

Introducción a la Computación Cuántica

Corrección de errores cuánticos

Sergio Montoya Ramirez
Kenneth Alejandro Rodriguez Peña

Contents

Chapter 1		Page 2
1.1	Introducción	2
1.2	Ejemplo: Phase Flip	2
1.3	Formalismos	3
1.4	Códigos Importantes	4
	Código de Shor — 4	
1.5	Códigos CSS	5
1.6	Conclusiones	5
Chapter 2	Bibliografía	Page 6
Chapter 3	Annexos	Page 7
3.1	Temas faltantes	7
3.2	Demostración de las condiciones de Corrección	7
3.3	Demostración de las condiciones de Corrección	7
3.4	Circuito de Shor	7
3.5	Shor Corrige errores arbitrarios	7
3.6	Shor Corrige errores arbitrarios	7

Chapter 1

1.1 Introducción

Uno de los puntos más difíciles para convertir la computación cuántica en una realidad es la gran sensibilidad de los qubits físicos. Dada la naturaleza cuántica de estos, mantener un estado y actuar sobre él sin que colapse o sea alterado por variables externas (como la temperatura o los cambios electromagnéticos) hace que la necesidad de manejar errores en la computación cuántica sea de primera categoría. Al mismo tiempo, el teorema de no clonación impide aplicar estrategias clásicas de corrección de errores, como la redundancia de información. Sin embargo, en 1995 (y posteriormente en 1996), Shor y Calderbank encontraron una forma de abordar este problema: distribuir la información en varios qubits y crear un código que permitiera interpretarlos correctamente para recuperar la información original.

Este proyecto tiene como objetivo resumir brevemente algunos de los puntos más importantes del campo *quantum error correction* (QEC). Para lograrlo, se abordarán tres aspectos fundamentales. En primer lugar, se presentará un ejemplo de código de corrección que permitirá entender los patrones generales del campo. A continuación, se examinará una teoría general de errores que proporciona los fundamentos matemáticos necesarios para comprender varios aspectos cruciales de QEC. Por último, se analizarán ejemplos de códigos importantes pertenecientes a la familia de códigos estabilizadores, como el código de Shor y la familia de códigos CSS. No obstante, dada la considerable diferencia entre la notación de estabilizadores y las notaciones aprendidas en el curso, se utilizarán principalmente nociones alternativas a la estabilización para explicarlos.

Esta ruta, aunque limitada por el espacio y tiempo disponibles, permitirá obtener una visión estructurada de las bases de este campo.

1.2 Ejemplo: Phase Flip

Imagine que tiene un canal con un ruido que, con probabilidad p , produce un *phase flip* en su qubit. Esto significa que, con probabilidad p , transforma $|\psi\rangle$ en $X|\psi\rangle$. Para este caso, podemos crear nuestro primer código cambiando a una base conocida, con elementos

$$|0_L\rangle = |000\rangle; \quad |1_L\rangle = |111\rangle.$$

Es decir, el estado $|\psi\rangle = a|0\rangle + b|1\rangle$ se convierte en $a|0_L\rangle + b|1_L\rangle$.

Ahora bien, también debemos contar con un procedimiento que nos permita determinar en cuál de los qubits ocurrió el cambio. Para ello, vamos a medir sin hacer distinciones entre estados, de modo que no se destruya el qubit, utilizando los siguientes operadores:

$$\begin{aligned} P_0 &\equiv |000\rangle\langle 000| + |111\rangle\langle 111|, \\ P_1 &\equiv |100\rangle\langle 100| + |011\rangle\langle 011|, \\ P_2 &\equiv |010\rangle\langle 010| + |101\rangle\langle 101|, \\ P_3 &\equiv |001\rangle\langle 001| + |110\rangle\langle 110|. \end{aligned}$$

De esta manera, podemos identificar en qué qubit ocurrió el error y, por lo tanto, aplicar nuevamente X para regresar al estado original.

Ahora bien, ¿cómo sabemos que este procedimiento funciona mejor que el caso original? Para ello, definamos la fidelidad, de modo que podamos analizar el peor caso en este canal.

Definition 1.2.1: Fidelidad

Sea ρ la acción de un canal; entonces, la fidelidad se define como

$$F = \sqrt{\langle \psi | \rho | \psi \rangle}, \quad (1.1)$$

donde $|\psi\rangle$ es el estado de interés (normalmente, el estado inicial).

Note que este canal se comporta como

$$\rho = (1 - p) |\psi\rangle \langle \psi| + p X |\psi\rangle \langle \psi| X.$$

Por lo tanto, tiene una fidelidad de:

$$F = \sqrt{(1 - p) + p \langle \psi | X | \psi \rangle \langle \psi | X | \psi \rangle}.$$

Observe que $\langle \psi | X | \psi \rangle \geq 0$, por lo que, en general para este canal, la fidelidad mínima estará definida por la amplitud de probabilidad del estado de interés. En nuestro caso original, $F = \sqrt{1 - p}$. Por otro lado, para el caso de nuestro código, nos interesan únicamente los procesos en los que haya ocurrido un error o menos, ya que nuestro código solo corrige en ese caso. La probabilidad de que esto ocurra es la probabilidad de que no ocurra ningún error, es decir, $(1 - p)^3$, más tres veces la probabilidad de que ocurra un solo error, es decir, $3p(1 - p)^2$. Así, la probabilidad total es

$$(1 - p)^3 + 3p(1 - p)^2 = 1 - 3p^2 + 2p^3,$$

y, en consecuencia, la fidelidad mínima sería $F = \sqrt{1 - 3p^2 + 2p^3}$, que es mayor que la original siempre y cuando $p < \frac{1}{2}$.

Este ejemplo nos proporciona algunas de las bases necesarias para comprender el tema de la corrección cuántica de errores. En particular, muestra el funcionamiento general de estos mecanismos, que consiste en codificar la información, distribuirla en más qubits de los estrictamente necesarios y desarrollar un procedimiento que revierta los efectos de los errores. Sin embargo, aún necesitamos mayor formalidad en lo que estamos tratando.

1.3 Formalismos

En el caso anterior vimos un ejemplo muy concreto de un código que mejoraba la fiabilidad en un canal con un ruido concreto. Sin embargo, este ejemplo particular se aprovechaba de las características del ruido para representar la corrección del error y el aumento en la fiabilidad. Ahora nos interesa construir una teoría más general de los errores.

Para comenzar es importante definir lo que es un código

Definition 1.3.1: Código

Sea \mathcal{L} un espacio de tamaño fijo, decimos que \mathcal{M} un subespacio de algún espacio \mathcal{B}^n es un código que codifica \mathcal{L} si existe una operación $\mathcal{V} : \mathcal{L} \rightarrow \mathcal{M}$ que se le conoce como codificador.

Esto es similar a lo que vimos arriba, es un subespacio de n qubits que pueden codificar la información de algún conjunto. Ahora bien, esto es una definición general, estos mismos códigos podrían tomarse simplemente como maneras de dar significado a un subespacio.

Definition 1.3.2: Código que Corrige errores

Sea C un código y \mathcal{E} un procedimiento que define un error. Decimos que C corrige \mathcal{E} si existe un procedimiento \mathcal{R} tal que

$$\forall p \in C : (\mathcal{R} \circ \mathcal{E})(p) \propto p \quad (1.2)$$

Una nota importante que ver es que el \propto nos representa que realmente la operación de recuperación no nos devuelve exactamente al qubit original. Sin embargo, si nos debe devolver a un qubit que codifique la misma información que el anterior.

Theorem 1.3.1 Condiciones para que un Código Corrija un error

Sea C un código y P el proyector a C . Suponga que \mathcal{E} es una operación cuántica con elementos $\{E_i\}$. Una condición necesaria y suficiente para que exista una operación \mathcal{R} que corrija \mathcal{E} en C es

$$PE_i^\dagger E_j P = \alpha_{ij} P \quad (1.3)$$

para alguna matriz hermitica α de números complejos.

Nota: La demostración de este teorema lo puede encontrar en 3.2.

Sin embargo, esta es solo una manera de saber que un código corrige un error. Algo importante es poder encontrar el mejor código para cierto error. Para esto podemos usar la cota de Hamming que nos dice el límite teórico de eficiencia de un código que corrija un error.

Theorem 1.3.2 Cota de Hamming

Sea C un código que codifica k qubits en n qubits y que puede corregir cualquier subconjunto de t errores. Particularmente, asuma sin pérdida de generalidad que corrige $j \leq t$ errores. Entonces se cumple la cota

$$\sum_{j=0}^t \binom{n}{j} 3^j 2^k \leq 2^n \quad (1.4)$$

Nota: La deducción de esta cota la puede encontrar en 3.2.

Como última nota importante antes de pasar a los códigos que vamos a explicar es el formalismo de estabilizadores. Este es importante pues muchos de los códigos más importantes en QEC son basados en este mismo formalismo.

Las bases del formalismo de estabilizadores consiste en mirar un estado no desde sus características sino desde los operadores que lo estabilizan. Un operador se dice que estabiliza a un estado si el estado es un eigenvector con eigenvalue $+1$. Es decir, S estabiliza $|\psi\rangle$ si $S|\psi\rangle = |\psi\rangle$.

Example 1.3.1 (Ejemplo de Estabilizador)

En nuestro primer ejemplo para el qubit

$$|\psi\rangle = a|000\rangle + b|111\rangle$$

Los operadores definidos P_i eran estabilizadores de este. Puesto que al aplicarlo nos devolvían el mismo qubit sin cambios.

Una de las características más importantes es que al definir de esta manera un estado estamos construyendo grupos y por tanto podemos usar teoría de grupos para realizar las interpretaciones de los diversos códigos. Esto es parte de lo que hace tan increíblemente fuerte el formalismo de estabilizadores y lo que nos va a permitir hablar de los siguientes códigos que usan este formalismo.

1.4 Códigos Importantes

A continuación, veremos ejemplos de códigos de corrección de errores importantes en el campo. Todos estos códigos son estabilizadores. Mencionaremos las operaciones que estabilizan estos códigos para ejemplificar su funcionamiento. Sin embargo, debido a la extensión de los temas y a lo novedoso de la notación de estabilizadores, utilizaremos principalmente explicaciones que no dependan de ellos.

1.4.1 Código de Shor

Este fue uno de los primeros códigos de corrección de errores, diseñado por Shor en 1995 en el artículo *Scheme for reducing decoherence in quantum computer memory*. Este código permite corregir errores arbitrarios en un solo qubit.

El código se define mediante el circuito que puede verse en el anexo 3.4, transformando $|0\rangle \rightarrow |++\rangle$ y $|1\rangle \rightarrow |--\rangle$. La operación de recuperación se divide en dos partes:

1. Determinar el qubit afectado. Aquí resulta crucial que este código sea estabilizador, ya que utilizamos sus operadores para identificar el qubit donde ocurrió el error.
2. Una vez identificado el qubit erróneo, se procede a corregirlo. La corrección es un término genérico, ya que este código corrige errores arbitrarios. Puede consultarse el anexo 3.5 para más detalles.

1.5 Códigos CSS

Los códigos CSS (Calderbank-Shor-Steane) son un subconjunto de los códigos estabilizadores. Informalmente, permiten construir códigos que corrigen los mismos errores a partir de dos códigos existentes, utilizando menos qubits para la codificación.

Definition 1.5.1: Códigos CSS

Sean C_1 y C_2 códigos lineales $[n, k_1]$ y $[n, k_2]$ (cerrados bajo suma módulo 2) tales que:

- $C_2 \subset C_1$,
- C_1 y C_2^\perp corrigen t errores.

Definimos el código CSS de C_1 sobre C_2 como el código $[n, k_1 - k_2]$ denotado por $CSS(C_1, C_2)$ mediante el siguiente procedimiento. Sea $x \in C_1$, entonces definimos el estado:

$$|x + C_2\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x + y\rangle$$

donde $+$ denota suma módulo 2. Nótese que si $x' \in C_1$ y $x - x' \in C_2$, entonces $|x + C_2\rangle = |x' + C_2\rangle$. El código $CSS(C_1, C_2)$ corresponde al espacio generado por los estados $|x + C_2\rangle$, los cuales forman un conjunto ortonormal.

Nota: Para más detalles sobre la igualdad $|x + C_2\rangle = |x' + C_2\rangle$, consúltese el anexo 3.6.

La ventaja de esta construcción es evidente: si encontramos un subcódigo que corrija los mismos t errores, podemos reducir considerablemente el tamaño del código original. Un ejemplo notable de código CSS es el código de Shor.

1.6 Conclusiones

En este trabajo se estudiaron y explicaron los conceptos básicos de la corrección de errores cuánticos. Con ello, se comprendió que un código cuántico permite distribuir la información en múltiples qubits, lo cual ofrece ventajas significativas para manejar los errores característicos de la computación cuántica. Además, se presentaron las bases teóricas de la corrección de errores y se enunciaron algunos de sus teoremas más importantes, que permiten determinar si un código corrige un error determinado. Finalmente, se analizaron de manera general dos casos relevantes: el código de Shor y la familia de códigos CSS. Si bien esta es una introducción incompleta al tema, resulta suficiente para obtener una visión general de los patrones que sigue.

Chapter 2

Bibliografia

Chapter 3

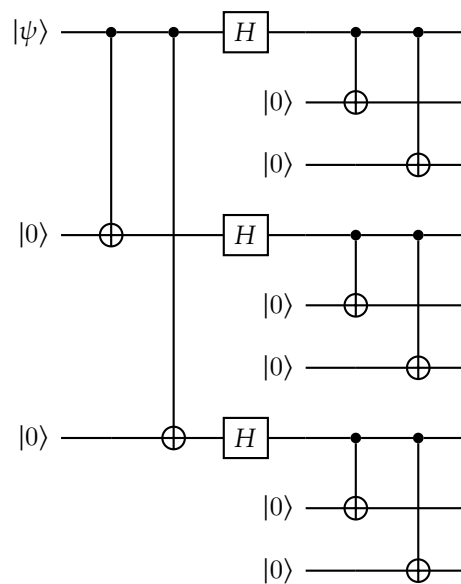
Annexos

3.1 Temas faltantes

3.2 Demostración de las condiciones de Corrección

3.3 Demostración de las condiciones de Corrección

3.4 Circuito de Shor



3.5 Shor Corrige errores arbitrarios

3.6 Shor Corrige errores arbitrarios