadamard de la siguiente manera:

$$Pj \to H \otimes^3 P_j \otimes^3 H$$

$$Z_i Z_j \to H \otimes^3 Z_i Z_j \otimes^3 H$$

Donde $\otimes^n$ indica el producto tensorial de una puerta consigo misma n veces.

Ejemplo 23. Ejemplo. Código de cambio de fase de 3 cúbits Supongamos que Alice quiere enviar a Bob el siguiente estado $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ a través de un canal que deja intacto con probabilidad $1 - p$, y con probabilidad p se invierte la fase relativa de los estados $|0\rangle$ y $|1\rangle$.
Entonces, Alice codifica el mensaje usando el código de cambio de bit de 3 cúbits y envía $|\psi_c\rangle = \alpha|+++\rangle + \beta|---\rangle$ a Bob a través de dicho canal.
La palabra recibida por Bob es sin embargo $|\psi_r\rangle = \alpha|-++\rangle + \beta|+--\rangle$, de esta manera, si calculamos los síndromes usando las proyecciones Z_1Z_2 y Z_2Z_3 componiendo con la puerta de Hadamard como se indicaba anteriormente, obtenemos:

$$(H\otimes^3 Z_1Z_2 \otimes^3 H)(|\psi_r\rangle) = -1$$

$$(H\otimes^3 Z_2Z_3 \otimes^3 H)(|\psi_r\rangle) = 1$$

Por lo tanto, con lo que se ha producido un error de cambio de fase en el primer cúbit y para corregirlo aplicamos la puerta Z_1 a la palabra recibida

$$Z_1(|\psi_r\rangle) = \alpha|+++\rangle + \beta|---\rangle$$

por lo que la palabra enviada por Alice era: $\alpha|0_L\rangle + \beta|1_L\rangle$

4.3. Código de Shor

Existe un código cuántico simple que puede proteger contra los efectos de un error arbitrario en un solo cúbit, este código se conoce como el código de Shor [1], en honor a su inventor. El código es una combinación de los códigos de cambio de fase de tres cúbits y de cambio de bit, concretamente se emplea una jerarquía de niveles que se conoce como concatenación, la cuál nos permite obtener nuevos códigos a partir de otros antiguos.

La razón por la cual este código permite recuperarse de un error arbitrario es que si suponemos que el error se ha producido por una puerta unitaria U, entonces podemos escribir U como una combinación lineal de las puertas I (Identidad, no se ha producido error), X (Cambio de bit), Z (cambio de fase) e Y (que puede expresarse en términos de X y Z).

$$U = c_0I + c_1X + c_2Z + c_3Y = c_0I + c_1X + c_2Z + c_3(iXZ)$$

Donde c_i, $i = 0, \ldots, 3$ son constantes complejas.

Codificación del código de Shor
Primero, codificamos el cúbit usando el código de cambio de fase:

$$|0\rangle \to |+++\rangle$$

$$|1\rangle \rightarrow |- - -\rangle$$

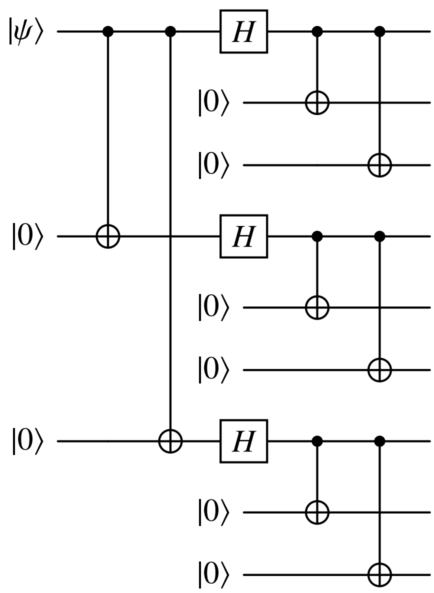
Luego, codificamos cada uno de estos cúbits usando el código de cambio de bit de tres cúbits:

$$|+\rangle \rightarrow (|000\rangle + |111\rangle)/\sqrt{2}$$
$$|-\rangle \rightarrow (|000\rangle - |111\rangle)/\sqrt{2}$$

El resultado es un código de nueve cúbits, con palabras de código dadas por:

$$|0\rangle \rightarrow |0_L\rangle \equiv \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{\sqrt{2^3}}$$
$$|1\rangle \rightarrow |1_L\rangle \equiv \frac{|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle}{\sqrt{2^3}}$$

Y su representación en forma de circuito sería:



Decodificación del código de Shor

La decodificación del algoritmo de Shor se realiza componiendo las descodificaciones vistas anteriormente en los códigos de cambio de bit y de cambio de fase (4.1 y 4.2 respectivamente). Es decir, para comprobar si se ha producido un cambio de bit en cualquiera de los nueves cúbits vamos a medir de manera indipendiente cada triplete como hemos definido en 4.1:

$$Z_1Z_2 \ \text{ y } \ Z_2Z_3$$
$$Z_4Z_5 \ \text{ y } \ Z_5Z_6$$
$$Z_7Z_8 \ \text{ y } \ Z_8Z_9$$

Para ver esto, supongamos que ocurre un cambio de bit en el primer cúbit, al igual que con el código de cambio de bit, realizamos una medición de Z_1Z_2 comparando los dos primeros cúbits, y si encontramos que son diferentes, concluimos que ocurrió un error de cambio de bit en el primer o segundo cúbit. Luego comparamos el segundo y tercer cúbit realizando una medición de Z_2Z_3. Encontramos que son iguales, por lo que no pudo haber sido el segundo cúbit el que cambió de bit. Concluimos que el primer cúbit debe haber cambiado de bit, y nos recuperamos del error cambiando nuevamente el primer cúbit a su estado original. De manera similar, podemos detectar y recuperarnos de los efectos de los errores de cambio de bit en cualquiera de los nueve cúbits en el código.

Manejamos de manera similar los cambios de fase en los cúbits. Supongamos que ocurre un cambio de fase en el primer cúbit. Tal cambio de fase cambia el signo del primer bloque de cúbits, cambiando $|000\rangle+|111\rangle$ a $|000\rangle-|111\rangle$, y viceversa. De hecho, un cambio de fase en cualquiera de los primeros tres cúbits tiene este efecto, y el procedimiento de corrección de errores que describimos funciona para cualquiera de estos tres posibles errores. La medición del síndrome comienza comparando el signo del primer y segundo bloques de tres cúbits, al igual que la medición del síndrome para el código de cambio de fase comenzó comparando el signo del primer y segundo cúbits. Por ejemplo, $(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)$ tiene el mismo signo (-) en ambos bloques de cúbits, mientras que $(|000\rangle - |111\rangle)(|000\rangle + |111\rangle)$ tiene signos diferentes. Cuando ocurre un cambio de fase en cualquiera de los primeros tres cúbits, encontramos que los signos del primer y segundo bloques son diferentes. La segunda y última etapa de la medición del síndrome es comparar el signo del segundo y tercer bloques de cúbits. Encontramos que son iguales, y concluimos que la fase debe haber cambiado en el primer bloque de tres cúbits. Nos recuperamos de esto cambiando el signo en el primer bloque de tres cúbits de vuelta a su valor original. Podemos recuperarnos de un cambio de fase en cualquiera de los nueve cúbits de manera similar.

Veamos con un ejemplo la corrección de un error que combina los dos tipos de errores.

Ejemplo 24. Tomemos $|\psi\rangle = |0\rangle$ que se codifica por $|0_L\rangle$. Supongamos que se produce un error dado por la transformación unitaria

$$U = XZ$$

en el primer cúbit. Aplicando U al primer cúbit tenemos el estado

$$\frac{(|100\rangle - |011\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{\sqrt{2^3}}$$

En primer lugar, para este caso $Z_{12} = -1$, $Z_{ij} = 1$ para el resto de $Z_i j$. Concluimos que se ha producido un error de cambio de bit en el primer cúbit y el primer bloque queda $(|000\rangle - |111\rangle)$.

Ahora corregimos el error de cambio de fase. Los otros dos bloques tienen signo +, con lo que concluimos que un error se ha producido en este primer bloque. Obtenemos de nuevo una decodificación correcta como $|0_L\rangle$.

4.4. Condiciones para la corrección de errores cuánticos

Al igual que en el caso de los códigos clásicos, un código cuántico va ser un subespacio vectorial de un espacio de los posibles estados.

Definición 32. Código corrector cuántico. Un código corrector cuántico (C) es un subespacio vectorial de un espacio de Hilbert $\mathcal{H}$ más grande.

Durante esta discusión asumiremos que podemos modelar el error introducido en el sistema mediante un superoperador $\mathcal{E}$. Igualmente pretendemos haciendo uso del código encontrar una transformación, también un superoperador, que encapsule todo el proceso de corrección. A este operador lo denotaremos de manera general $\mathcal{R}$ y lo llamamos operador de recuperación. El operador de recuperación incluirá normalmente, de manera análoga a los códigos correctores clásicos, una etapa de obtención de un síndrome y una etapa de corrección propiamente dicha.

Bajo estas hipotesis tendremos que ($\propto$ indica proporcionalidad).

$$\mathcal{R} \circ \mathcal{E}(\rho) \propto \rho$$

¿Bajo que condiciones es posible encontrar un operador de recuperación? Esto dependerá tanto del código como de los posibles errores que tengamos que corregir. El siguiente teorema (también conocido como condiciones de Knill-Laflamme

knill1997theory) responde a esta pregunta.

Pero antes enunciamos un resultado auxiliar previo, que no demostramos dado que es un teorema que trata de matrices (operadores lineales) en general:

Proposición 16. Descomposición polar. Sea A un operador lineal actuando sobre un espacio vectorial V, entonces existe un operador unitario U tal que

$$A = U\sqrt{A^*A} = \sqrt{AA*}U$$

Observamos que AA^* y A^*A son operadores semidefinidos positivos, y que para esta clase de operadores la raíz cuadrada se define de manera única: Si B es semidefinido positivo entonces tiene una descomposición espectral $B = \sum_i \lambda_i |i\rangle\langle i|$ con $\lambda_i \geq 0$, entonces $\sqrt{B} = \sum_i \sqrt{\lambda_i}|i\rangle\langle i|$, tomando las raices cuadradas positivas de los autovalores. Como es de esperar $\sqrt{B}\sqrt{B} = B$.

Teorema 8. Condiciones para la corrección de errores cuánticos. Sea C un código cuántico, P la proyección sobre este código, y $\mathcal{E}$ un superador con elementos $\{E_i\}$, es decir $\mathcal{E}(\rho) = \sum_i E_i\rho E_i^*$. Entonces existe un superoperador de recuperación $\mathcal{R}$ sí y solo sí

$$PE_i^*E_jP = \alpha_{ij}P$$

Donde los coeficientes α_{ij} forman una matriz hermítica (compleja).

Llamaremos errores a los operadores E_i de la descomposición, y en caso de existencia de de $\mathcal{R}$ diremos que constituyen un conjunto de errores corregible.

Demostración

Vamos a demostrar primero que si se dan estas condiciones entonces existe el superoperador de recuperación dando una construcción de este.

La matriz α de los coeficientes, por hipótesis, es hermítica luego diagonalizable:

$$d = u^*\alpha u$$

con d diagonal y u unitaria. Sean $F_k = \sum_i u_{ik}E_i$. Estos definen un conjunto de errores equivalente. Tenemos entonces:

$$\begin{aligned} PF_k^*F_lP &= \sum_{ij} u_{ki}^*u_{jl}PE_i^*E_jP \\ &= \sum_{ij} u_{ki}^*\alpha_{ij}u_{jl}P \\ &= d_{kl}P \end{aligned}$$

Esto nos da unas ecuaciones simplificadas en las que ya no tenemos que considerar cada pareja de errores si no simplemente cada error por separado, ya que d es diagonal:

$$
\begin{aligned}
PF_k^*F_kP &= d_{kk}P, \\
PF_k^*F_lP &= 0 \quad \forall k \neq l.
\end{aligned}
$$

Definiremos ahora la parte correspondiente a la medición del síndrome. Aplicamos la descomposición polar a F_kP para cada k:

$$F_kP = U_k\sqrt{PF_k^*F_kP} = \sqrt{d_{kk}}U_kP$$

Siendo U_k cierta matriz unitaria. Es decir en el subespacio del código F_k actúa como $\sqrt{d_{kk}}U_k$, y los proyectores sobre los espacios de la imágen del código por U_k serán $P_k = U_kPU_k^*$.

Estos, dadas las ecuaciones simplificadas, van a resultar ser espacios ortogonales:

$$P_lP_k = P_l^*P_k = \frac{U_lPF_l^*F_kPU_k}{\sqrt{d_{ll}d_{kk}}}$$

habiendo usado también $U_kP = F_kP/\sqrt{d_{kk}}$.

Por lo tanto tendríamos una medición proyectiva (añadiendo el espacio ortogonal a todos ellos restante en caso de que su suma directa no fuese el espacio completo para tener $\sum_k P_k = I$).

Si el resultado de la medición resulta ser el correspondiente a P_k sabremos que el error será F_k, y podremos corregirlo mediante U_k^*.

El proceso completo, detección y corrección queda entonces expresado por el superoperador

$$\mathcal{R}(\sigma) = \sum_k U_k^*P_k\sigma P_kU_k$$

Veamos entonces que $\mathcal{R}(\mathcal{E})(\rho) \propto \rho$ para estados ρ pertenecientes al código.

Primero observamos que

$$
\begin{aligned}
U_k^*P_kF_l\sqrt{\rho} &= U_k^*P_kF_lP\sqrt{\rho} \\
&= \frac{U_k^*U_kPF_k^*F_lP\sqrt{\rho}}{\sqrt{d_{kk}}} \\
&= \frac{d_{kl}}{\sqrt{d_{kk}}}P\sqrt{\rho} \\
&= \frac{d_{kl}}{\sqrt{d_{kk}}}\sqrt{\rho}
\end{aligned}
$$

Donde también hemos utilizado que $P\sqrt{\rho} = \sqrt{\rho}$.

Finalmente entonces

$$
\begin{aligned}
\mathcal{R}(\mathcal{E}(\rho)) &= \sum_{kl} U_k^* P_k F_l \rho F_l^* P_k U_k \\
&= \sum_{kl} \frac{d_{kl}^2}{d_{kk}} \rho \\
&= \left(\sum_k d_{kk} \right) \rho \propto \rho
\end{aligned}
$$

Ahora pasamos a la implicación opuesta. Supongamos que existe el superoperador de recuperación y se descompone de la siguiente manera

$$\mathcal{R}(\sigma) = \sum_i R_i \sigma R_i^*$$

Pongamos $\mathcal{E}_C(\rho) = \mathcal{E}(P\rho P)$. Como $P\rho P \in C$ para cualquier estado ρ (ya que lo proyectamos sobre el código), entonces

$$\mathcal{R}(\mathcal{E}_C(\rho)) \propto P\rho P \quad \forall \rho$$

De manera explícita

$$\sum_{ij} R_j E_i P\rho P E_i^* R_j^* = cP\rho P$$

Entonces el superador con componentes $\{R_j E_i P\}$ es el mismo que el superoperador con un único componente $\sqrt{c}P$, entonces se deben poder transformar de un conjunto de componentes a otro, que en este caso en particular implica:

$$R_k E_i P = c_{ki} P$$

para algún coeficiente complejo c_{ki}. Por lo tanto $PE_i^* R_k^* R_k E_j P = \alpha_{ij} P$ siendo $\alpha_{ij} = \sum_k c_{ki}^* c_{kj}$.

Si sumamos la ecuación anterior para todos los k, como $\mathcal{R}$ preserva la traza tenemos que $\sum_k R_k^* R_k = I$ y entonces

$$PE_i^* E_j P = \alpha_{ij} P$$

Además la matriz de coeficientes α_{ij}, α es hermítica porque $\alpha_{ji}^* = \alpha_{ij}$.

Ejemplo 25. Demostremos aplicando este resultado que el código de cambio de bit efectivamente es capaz de corregir cambios de bit. El operador de proyección sobre el código es

$$P = |000\rangle\langle 000| + |111\rangle\langle 111|$$

Por otra parte los únicos componentes del error serán X_i, entonces basta con

comprobar que

$$PX_i^* X_j P = c_{ij} P$$

Es sencillo ver que $c_{ij} = \delta_{ij}$ y que entonces siguiendo la construcción de la

demostración el superoperador de recuperación resultante es

$$\mathcal{R}(\sigma) = \sum_k PX_k \sigma X_k P$$

Probemoslo en el caso en que $\sigma = |001\rangle\langle 001|$, resultante de un error de cambio de bit en el último cúbit. En este caso solo el último sumando será distinto de 0 y resulta correctamente en $|000\rangle\langle 000|$.

4.5. Códigos estabilizadores

En esta sección vamos a introducir una construcción útil para obtener códigos correctores cuánticos. Nos basamos en otra referencia auxiliar para esta seccion, una tesis doctoral titulada Towards fault tolerant quantum computation **nigg2016** además del libro ya mencionado *Quantum computation and quantum information* [1].

La idea es determinar el espacio de posibles estados en el código mediante ciertos operadores que lo *estabilizan*, explicaremos más adelante a qué nos referimos con esto.

Empezamos con una definición esencial para esta construcción.

4.5.1. Grupo de Pauli

Definición 33. Grupo de Pauli. El grupo de Pauli Π es el grupo generado por las puertas de Pauli, es decir, el grupo generado por $\{X, Y, Z\}$. (Considerando la multiplicación matricial).

Por lo tanto Π es un subgrupo del grupo de matrices unitarias con coeficientes complejos de dimensión $2x2$.

El grupo de Pauli para n cúbits es

$$\Pi^n = \{A_1 \otimes A_2 \otimes \cdots \otimes A_n : A_i \in \Pi \quad \forall i = 1, \dots, n\}$$

Ejemplo 26. Todos los elementos del grupo de Pauli para un cúbit son

$$\Pi = \{\pm I, \pm iI, \pm X, \pm iX, \pm Z, \pm iZ, \pm Y, \pm iY\}$$

Proposición 17. Los (pares de) operadores del grupo de pauli son conmutativos o anti-conmutativos.

Demostración. Basta con verlo para las puertas de Pauli básicas.

$$XZ = -ZX$$

$$XY = -YX$$

$$YZ = -ZY$$

Obviamente las puertas conmutan consigo mismas y con la identidad.

Vamos a ver ahora a qué nos referíamos con estabilizar.

4.5.2. Estabilizadores

Definición 34. Estabilizador. Un subgrupo S de Π^n es un estabilizador de un espacio vectorial V dado si

$$P|\psi\rangle = |\psi\rangle \quad \forall P \in S, |\psi\rangle \in V$$

Proposición 18. Sea S un estabilizador de un espacio vectorial $V \neq \{0\}$, entonces $-I \notin S$

Demostración. Si $-I \in S$ entonces $|\psi\rangle = -I|\psi\rangle = -|\psi\rangle$ lo cual es imposible para $|\psi\rangle \neq 0$.

Proposición 19. Ses S un estabilizador de un espacio vectorial $V \neq \{0\}$, entonces S es abeliano.

Demostración. Los operadores del grupo de Pauli son conmutativos o anti-conmutativos, si existe un par de operadores A, B anti-conmutativos en S ($AB = -BA$) entonces $AB|\psi\rangle = |\psi\rangle = BA|\psi\rangle = -AB|\psi\rangle = -|\psi\rangle$. Con lo que llegamos a la misma conclusión contradictoria que en la prueba anterior.

Igualmente un subgrupo del grupo de Pauli con dichas características define el espacio del que es estabilizador, es la intersección de los subespacios propios asociados al autovalor 1 para cada operador del subgrupo.

Vamos a ver como aplicar este concepto a los códigos correctores cuánticos mediante un ejemplo.

Ejemplo 27. Vamos a considerar el código de cambio de bit de la sección anterior. Los estados codificados se encuentran en el espacio generado por $|000\rangle, |111\rangle$. Su estabilizador es

$$S = \{I, Z_{12}, Z_{23}, Z_{13}\}$$

Usábamos estos operadores para detectar y corregir si se había producido un error de cambio de bit ya que caracterizan el código. Comprobamos que

$$Z_{ij} |000\rangle = |000\rangle$$

$$Z_{ij} |111\rangle = |111\rangle$$

Para determinar que el estabilizador no es más grande utilizaremos el concepto de generador, y daremos un argumento por dimensionalidad más adelante.

Usamos los estabilizadores para definir el espacio de estados permisible para estados codificados. De esta manera si se produce un error, alguna de las condiciones dadas por el estabilizador ($P|\psi\rangle = |\psi\rangle$) no se cumplirá. Esto nos permitirá detectar y caracterizar un error.

No todos los códigos pueden darse en función de un estabilizador. Los que sí pueden, forman una subclase de códigos, los códigos estabilizadores.

Como hemos visto el código de cambio de bit es un código estabilizador, también lo es por tener una construcción similiar el código de cambio de fase (Siendo sus generadores $\{X_{12}, X_{23}\}$), y por ser concatenación de estos dos el código de Shor

4.5.3. Generadores de un estabilizador

Es útil fijar un conjunto generador del estabilizador, de esta manera podemos comprobar las condiciones en una selección reducida de operadores.

Ejemplo 28. Un conjunto generador del estabilizador del ejemplo anterior es

$$\{Z_{12}, Z_{23}\}$$

(Es el conjunto de mediciones que realizamos en la seccion anterior para determinar el síndrome).

El estabilizador juega un papel similar al de la matriz de paridad en la teoria de códigos correctores clásica. Se utiliza tanto para definir el código como para detectar posibles errores.

De una manera análoga a las matrices de paridad, el estabilizador, y en concreto el número de generadores que tenga, nos va a aportar información sobre la tasa del código.

Para el código de cambio de bit códificamos 1 cúbit por medio de 3 cúbits. Esto es lo equivalente en teoría clásica de códigos a $n = 3$, $k = 1$. La matriz de paridad de un código clásico con estos parámetros tendría $n - k$ filas, $n - k$ condiciones que determinan el codigo. Lo que en nuestro caso se corresponde con $n - k$ generadores.

Los estabilizadores nos acercan desde la cuántica a la teoría clásica de códigos correctores. Siguendo esta idea daremos más adelante una construcción de códigos correctores cuánticos a partir de 2 códigos correctores clásicos.

4.6. Cúbits y operadores lógicos

En el contexto de la computación cuántica, un cúbit lógico es una abstracción que permite la corrección de errores cuánticos. A diferencia de un cúbit físico, que es el estado cuántico de una partícula individual o sistema físico, un cúbit lógico está codificado en un conjunto de cúbits físicos y es manipulado de manera que los errores introducidos por el entorno puedan ser detectados y corregidos.

Formalmente, un cúbit lógico se define a través de un código de corrección de errores cuántico, que como sabemos es un subespacio del espacio de Hilbert de varios cúbits físicos, donde la información cuántica puede ser protegida contra ciertos tipos de errores.

Definición 35. Estados lógicos. Cúbits lógicos. Consideremos un sistema con n cúbits físicos, donde el espacio de Hilbert total es $\mathcal{H}^{\otimes n}$. Un código de corrección de errores cuánticos se define por un subespacio $C \subset \mathcal{H}^{\otimes n}$ de dimensión 2^k. Llamamos estados lógicos a los estados pertenecientes al código en el contexto de una aplicación que nos ofrece una correspondencia:

$$\begin{aligned} \cdot_L \colon \mathcal{H}^{\otimes k} &\to C \\ |\psi\rangle &\mapsto |\psi_L\rangle \end{aligned}$$

Podemos ver esta aplicación como una codificación. Marcamos con una L a los estados correspondientes en el código. En concreto para $k = 1$ tendremos un estado codificado que representa el estado de un cúbit, a esto le llamamos un cúbit lógico. En particular para los estados de la base canónica tenemos $|0_L\rangle$ y $|1_L\rangle$.

Con el objetivo de realizar cálculos cuánticos resistentes a errores (*Fault Tolerant*

Quantum Computation, *FTQC*), se realizan las operaciones sobre los estados lógicos:

Definición 36. Operadores lógicos. Un operador lógico $\bar{O}$ actúa sobre el subespacio del código C de manera que preserva la estructura del código. Es decir son aplicaciones lineales $\bar{O} : C \to C$.

Para un operador cuántico O que actúa sobre los cúbits físicos, denotamos $\bar{O}$ a su versión lógica que satisface:

$$(O(|\psi\rangle))_L = \bar{O}(\psi_L) \quad \forall\, |\psi\rangle \in \mathcal{H}^{\otimes k}$$

Dicho de otro modo, más visual, el siguiente diagrama conmuta (Los caminos que empiezan y acaban en los mismos vertices del diagrama son iguales)

$$\begin{tikzcd}
\mathcal{H}^{\otimes k} \arrow[r, "O"] \arrow[d, "\cdot_L"'] & \mathcal{H}^{\otimes k} \arrow[d, "\cdot_L"] \\
C \arrow[r, "\bar{O}"'] & C
\end{tikzcd}$$

Proposición 20. Propiedades de los operadores lógicos. Los operadores lógicos satisfacen las siguientes propiedades:

1. **Conmutatividad con Estabilizadores**: Los operadores lógicos $\bar{O}$ deben conmutar con los operadores estabilizadores del código. Los estabilizadores son operadores S_i que definen el subespacio del código y que actúan como la identidad en el subespacio del código:
$$\bar{O}S_i|\psi_L\rangle = S_i\bar{O}|\psi_L\rangle \quad \forall|\psi_L\rangle \in C. \quad \forall i$$

2. **Cierre bajo Multiplicación**: Si O_1 y O_2 son operadores cuánticos que actúan en el espacio de Hilbert de los cúbits lógicos, entonces sus versiones lógicas $\bar{O}_1$ y $\bar{O}_2$ deben cumplir:
$$\overline{O_1 O_2} = \bar{O}_1\bar{O}_2$$

3. **Anticonmutatividad (en el caso de operadores de Pauli)**: Si O_1 y O_2 son operadores de Pauli y conmutan/anticonmutan entre sí, entonces sus versiones lógicas $\bar{O}_1$ y $\bar{O}_2$ deben cumplir:
$$\bar{O}_1\bar{O}_2 = \bar{O}_2\bar{O}_1 \text{ ó } \bar{O}_1\bar{O}_2 = -\bar{O}_2\bar{O}_1 \text{ respectivamente}$$

Demostración.

1. Consideremos un código cuántico con estabilizadores S_i que definen el subespacio del código C. Los estabilizadores son operadores que actúan como la identidad en el subespacio del código, es decir, para cualquier estado lógico $|\psi_L\rangle \in C$,

$$S_i|\psi_L\rangle = |\psi_L\rangle \quad \forall i.$$

Para un operador lógico $\bar{O}$ que actúa sobre C,

$$\bar{O}|\psi_L\rangle \in C.$$

Queremos demostrar que

$$\bar{O}S_i|\psi_L\rangle = S_i\bar{O}|\psi_L\rangle \quad \forall|\psi_L\rangle \in C.$$

Dado que S_i actúa como la identidad en C,

$$S_i|\psi_L\rangle = |\psi_L\rangle,$$

y por lo tanto,

$$\bar{O}S_i|\psi_L\rangle = \bar{O}|\psi_L\rangle.$$

Además,

$$S_i\bar{O}|\psi_L\rangle = \bar{O}|\psi_L\rangle, \text{ dado que } \bar{O}|\psi_L\rangle \in C.$$

Entonces,

$$\bar{O}S_i|\psi_L\rangle = S_i\bar{O}|\psi_L\rangle,$$

2. Consideremos los operadores cuánticos O_1 y O_2 que actúan en el espacio de Hilbert de los cúbits lógicos. Sus versiones lógicas $\bar{O}_1$ y $\bar{O}_2$ actúan sobre el subespacio del código C. Queremos demostrar que:

$$\overline{O_1 O_2} = \bar{O}_1\bar{O}_2.$$

Por la definición de estado lógico que hemmos dado tenemos que:

$$((O_1 O_2)\,|\psi\rangle)_L = \overline{O_1 O_2}\,|\psi_L\rangle$$

Por otra parte tenemos además que:

$$((O_1 O_2)\,|\psi\rangle)_L = (O_1(O_2\,|\psi\rangle))_L = \overline{O_1}(O_2\,|\psi\rangle)_L$$

Y de nuevo por definición:

$$(O_2 |\psi\rangle)_L = \overline{O_2} |\psi_L\rangle$$

Con lo que $\overline{O_1}(O_2 |\psi\rangle)_L = \bar{O}_1\bar{O}_2 |\psi_L\rangle$

Y llegamos a que:

$$\overline{O_1 O_2} |\psi_L\rangle = \bar{O}_1\bar{O}_2 |\psi_L\rangle$$

3. Consideremos dos operadores de Pauli O_1 y O_2 que actúan en el espacio de Hilbert de los cúbits lógicos y que conmutan (la demostración es análoga para los operadores que anticonmutan): Dado que dichos operadores conmutan se tiene que:

$$((O_1 O_2) |\psi\rangle)_L = ((O_2 O_1) |\psi\rangle)_L$$

Además, por la propiedad anterior tenemos paralelamente que:

$$((O_1 O_2) |\psi\rangle)_L = \bar{O}_1\bar{O}_2 |\psi_L\rangle$$

$$((O_2 O_1) |\psi\rangle)_L = \bar{O}_2\bar{O}_1 |\psi_L\rangle$$

Luego obtenemos como resultado que $\bar{O}_1\bar{O}_2 |\psi_L\rangle = \bar{O}_2\bar{O}_1 |\psi_L\rangle$

Ejemplo 29. Para el código de repetición de tres cúbits, los estados lógicos $|0_L\rangle$ y $|1_L\rangle$ son:

$$|0_L\rangle = |000\rangle$$
$$|1_L\rangle = |111\rangle$$

El operador lógico X_L debe actuar en el subespacio del código de tal manera que intercambie los estados lógicos $|0_L\rangle$ y $|1_L\rangle$.

El operador que realiza esta transformación es:

$$X_L = X \otimes X \otimes X$$

Esto significa aplicar el operador de Pauli-X en cada uno de los tres cúbits físicos.

El operador lógico Z_L debe actuar en el subespacio del código de tal manera que introduzca una fase entre los estados lógicos $|0_L\rangle$ y $|1_L\rangle$, similar al operador de Pauli-Z que introduce una fase de -1 en el estado $|1\rangle$.

Para el código de repetición de tres cúbits, una elección adecuada para Z_L es:

$$Z_L = Z \otimes Z \otimes Z$$

Esto significa aplicar el operador de Pauli-Z en cada uno de los tres cúbis físicos.

Acción de Z_L:

$$Z_L |000\rangle = (Z \otimes Z \otimes Z) |000\rangle = |000\rangle = |0_L\rangle$$

$$Z_L |111\rangle = (Z \otimes Z \otimes Z) |111\rangle = (-1) |111\rangle = -|111\rangle = -|1_L\rangle$$

Aunque también se puede escoger de manera análoga $Z_L = Z_i, i = 1, 2, 3$ ya que estos operadores también nos introducen un cambio de fase entre los estados lógicos.

4.7. Calderbank-Shor-Steane codes

Los códigos CSS (Calderbank-Steane Shor) fueron el primer tipo de códigos de corrección de errores cuánticos descubiertos y forman un subconjunto de la familia de códigos estabilizadores.

Definición 37. Código CSS. Son códigos estabilizadores donde los generadores del estabilizador se pueden dividir en dos conjuntos, uno con solo estabilizadores de tipo X (operadores de Pauli con solo elementos X) y el otro con solo estabilizadores de tipo Z (operadores de Pauli con solo elementos Z).

La propiedad clave de estos códigos es que se pueden describir mediante dos códigos clásicos, $C_X \subset F_2^n$ y $C_Z \subset F_2^n$, que generan los vectores correspondientes a los estabilizadores X y Z; y esto nos permite tener una manera de construir códigos cuánticos a partir de códigos clásicos. La manera en que se procede para esto, es a partir de las matrices de paridad de los códigos clásicos H_X y H_Z respectivamente. El estabilizador construido a partir de ellas será

$$S = \left\langle \left\{ \prod_j (H_X)_{ij} X_j : i \right\} \cup \left\{ \prod_j (H_Z)_{ij} Z_j : i \right\} \right\rangle$$

Vamos a ver sin embargo que no cualquier par de códigos es válido para esta construcción, explorando un poco más la idea de correspondencia entre matrices de paridad y los generadores de un estabilizador siguiendo otra vez el ejemplo del código de repetición de tres bits.

La matriz de paridad de este código es

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

Ahora, nos damos cuenta de que las posiciones de los 1s en la matriz de verificación de paridad son exactamente las mismas que las de los Zs en los generadores de estabilizadores del código de cambio de bit de tres cúbits (Z_{12}, Z_{23}).

Si quisiesemos usar el mismo codigo para X como para Z en la construcción de un código CSS tendríamos que el estabilizador tendría generadores

$$\{X_{12}, X_{23}, Z_{12}, Z_{23}\}$$

Pero esto supone un problema dado que

$$X_{12}Z_{23} = (X \otimes X \otimes I)(I \otimes Z \otimes Z) = (X \otimes XZ \otimes Z) = -(X \otimes ZX \otimes Z) = -(Z_{23}X_{12})$$

Con lo que comprobamos que esos elementos no conmutan y el grupo generado por estos elementos no podría ser un estabilizador.

En general los elementos que suponen un problema para la conmutatividad son los productos XZ que anti-conmutan. Un numero par de estos productos no supondría un problema pero sí lo sería un número impar. Volviendo a las matrices de paridad esto se expresa limpiamente de la siguiente forma:

$$H_X H_Z^t = 0 \tag{4.1}$$

Proposición 21. Dados dos códigos $C_X \subset F_2^n$ y $C_Z \subset F_2^n$ con matrices de paridad H_X, H_Z respectivamente, entonces el grupo definido como 4.7 es un estabilizador $\iff H_X H_Z^t = 0$. (Y por tanto se puede construir un código de Calderbank-Steane-Shor).

A modo de observación, dado que la matriz generadora de un código dual es la transpuesta de la matriz de paridad del código original, podemos construir de manera totalmente equivalente un código CSS a partir de las matrices generadoras (G_X, G_Z) de dos códigos que cumplan

$$G_X G_Z^t = 0$$

Ejemplo 30. Veamos un ejemplo importante de códigos CSS, el código de 7 cúbits introducido por Steane. Específicamente, utilizamos un código lineal clásico, el código Hamming $[[7,4,3]]$, cuyas matrices generadora y de verificación de paridad son

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}, \quad H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Dado que $HH^T = 0$, podemos emplear el código Hamming para definir tanto los generadores de estabilizadores de tipo X como de tipo Z:

$$\begin{aligned} S_1 &= IIIXXXX, \\ S_2 &= IXIXIX, \\ S_3 &= XIXIXIX, \\ S_4 &= IIIZZZZ, \\ S_5 &= IZZIIZZ, \\ S_6 &= ZIZIZIZ. \end{aligned}$$

Cabe resaltar que esto supone una mejora con respecto del código de Shor, ya que estamos codificando 1 cúbit lógico mediante 7 cúbits físicos en vez de los 9 que utilizaba Shor.

5. EL CÓDIGO TÓRICO. CÓDIGOS TOPOLÓGICOS

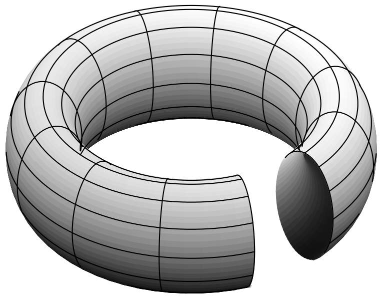


Figura 5.1: Figura del toro en 3D

El código tórico, introducido por primera vez por Aleksander Kitaev en 2003 [3] centra muchos de los esfuerzos e investigaciones que intentan implementar un modelo de computación cuántica tolerante a fallos (*FTQC*). En este capítulo definiremos este código y una clase más general a la que pertenece, los códigos topológicos además de la conexión que sugiere su nombre con la topología. Para este capítulo se toman como referencias los artículos [4], [5]. Además algunas de las imágenes que hemos utilizado para visualizar han sido generadas con la herramienta interactiva [6].

5.1. Definición mediante estabilizadores.

Vamos a visualizar el código tórico en una superficie plana, coloquemos cúbits en las aristas de una cuadrícula identificando el borde inferior del cuadrado con el borde superior y el izquierdo con el derecho. El código se definirá por sus estabilizadores que daremos mediante un conjunto generador.

Como se muestra en la figura 5.2 tenemos estabilizadores de dos tipos (X y Z separados, luego veremos que también es un código CSS.) Los estabilizadores de tipo X se corresponden a las caras del grafo que es la cuadrícula mientras que los de tipo Z afectarán a las aristas que rodean un eje.

Centrémonos primero en los estabilizadores de tipo X. Observamos que forman un bucle. También si consideramos el estabilizador correspondiente a dos caras adyacentes vemos que tambien forman un bucle más grande ya que para la arista compartida los dos operadores X se cancelarían. En general es fácil ver que todos los estabilizadores de este tipo están formados por uno o más bucles.

Lo mismo ocurre para los estabilizadores de tipo Z, solo que es más complicado de

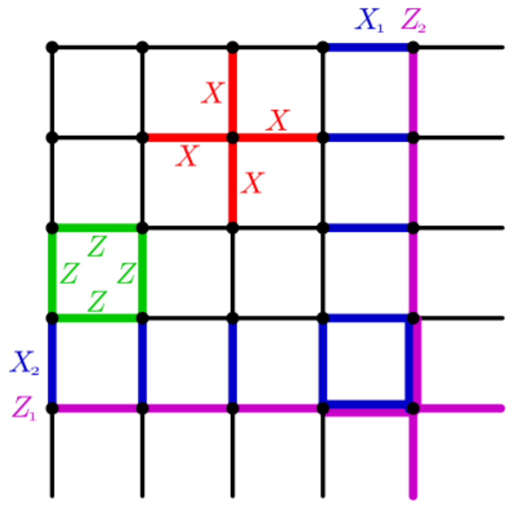


Figura 5.2: Cuadrícula en un toro. Estabilizadores y operadores lógicos del código tórico

visualizar ya que los bucles se dan en el grafo dual (aquel cuyos vértices son las caras del grafo original y existe una arista entre dos vértices si las caras en el grafo original eran adyacentes, es decir comparten una arista).

5.2. Corrección de errores

Nos centraremos en los errores de tipo de cambio de bit, siendo todo análogo para los errores de cambio de fase. Los errores de cambio de bit generarán un síndrome para los estabilizadores Z (que se identifican por vértices). La idea es que los síndromes aparecen siempre en pareja ya que un error en una arista afectará a dos vértices (los que une), igualmente si se producen errores en 2 aristas adyacentes en el vértice intermedio no se medirá un síndrome.

Corregiremos entonces las aristas a lo largo de caminos que unan las parejas de vértices que presentan un síndrome. Cualquier camino vale ya que podemos transformarlo en cualquier otro multiplicando por estabilizadores. Un ejemplo muy sencillo, en rojo aquellas aristas a las que le hemos aplicado X:

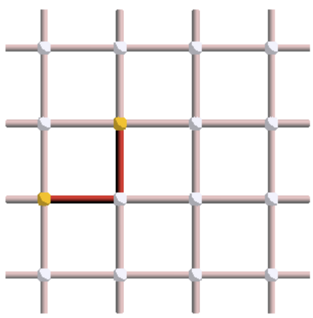


Figura 5.3: Camino original

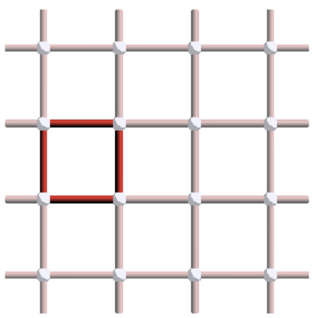


Figura 5.4: Estabilizador que se aplica

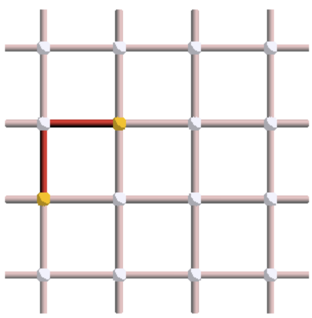


Figura 5.5: Camino resultante

Otra manera de verlo es que si aplicamos la puerta X a las aristas a lo largo de un camino cualquiera entre dos vértices en los que hemos medido una perturbación entonces o bien es el mismo camino que causó ambas perturbaciones o bien se forma un bucle, un camino cerrado que igualmente es un estabilizador.

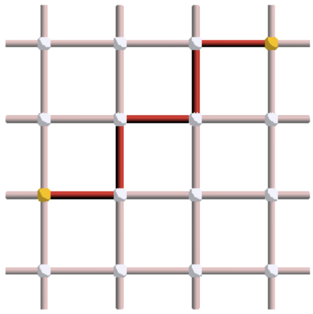


Figura 5.6: Camino original

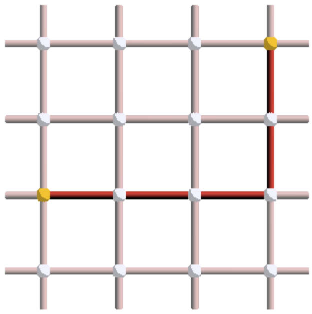


Figura 5.7: Camino elegido

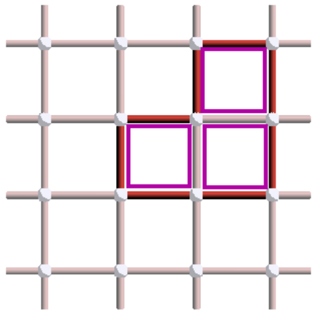


Figura 5.8: Bucle resultante donde se ve que es combinación de estabilizadores generadores.

De esta manera, cerrando bucles, se pueden corregir errores. ¿Cuál es el límite para la corrección de errores? El problema se va a dar cuando se producen bucles que *no* son un estabilizador ya que no podremos identificar ningún síndrome pero tendremos un estado diferente al original: Otro estado válido del código.

Estos van a corresponder con los **bucles no triviales** del toro, con los que enlazamos con la siguiente sección.

5.3. Operadores lógicos

En esta sección vamos a ver la conexión de este código con la topología algebráica. En topología algebráica dos bucles son equivalentes si se puede transformar un bucle en otro deformándolo de manera continua y los bucles triviales son aquellos que se pueden deformar en un sólo punto. En el caso del toro los bucles no triviales son los que dan vueltas alrededor de uno de los dos agujeros del toro.

En el caso del código tórico la equivalencia de bucles se da cuando podemos pasar de uno a otro aplicando un estabilizador y los bucles triviales son los estabilizadores.

Para cada tipo de estabilizador tenemos 2 bucles no triviales correspondientes de nuevo a cada uno de los agujeros del toro. El número de agujeros de una superficie es clave y sabemos que son una invariante topológica, también se llama a este número **número de Betti**.

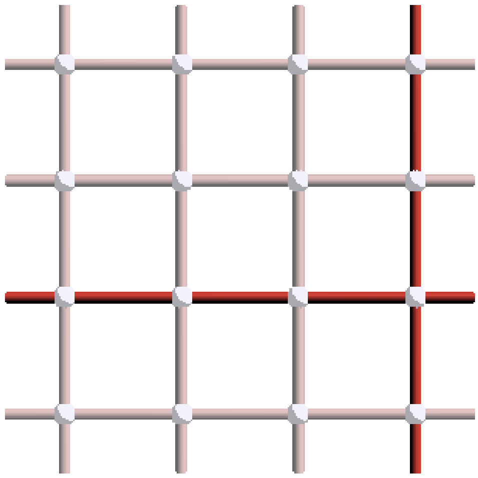


Figura 5.9: Bucles no triviales X

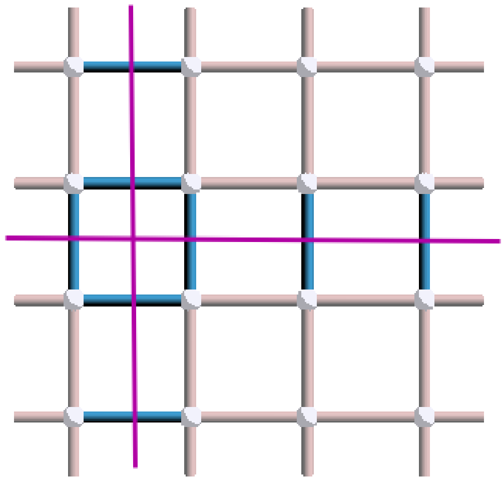


Figura 5.10: Bucles no trivuales Z

Como hemos dicho este tipo de bucle resulta en otro estado válido del código. De hecho corresponden a los operadores lógicos X y Z en 2 cúbits lógicos ya que podemos ver que cumplen $P^2 = I$ y $X_i Z_j = -Z_j X_i$.

Con esto podemos hablar ya de los parámetros de este código en función del tamaño de la cuadrícula L. Los cúbits se colocan en cada una de las aristas de la cuadrícula que son $2L^2$. Los cúbits lógicos son 2 porque tenemos 2 operadores lógicos de cada tipo. La distancia mínima es la longitud mínima de un bucle no trivial, porque es la mínima cantidad de errores con la que podemos pasar a otro estado válido, en este caso L.

Los parámetros del código son finalmente $[2L^2, 2, L]$.

5.4. Generalización: Códigos topológicos

Vamos a partir de la construcción de los códigos CSS, la idea ahora es facilitar la búsqueda de códigos que cumplan la condición necesaria para construir un código CSS. Hablábamos antes de la capacidad de construir códigos cuánticos a partir de códigos clásicos, pero realmente podemos realizar la construcción de un código CSS a partir de dos matrices A,B cualesquiera que cumplan:

$$AB = 0$$

ó, en términos de aplicaciones lineales, nos basta con buscar un par de aplicaciones lineales f, g tales que

$$f \circ g = 0$$

Este es el enfoque de los códigos topológicos, que nos va a permitir dar un par de tales aplicaciones a partir de una *teselación* de una *superficie*, y con ellas dos códigos clásicos con los que construir un código CSS.

Definición 38. Superficie. Una superficie S es un espacio topológico que, para cada punto $p \in S$, tiene un entorno U tal que existe una función homeomorfa $\phi : U \to V$, donde V es un subconjunto abierto de $\mathbb{R}^2$. Es decir, localmente, la superficie se comporta como un plano bidimensional.

Definición 39. Teselación de una superficie. El concepto de teselación puede extenderse a superficies más generales, como el toro. En este caso, una teselación de una superficie S es una colección de subconjuntos $\{T_i\}_{i\in I}$ de S tal que:

1. $\bigcup_{i\in I} T_i = S$, es decir, la unión de todas las teselas cubre completamente la superficie.
2. $\mathrm{Int}(T_i) \cap \mathrm{Int}(T_j) = \emptyset$ para $i \neq j$, es decir, las interiores de dos teselas cualesquiera son disjuntas (no se superponen).

Cada T_i en la colección se llama **tesela** o **mosaico**.

Los códigos topológicos son códigos CSS que se construye utilizando las propiedades de los teselados de superficies. La idea es partir de una superficie teselada, luego, se coloca un cúbit en cada borde del teselado y se define un estabilizador X para cada vértice que actúa sobre las cuatro aristas adyacentes y un estabilizador Z para cada cara que actúa sobre las cuatro aristas adyacentes.

El hecho de que esto defina un código CSS válido puede considerarse como proveniente de una propiedad del mapa de frontera de un teselado, a saber, "la frontera de la frontera es vacía". Más precisamente, dado el conjunto de caras F, bordes E y vértices V, se pueden definir los correspondientes espacios vectoriales sobre $\mathbb{F}^2$ como $\mathcal{F} = F^{\mathbb{F}^2}$, $\mathcal{E} = E^{\mathbb{F}^2}$ y $\mathcal{V} = V^{\mathbb{F}^2}$. Entonces, la relación de adyacencia entre ellos puede considerarse como un mapa lineal, llamado mapa de frontera, denotado como $\partial_{\mathcal{F}\to\mathcal{E}}$ y $\partial_{\mathcal{E}\to\mathcal{V}}$,

$$\mathcal{F} \xrightarrow{\partial_{\mathcal{F}\to\mathcal{E}}} \mathcal{E} \xrightarrow{\partial_{\mathcal{E}\to\mathcal{V}}} V.$$

Dado que $\partial_{\mathcal{F}\to\mathcal{E}} \circ \partial_{\mathcal{E}\to\mathcal{V}} = 0$, si escribimos estos mapas lineales como matrices (denotadas como $\mathrm{Mat}(\cdot)$), obtenemos exactamente la condición CSS en la Ecuación (4.1) al elegir

$$H_Z = \mathrm{Mat}(\partial_{\mathcal{F}\to\mathcal{E}})^T, \quad H_X = \mathrm{Mat}(\partial_{\mathcal{E}\to\mathcal{V}}).$$

Veremos todo esto con un ejemplo muy sencillo

5.5. Ejemplo

Consideramos para este ejemplo una teselación muy simple. Cuatro cuadrados sobre una sección también cuadrada del plano.

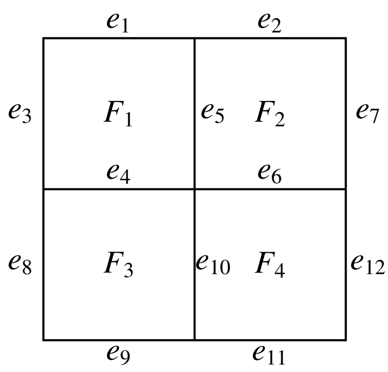


Figura 5.11: Caras y aristas de una teselación con cuatro teselas cuadradas.

Nombremos también los vertices de v_1 a v_9 de izquierda a derecha y de arriba a abajo.

Los espacios vectoriales $\mathcal{F}$, $\mathcal{E}$, $\mathcal{V}$ poodemos entenderlos como los espacios de los subconjuntos de F, E y V donde la suma de dos subconjuntos viene dada por

$$A + B = A \cup B - A \cap B$$

De esta manera son isomorfos a $\mathbb{F}^4$, $\mathbb{F}^{12}$, $\mathbb{F}^9$ respectivamente, para dar un isomorfismo concreto tomamos para cada subconjunto el elemento del espacio vectorial con unos en las posiciones dadas por los indices de los eslementos que estan presentes en el subconjunto. Por ejemplo

$$\{F_1, F_2\} \to (1, 1, 0, 0)$$

La aplicación frontera $\partial_{\mathcal{F}\to\mathcal{E}}$ lleva cada cara a las aristas que la rodean, por ejemplo $\{F_1\} \to \{e_1, e_3, e_4, e_5\}$ y $\{F_2\} \to \{e_2, e_5, e_6, e_7\}$. La linealidad de esta aplicación tiene sentido porque la frontera de dos caras adyacentes deja de incluir la arista que comparten.

$$\{F_1, F_2\} \to \{e_1, e_2, e_3, e_4, e_6, e_7\}$$

Igualmente la aplicación frontera $\partial_{\mathcal{E}\to\mathcal{V}}$ lleva a cada arista a los dos vertices que une.

La clave es que los conjuntos de aristas que son frontera de un conjunto de caras estarán necesariamente formados por uno o varios caminos cerrados. Para cada uno de estos caminos por separado de cada vertice salen exactamente dos aristas, luego en total siempre estaremos *contando* cada vértice un número par de veces.

Es por este motivo que para una teselación

$$\partial_{\mathcal{F}\to\mathcal{E}} \circ \partial_{\mathcal{E}\to\mathcal{V}} = 0$$

Vamos a ver las matrices correspondientes a estas aplicaciones lineales, primero la correspondiente a $\partial_{\mathcal{F}\to\mathcal{E}}$, escribiendo en cada fila la imagen de cada cara:

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

e igualmente para la apicacion $\partial_{\mathcal{E}\to\mathcal{V}}$ escribiendo en cada fila la imagen de cada arista:

$$H_Z{}^t = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Usamos la primera para los errores Z y la segunda para los errores X. Estas matrices determinan unos códigos lineales clásicos C_X, C_Z si consideramos que sus matrices generadoras son H_X y H_Z. Entonces tendrán parámetros $[12, 4]$ y $[12, 9]$ y el código CSS resultante $[12, 1]$. Esto es un código que necesita 12 cúbits auxiliares por cada cúbit original.

Los cúbits se van a corresponder con las aristas de la teselación. Los estabilizadores X actuarán sobre las aristas que rodeen las caras y los estabilizadores Z aquellos que rodeen un vertice.

Para corregir un error con este código podemos proceder como haríamos generalmente para un código estabilizador. Como vimos para el código de cambio de bit podemos realizar mediciones para un conjunto de generadores del grupo estabilizador y usar ese síndrome para detectar un error.

También podemos proceder como en lo explicado para códigos topológicos. Por ejemplo, supongamos que se han producido errores de cambio de bit en todas las aristas inferiores. Entonces mediante a los estabilizadores Z detectaríamos un síndrome relacionado con los vértices inferior izquierdo e inferior derecho. Para corregirlo, podríamos aplicar la puerta X en todos los cúbits de las aristas de cualquier camino entre ambos vértices marcados. Por supuesto, dado un error de cambio de fase el proceso es análogo.

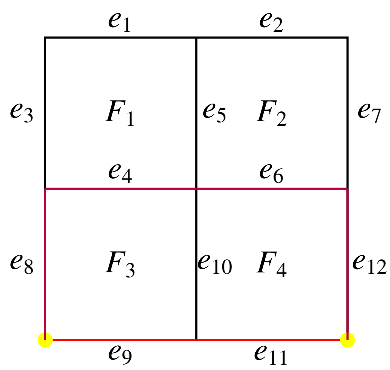


Figura 5.12: En rojo los errores de cambio de bit, en amarilo los vértices con síndrome y en morado una posible corrección.

CONCLUSIÓN

Este trabajo ha ofrecido una revisión detallada de los fundamentos matemáticos y teóricos que sustentan el código tórico como herramienta en la corrección de errores cuánticos. A partir de conceptos básicos de la computación cuántica y los códigos correctores, se ha explorado cómo los principios topológicos pueden aplicarse al diseño de sistemas capaces de proteger la información cuántica frente a errores.

El análisis realizado permite comprender cómo las propiedades matemáticas del código tórico, como su formulación mediante estabilizadores y operadores lógicos, están profundamente ligadas a conceptos topológicos, destacando su potencial dentro de los códigos cuánticos avanzados. Además, se ha puesto de manifiesto cómo la estructura teórica del código tórico lo convierte en un objeto de estudio relevante dentro del campo de la computación cuántica, con implicaciones para el desarrollo de nuevos enfoques en el manejo y protección de información cuántica.

BIBLIOGRAFÍA

[1] M. A. Nielsen e I. L. Chuang, *Quantum computation and quantum information*. 2001, vol. 54, pág. 60.

[2] C. Albornoz et al. “Xanadu Quantum Codebook.” (2023), dirección: `https://codebook.xanadu.ai` (visitado 04-09-2023).

[3] A. Y. Kitaev, “Fault-tolerant quantum computation by anyons,” *Annals of physics*, vol. 303, n.º 1, págs. 2-30, 2003.

[4] K. Fujii, *Quantum Computation with Topological Codes: from qubit to topological fault-tolerance*. Springer, 2015, vol. 8.

[5] C. Vuillot, “Fault-tolerant quantum computation: Theory and practice,” -, 2020.

[6] Pesha, Arthur. “An interactive introduction to the surface code.” (2023), dirección: `https://arthurpesah.me/blog/2023-05-13-surface-code` (visitado 24-11-2024).