Universidad de Oviedo

Complejidad en computación cuántica

David Álvarez González

Supervisado por: Daniele Musso

UNIVERSIDAD DE OVIEDO

Facultad de Ciencias

Grado en Física

Junio de 2023

Índice general

Capítulo 1

Introducción

1.1. Motivación

En los últimos años, la computación cuántica ha emergido como un área de investigación apasionante y prometedora en la ciencia de la computación. A diferencia de la computación clásica, que utiliza bits clásicos para representar y manipular información, la computación cuántica se basa en los principios de la mecánica cuántica para aprovechar las propiedades únicas de los qubits cuánticos. Estos qubits, a diferencia de los bits clásicos, pueden existir en estados de superposición y entrelazamiento, lo que abre la puerta a nuevas formas de procesamiento y cálculo.

La motivación detrás de este trabajo radica en comprender la complejidad en la computación cuántica. La complejidad computacional es un área fundamental en la teoría de la computación que se centra en la cantidad de recursos necesarios para resolver problemas computacionales específicos. En el contexto de la computación cuántica, es crucial comprender cómo la introducción de los principios cuánticos afecta la complejidad de los algoritmos y los problemas que pueden ser resueltos eficientemente en esta nueva forma de computación.

1.2. Objetivos

Los objetivos principales de este trabajo son:

- Explorar los fundamentos de la mecánica cuántica relevantes para la computación cuántica, incluyendo los operadores en espacios de Hilbert, los estados entrelazados y el teorema de no clonación.
- Comprender los conceptos básicos de la computación clásica y la complejidad computacional, incluyendo los circuitos, la computación reversible y las clases de complejidad.
- Estudiar el modelo cuántico de circuitos y las puertas cuánticas, así como las diferencias entre la computación cuántica y probabilística.
- Analizar la complejidad cuántica y explorar el modelo de cajas negras, la distinguibilidad de estados y los límites inferiores para problemas cuánticos, centrándonos en el problema de búsqueda y el algoritmo cuántico de Grover.
- Investigar el problema del subgrupo oculto y su complejidad en términos de circuitos y estados cuánticos.
- Explorar las sombras clásicas en el contexto de la computación cuántica y entender sus aplicaciones a la hora de predecir funciones lineales.

1.3. Estructura del trabajo

En base a los objetivos marcados la estructura del trabajo será la siguiente:

En el capítulo 2 se introducirán los conceptos fundamentales de la mecánica cuántica que son relevantes para la computación cuántica.

En el capítulo 3 se explorarán los fundamentos de la computación clásica definiendo los circuitos y la computación reversible, así como las máquinas de Turing.

En el capítulo 4 se unirán los conceptos anteriores indagando en el área de la computación cuántica, presentando el modelo cuántico de circuitos, las puertas cuánticas. También se investigará sobre las diferencias entre la computación cuántica y la probabilística.

En el capítulo 5 se examinará el área de la complejidad cuántica, definiendo las cajas negras y ciertos algoritmos cuánticos.

En el capítulo 6 se estudiará el problema del subgrupo oculto y se introducirán los conceptos de complejidad de circuito y de estado, haciendo una breve comparación entre ambos.

En el capítulo 7 se introducirá el concepto de sombra clásica y se explorará su uso para predecir funciones lineales. Por último, en el capítulo 8 se discutirán las conclusiones principales del trabajo desarrollado.

Capítulo 2

Fundamentos de la mecánica cuántica

2.1. Dimensión finita

Los vectores toman una parte importante en la mécanica cuántica, al ser esta una teoría lineal. Un estado será representado matemáticamente como un vector en un espacio de vectores cuyas características se definirán y que tiene como nombre **espacio de estados**.

Observación 1. *[1] El estado de un sistema cuántico no se corresponde a un vector, sino a un rayo, un vector a menos de una fase global. Pero, gracias al teorema de Wigner, se puede trabajar con representaciones proyectivas a nivel de los vectores.*

Sea $\mathcal{H}$ un espacio de vectores de dimensión N sobre números complejos. A los elementos de $\mathcal{H}$ los denotaremos $|\phi\rangle, |\chi\rangle, ...$ Si $\lambda, \mu \in \mathbb{C}$ y $|\chi\rangle, |\phi\rangle \in \mathcal{H}$ la linealidad implica que $\lambda |\phi\rangle \equiv |\lambda\phi\rangle \in \mathcal{H}$ y $(\lambda |\phi\rangle + \mu |\chi\rangle) \in \mathcal{H}$. Este espacio es dotado de un producto escalar interno que lo hace *espacio de Hilbert*. Dados dos vectores $|\chi\rangle, |\phi\rangle$ su producto escalar se denota $\langle\chi|\phi\rangle$ y es lineal

en el segundo elemento:

$$\langle\chi|(\lambda\phi_1 + \mu\phi_2)\rangle = \lambda\langle\chi|\phi_1\rangle + \mu\langle\chi|\phi_2\rangle \tag{2.1}$$

Su complejo conjugado viene dado por:

$$\langle\chi|\phi\rangle = \langle\phi|\chi\rangle^* \tag{2.2}$$

lo que implica que $\langle\phi|\phi\rangle$ es un número real. De las ecuaciones 2.1 y 2.2 deducimos:

$$\langle\chi_1 + \lambda\chi_2|\phi\rangle = \langle\chi_1|\phi\rangle + \lambda^*\langle\chi_2|\phi\rangle \tag{2.3}$$

Finalmente, el producto escalar está definido positivamente:

$$\langle\phi|\phi\rangle = 0 \Leftrightarrow |\phi\rangle = 0 \tag{2.4}$$

2.1.1. Operadores en $\mathcal{H}$

Definición 2.1. *[1] Un **operador lineal** establece una correspondencia lineal $|\phi\rangle \to A|\phi\rangle := |A\phi\rangle$ tal que:*

$$A|\phi + \lambda\chi\rangle = A|\phi\rangle + \lambda A|\chi\rangle \tag{2.5}$$

Definición 2.2. *[1] Dados dos vectores $|\chi\rangle, |\phi\rangle$ se define el **hermítico conjugado** de un operador A, $A^\dagger$ como:*

$$\langle\chi|A^\dagger|\phi\rangle = \langle\phi|A|\chi\rangle^* \tag{2.6}$$

*Un operador A se dice **hermítico** si $A = A^\dagger$.*

Definición 2.3. *[1] Un operador que cumple que $U^\dagger = U^{-1}$ se dice unitario.*

Sea $\mathcal{H}_1$ un subespacio de $\mathcal{H}$ y $\mathcal{H}_2$ el subespacio ortogonal. Cualquier vector $|\phi\rangle$ puede descomponerse en $|\phi_1\rangle$ y $|\phi_2\rangle$ de manera que $|\phi_1\rangle \in \mathcal{H}_1$ y $|\phi_2\rangle \in \mathcal{H}_2$ y $\langle\phi_1|\phi_2\rangle = 0$. El proyector P_1 en $\mathcal{H}_1$ se define bajo la acción:

$$P_1|\phi\rangle = |\phi_1\rangle \tag{2.7}$$

Un operador proyección P cumple que es hermítico y $P^2 = P$

Teorema 2.1. *[1] Los valores propios de un operador hermítico son reales y los vectores propios correspondientes a dos valores propios diferentes son ortogonales.*

Dos operadores A, B conmutan si $AB = BA$ y en este caso, su conmutador, definido como:

$$[A, B] = AB - BA \tag{2.8}$$

se anula.

Teorema 2.2. *Dados dos operadores A y B que conmutam se puede encontrar una base de $\mathcal{H}$ construida con vectores propios comunes de A y de B.*

Esto quiere decir que ambos operadores A y B pueden ser diagonalizados "*a la vez*", es decir, una misma base hace diagonales a ambos. Un conjunto de operadores hermíticos $A_1, ..., A_M$ que conmutan dos a dos y cuyos valores propios definen los vectores de una base de $\mathcal{H}$ se dice *conjunto completo de operadores conmutantes* o $CCOC$.

Es útil comprender como construir una función $f(A)$ evaluada en un operador. Sea f una función definida por su serie de Taylor que converge en una cierta región del plano comlpejo $|z| < R$:

$$f(z) = \sum_{p=0}^{\infty} c_p z^p \tag{2.9}$$

Si el operador A es diagonalizable; $A = XDX^{-1}$ donde D es diagonal y sus elementos son d_n:

$$f(A) = \sum_{p=0}^{\infty} c_p A^p = \sum_{p=0}^{\infty} c_p X D^p X^{-1} = X \left[\sum_{p=0}^{\infty} c_p D^p \right] X^{-1} \tag{2.10}$$

Un caso particular de esta expresión es la exponencial de una matriz, $exp\, A$:

$$exp\, A = \sum_{p=0}^{\infty} \frac{A^p}{p!} \tag{2.11}$$

2.2. Postulados

Postulado 1 (El espacio de estados)**.** *[1] Las propiedades de un sistema cuántico están completamente determinadas por su vector estado, $|\phi>$, que fija la representación matemática del estado físico del sistema. Es conveniente que esté normalizado; $||\phi||^2 = \langle\phi|\phi\rangle = 1$. El vector es un elemento del espacio de Hilbert complejo $\mathcal{H}$.*

La linealidad de la teoría implica el principio de superposición: si $|\phi\rangle$ y $|\chi\rangle$ son dos vectores de $\mathcal{H}$, el estado

$$|\psi\rangle = \frac{\mu\,|\phi\rangle + \lambda\,|\psi\rangle}{||\mu\,|\phi\rangle + \lambda\,|\psi\rangle\,||} \qquad \mu, \lambda \in \mathbb{C}$$

es también un vector de $\mathcal{H}$ que representa un estado físico. [1]

Postulado 2 (Regla de Born)**.** *[1] Si $|\phi\rangle$ es un vector representando un sistema y $|\chi\rangle$ es otro vector representando a otro sistema, existe la amplitud de probabilidad de encontrar $|\phi\rangle$ en el estado $|\chi\rangle$ y viene dada por su producto escalar, $a(\phi \to \chi) = \langle\chi|\phi\rangle$. La probabilidad viene dada por su módulo al cuadrado:*

$$p(\phi \to \chi) = |a(\phi \to \chi)|^2 = |\,\langle\chi|\phi\rangle\,|^2 \tag{2.12}$$

Postulado 3 (Propiedades físicas y operadores)**.** *[1] Para cada propiedad física (observable) $\mathcal{A}$ existe un operador hermítico A que actúa sobre el espacio de estados $\mathcal{H}$; A es la representación matemática de $\mathcal{A}$*

Postulado 4 (Colapso de la función de onda)**.** *[1] Si un sistema es inicialmente en el estado $|\phi\rangle$ y si el resultado de una medida ideal $\mathcal{A}$ es a_n, inmediatamente después de la medida el sistema está en el estado proyectado en el subespacio de valor propio a_n:*

$$|\phi\rangle \to |\psi\rangle = \frac{P_n\,|\phi\rangle}{(\langle\phi|\,P_n\,|\phi\rangle)^{1/2}} \tag{2.13}$$

Si dos observables $\mathcal{A}$ y $\mathcal{B}$ son compatibles, el orden de la medida no es relevante. Si son compatibles, los operadores asociados conmutan, $AB =$

BA y en este caso se puede encontrar una base de estados $|\phi_n\rangle$ que sean vectores propios de ambos operadores, por lo que $A(B|\phi_n\rangle) = A(b_n|\phi_n\rangle) = b_n(A|\phi_n\rangle) = a_n b_n|\phi_n\rangle$ y $B(A|\phi_n\rangle) = B(a_n|\phi_n\rangle) = a_n(B|\phi_n\rangle) = a_n b_n|\phi_n\rangle$. Ahora bien, si las medidas fisicas son incompatibles, el conmutador $[A, B] \neq 0$. Si la medida de A ha dado el valor a y ha proyectado el sistema al estado $|a\rangle\,; A|a\rangle = a|a\rangle$ e inmediatamente después se hace la medida de $\mathcal{B}$, el vector $|a\rangle$ no siempre será vector propio de B y solo se podrá saber el resultado de la medida con cierta probabilidad. Es conveniente ahora definir las dispersiones y los valores esperados de un operador A en el estado $|\phi\rangle$, respectivamente $\Delta_\phi A$ y $\langle A\rangle_\phi$: [1]

$$\langle A\rangle_\phi = \langle\phi|\, A\,|\phi\rangle \tag{2.14}$$

$$(\Delta_\phi A)^2 = \left\langle A^2\right\rangle_\phi - (\langle A\rangle_\phi)^2 \tag{2.15}$$

Con estas magnitudes, se pueden escribir las inecuaciones de Heisenberg:

$$(\Delta_\phi A)(\Delta_\phi B) \geq \frac{1}{2}|\,\langle [A, B]\rangle_\phi\,| \tag{2.16}$$

Postulado 5 (Evolución temporal)**.** *[1] La evolución temporal de un estado $|\phi(t)\rangle$ viene dada por la ecuación:*

$$i\hbar\frac{d\,|\phi(t)\rangle}{dt} = H(t)\,|\phi(t)\rangle \tag{2.17}$$

El operador hermítico $H(t)$ es llamado *Hamiltoniano*

Postulado 6 (El operador evolución)**.** *[1] El estado $|\phi(t)\rangle$ en el instante t se puede obtener a partir del estado $|\phi(t_0)\rangle$ en el instante t_0 aplicándole el operador unitario $U(t, t_0)$, llamado el operador evolución:*

$$|\phi(t)\rangle = U(t, t_0)\,|\phi(t_0)\rangle \tag{2.18}$$

De hecho los postulados 5 y 6 no son independientes. Derivando (2.18) con respecto al tiempo y comparando con (2.17) se obtiene: [1]

$$i\hbar\frac{d}{dt}U(t, t_0) = H(t)U(t, t_0) \tag{2.19}$$

2.3. Estados entrelazados

Para construir un espacio de dos sistemas cuáticos que son independientes (cada uno en su espacio de Hilbert) se utiliza el producto tensorial. Dados dos espacios de Hilbert, $\mathcal{H}_1^N$ y $\mathcal{H}_2^M$ de dimensión N y M respectivamente, y dados dos estados $|\phi\rangle \in \mathcal{H}_1^N$ y $|\chi\rangle \in \mathcal{H}_2^M$, se define el estado $|\phi\rangle \otimes |\chi\rangle$ perteneciente al espacio de Hilbert $\mathcal{H}_1^N \otimes \mathcal{H}_2^M$. Este espacio es un espacio de Hilbert de dimensión NM. Descomponiendo cada estado en la base de cada espacio, respectivamente $|n\rangle$ y $|m\rangle$:

$$|\phi\rangle = \sum_{n=1}^{N} c_n |n\rangle \qquad |\chi\rangle = \sum_{m=1}^{M} b_m |m\rangle \tag{2.20}$$

el producto tensorial sería:

$$|\Phi\rangle = |\phi\rangle \otimes |\chi\rangle = \sum_{n,m} c_n b_m |n\rangle \otimes |m\rangle \tag{2.21}$$

$|\Phi\rangle$ se dice un estado separable, entanto que se puede escribir de la forma $|\phi\rangle \otimes |\chi\rangle$, pero no todos los vectores de $\mathcal{H}_1^N \otimes \mathcal{H}_2^M$ se pueden descomponer. En el caso en el que un estado no se deje descomponer se dice entrelazado. De todas formas, un estado entrelazado siempre se puede escribir como combinación lineal de estados de la base separable, es decir, si $|\xi\rangle$ es un estado entrelazado, $|\xi\rangle = \sum_{n,m} a_{n,m} |n\rangle \otimes |m\rangle$. Un operador $C = A \otimes B$, con A y B actuando sobre $\mathcal{H}_1^N$ y $\mathcal{H}_2^M$ respectivamente actúa sobre un producto tensorial de la siguiente manera:

$$(A \otimes B) |\phi\rangle \otimes |\chi\rangle = A |\phi\rangle \otimes B |\chi\rangle \tag{2.22}$$

aunque un operador C actuando sobre $\mathcal{H}_1^N \otimes \mathcal{H}_2^M$ no siempre se prodrá escribir de la forma antes mencionada. Sin embargo, como el estado genérico se deja escribir como suma de estados factorizables y los operadores son lineales, es suficiente conocer la (2.22) para conocer la acción sobre un estado genérico.

2.3.1. Matriz de densidad

Dado un sistema de dos partículas definido por un estado $|\Phi\rangle$ separable, $|\Phi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle$, se puede decir que la partícula 1 se define mediante el estado $|\phi_1\rangle$ pero, ¿qué sucede si el estado es entrelazado? En general, no se puede afirmar que la partícula 1 venga dada por un estado del subespacio de Hilbert correspondiente. Cuando una partícula (o sistema cuático) puede ser definida mediante un estado, como se ha desarrollado hasta ahora, se dice que es un estado puro. Cuando por el contrario la información sobre el sistema es incompleta (no se puede definir el sistema mediante un estado) se dice que es un estado mixto, y el sistema viene definido por el operador densidad u operador estado. Un estado mixto nace del entrelazamiento de dos subsistemas. Cuando se quiere medir un subsistema entrelazado con otro, del cual no se tiene información (no se puede medir), se recurre a la probabilidad. Sea $|\Phi\rangle$ un estado entrelazado, $|\Phi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle$. Si $|\phi_2\rangle$ no puede ser medido, se puede trabajar con los posibles estados de $|\phi_2\rangle$:[1]

$$\sum_k {}_k\langle\phi_2|\Phi\rangle = \sum_{i,j,k} c_{i,j} |\phi_1\rangle_j {}_k\langle\phi_2|\phi_2\rangle_i = \sum_{i,j} c_{i,j} |\phi_1\rangle_j p_i = \sum_i p_i \sum_j c_{i,j} |\phi_1\rangle_j \tag{2.23}$$

con

$${}_j\langle\phi_2|\phi_2\rangle_i = \delta_{i,j} p_i \tag{2.24}$$

En un estado mixto solo se sabe la probabilidad p_i ($0 \leq p_i \leq 1$) de que el sistema esté en el estado $|\phi_i\rangle$. Se asume que los estados $|\phi_i\rangle$ están normalizados pero no tienen que ser ortogonales. Por definición, el operador densidad es: [1]

$$\rho = \sum_i p_i |\phi_i\rangle \langle\phi_i| = \sum_i p_i P_{\phi_i} \tag{2.25}$$

El valor esperado de un operador A en el estado $|\phi_i\rangle$ es:

$$\langle A\rangle_i = \langle\phi_i| A |\phi_i\rangle$$

y está asociado a la probabilidad p_i, por lo que el valor esperado global de A, $\langle A \rangle$ es: [1]

$$\langle A \rangle = \sum_i p_i \langle A \rangle_i = \sum_i p_i \langle \phi_i | A | \phi_i \rangle = Tr(\rho A) \tag{2.26}$$

Propiedades 2.3. *[1] Las propiedades del operador densidad que se siguen de la definición son:*

- *ρ es hermítico; $\rho^\dagger = \rho$*
- *ρ tiene traza unidad; $Tr(\rho) = 1$*
- *ρ es un operador positivo; $\langle \phi | \rho | \phi \rangle \geq 0$ para cualquier $|\phi\rangle$*
- *Una condición necesaria y suficiente para que ρ describa un estado puro es $\rho = \rho^2$. De hecho, como $\rho^\dagger = \rho$, la condición $\rho = \rho^2$ implica que ρ sea un proyector.*

De igual manera, un operador cuya traza es 1 y es positivo puede ser interpretado como un operador densidad.

Una pregunta natural que surje de la definición es, dado un operador de densidad en un espacio de Hilbert $\mathcal{H}_1 \otimes \mathcal{H}_2$ de dos sistemas cuánticos, ¿cuál es el operador densidad que identifica la partícula 1? Esto es lo que se conoce como "operador de densidad reducida". Se considera un operador C, que actúa solamente sobre la particula 1, es decir $A = C \otimes I_2$. El objetivo es encontrar un operador densidad, $\rho^{(1)}$ de manera que:

$$\langle A \rangle = Tr(\rho^{(1)} A) \tag{2.27}$$

Si los subíndices $_1$ recorren los vectores de la base asociados a la primera partícula y los subíndices $_2$ lo hacen para la segunda, se puede escribir el valor esperado de $\langle A \otimes I_2 \rangle$ en el espacio $\mathcal{H}_1 \otimes \mathcal{H}_2$ como sigue:

$$\langle A \otimes I_2 \rangle = Tr([A \otimes I_2]\rho) = \sum_{n_1,m_1;n_2,m_2} A_{n_1,m_1} \delta_{n_2,m_2} \rho_{n_1,m_1;n_2,m_2} =$$

$$= \sum_{n1,m_1} A_{n_1,m_1} \sum_{n_2} \rho_{n_1,m_1;n_2,n_2} = \sum_{n_1,m_1} A_{n_1,m_1} \rho^{(1)}_{n_1,m_1} = Tr(A\rho^{(1)}) \tag{2.28}$$

Por lo tanto, el operador densidad de la partícula 1 $\rho^{(1)}$ viene dado en la base $|n_1\rangle$ del subespacio $\mathcal{H}_1$ por:

$$\rho^{(1)}_{n_1,m_1} = \sum_{n_2} \rho_{n_1,m_1;n_2,n_2} \qquad o \qquad \rho^{(1)} = Tr_2(\rho) \tag{2.29}$$

donde Tr_2 representa la traza en el subespacio $\mathcal{H}_2$

Inicialmente se han enunciado los postulados para sistemas cerrados, sin interacción de sistemas externos, que se describían con estados puros. Si se permite que haya interacción con otro sistema, a menudo es conveniente utilizar estados mixtos para describirlos y por tanto las matrices de densidad. A las operaciones con estas matrices se les llama superoperadores. [2]

Un superoperador puede tomar como entrada un sistema con matriz de densidad ρ_{in} correspondiente a un espacio de Hilbert de dimensión N, añadir un estado auxiliar de dimensión arbitraria, hacer actuar una operación unitaria sobre el sistema conjunto y descartar algun subsistema. [2]

Explícitamente sería:

$$\rho_{in} \mapsto \rho_{out} = Tr_2(U(\rho_{in} \otimes |00...0\rangle \langle 00...0| U^\dagger)) \tag{2.30}$$

donde el estado $|00...0\rangle$ es un estado auxiliar de dimensión arbitraria,U es un operador unitario actuando sobre el sistema conjunto y $_2$ es el subespacio del sistema auxiliar.

2.4. Teorema de no clonación.

Éste es un teorema básico al que se hará referencia durante el trabajo

Teorema 2.4 (Teorema de no Clonación)**.** *[3] No existe procedimiento unitario tal que* $|\psi\rangle \longrightarrow |\psi\rangle^{\otimes 2}$ *donde* $|\psi\rangle^{\otimes 2}$ *es una abreviación de* $|\psi\rangle \otimes |\psi\rangle$

Demostración. [3] Sea una transformación que mapea $(\alpha|0\rangle + \beta|1\rangle)|0\rangle$ a $(\alpha|0\rangle + \beta|1\rangle)(\alpha|0\rangle + \beta|1\rangle)$. Esta expresión es igual a $\alpha^2|00\rangle + \alpha\beta|01\rangle + \alpha\beta|10\rangle + \beta^2|11\rangle$, lo que es una transformación no lineal y, por tanto, no puede ser unitaria □

El teorema ha sido enunciado para transformaciones unitarias. Realmente esta asunción no es necesaria y se puede demostrar utilizando superoperadores.

Capítulo 3

Computación y complejidad

3.1. Circuitos

Los circuitos son redes compuestas de cables que transportan bits hacia puertas, que hacen operaciones elementales con estos. Los circuitos utilizados son circuitos acíclicos, es decir, los bits circulan por el circuito de manera lineal y los cables nunca se conectan con una localización anterior. En cada instante t se puede colocar a lo sumo una puerta en cada cable. Se da un ejemplo gráfico en la Figura 3.1. Notar que, pese a estar así configurada, podría haber puertas en paralelo de dos bits al mismo instante, una en los cables i y j y otra en los cables k y l. [2]

Es conveniente mostrar que lo único que necesitamos para construir un circuito que pueda efectuar cualquier computación que queramos es un conjunto finito de puertas. Esto viene ilustrado en la siguiente definición:

Definición 3.1. *[2] Un conjunto de puertas es* ***universal*** *si, para cualesquiera enteros positivos n,m y una función Booleana* $f : \{0,1\}^n \longrightarrow \{0,1\}^m$*, se puede construir un circuito para computar f con solo puertas de ese conjunto.*

Un ejemplo de conjunto universal en computación cuática es la puerta

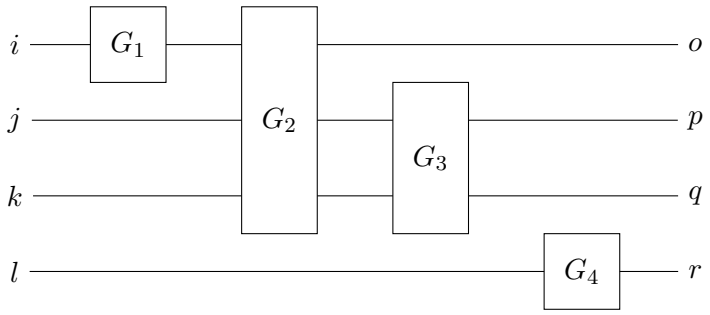

Figura 3.1.1: Esquema de circuito, donde las líneas horizontales son los cables, y los rectágulos G_1, G_2, G_3, G_4 son las puertas. Los inputs i, j, k, l son los bits de entrada, que están escritos en la parte izquierda de los cables. Tras pasar por las puertas se convierten en los outputs o, p, q, r, escritos en la parte derecha.

de Toffoli. La puerta de Toffoli es una puerta reversible de 3 bits que tiene el efecto de cambiar el tercer bit si y solo si los otros dos bits están en el estado 1. El conjunto formado por únicamente esta puerta es universal en computación cuántica. En computación clásica, un conjunto de puertas universal es el $NAND$, es decir, el AND negado.

A raíz de estos circuitos, se puede elaborar un modelo de circuitos probabilísticos, añadiendo una puerta 'lanza-monedas' que actúe solo sobre un bit y le asigne un valor aleatorio y binario, independientemente del bit de entrada. Es decir, un circuito general probabilístico se puede elaborar con un conjunto de puertas universal determinista junto con la puerta lanza-monedas.

3.2. Computación reversible

Una computación es reversible si es posible recuperar el input unívocamente dado el output. Por ejemplo, la puerta NOT es reversible, si el output es 0 es sabido que el input es 1 y viceversa. Sin embargo, la puerta AND

no lo es, si el output es 0, los inputs pueden ser 00, 01 o 10. Ahora bien, cualquier computación (generalmente irreversible) puede transformarse en una computación reversible. Es fácil de ver con el modelo de circuitos. Cada puerta de una familia finita puede hacerse reversible añadiendo cables de entrada (inputs) y de salida (outputs). Por ejemplo, la puerta AND puede hacerse reversible añadiendo un input y dos outputs.

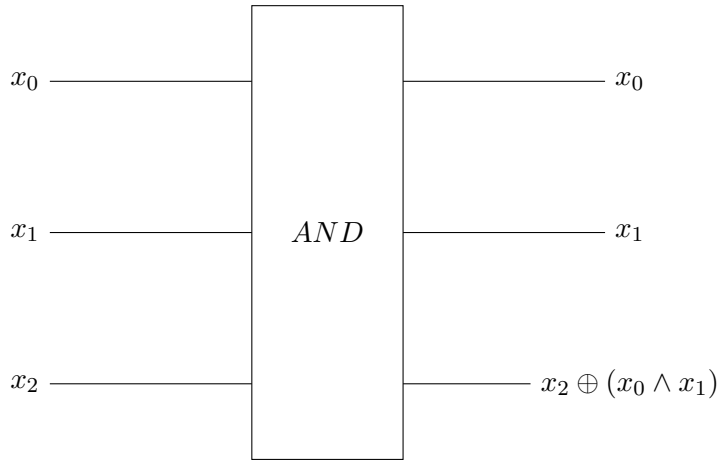

Figura 3.2.1: La puerta reversible AND mantiene una copia de los bits x_0 y x_1, y añade un AND de x_0 y x_1 (denotado $x_0 \wedge x_1$) al input adicional. Notar que, manteniendo el bit adicional a 0 y descartando los bits de salida x_0 y x_1 se tiene la puerta AND no reversible

La información adicional 'basura' puede ser borrada al final de la computación, haciendo una copia de los bits de salida y luego corriendo el circuito inverso reversible para obtener los estados iniciales de nuevo. Esta copia tiene que ser hecha de manera reversible, lo que significa que no vale con solo sobreesscribir los bits de salida en los bits de copia. Esta copia reversible puede hacerse con una secuencia de puertas $CNOT$. Un ejemplo de esto se ve en la Figura 3.2.2.

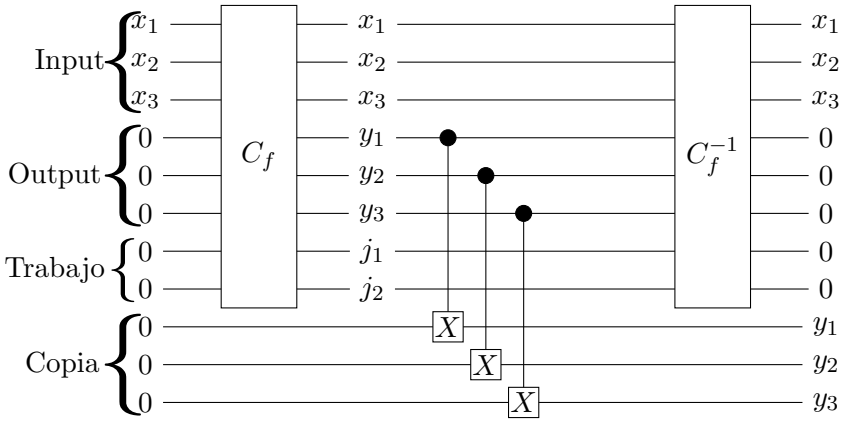

Figura 3.2.2:

3.3. Complejidad

Uno de los intereses de la computación es investigar el número de recursos que un ordenador emplea, lo que se llama comlpejidad. Un importante recurso es el tiempo que tarda el ordenador en ejecutar la tarea dada. Por ejemplo, si se necesita multiplicar dos números de n bits, un ordenador empleará un polinomio de orden dos de tiempo (que pueden ser segundos o el tiempo que tarda un ordenador en ejecutar un paso básico). Cabe destacar que la complejidad depende del ordenador utilizado. Otro ordenador puede ejecutar el mismo algoritmo en un polinomio de orden 3 de tiempo. [2]

Esto puede resultar un problema a la hora de estudiar un algoritmo, por lo que, con el fin de evitarlo, se hacen medidas más amplias. Una de ellas es considerar solamente el término de mayor exponente y eliminar el factor que lo multiplica. De esta manera, se diría que los tiempos anteriores son $\mathcal{O}(n^2)$ y $\mathcal{O}(n^3)$ respectivamente. Esta notación no solo se puede utilizar con polinomios, si no con cualquier función de n. $\mathcal{O}(f(n))$ supone una cota superior en el tiempo de ejecución de un algoritmo, es decir, si un algoritmo se ejecuta en una comlpejidad $\mathcal{O}(n^3)$, también lo hará en $\mathcal{O}(e^n)$. Para referirse

a cotas inferiores se utiliza la notación Ω. Un algoritmo se cosidera eficiente si se ejecuta en un tiempo $\mathcal{O}(n^k)$ para algún k, y se les llama polinómicos. Si un algoritmo se ejecuta en un tiempo $\Omega(c^n)$ para algún c se dice exponencial y se considera que no es eficiente. Si un algoritmo tiene el mismo límite superior o inferior se utiliza Θ. [2]

3.4. Máquinas de Turing

La tesis de Church-Turing dice que cualquier problema de computación puede ser resuelto en cualquier dispositivo que pueda ser creado si y solo si puede ser resuelto en una máquina simple, llamada la máquina de Turing. Esta máquina es una abstracción matemática y no un dispositivo físico. Una máquina de Turing es un modelo de computación que consiste en un conjunto finito de estados $q_1, ..., q_n$, una $'cinta'$ unidimensional infinita en una dirección dividida en cuadrados que pueden contener un único símbolo procedente de un alfabeto finito. En cada instante la máquina lee un cuadrado que puede no tener símbolo (se denota como S_0) o contener un símbolo $S_1, ..., S_m$ con $S_1 = 0$ y $S_2 = 1$.

Una máquina de Turing es una máquina automática, lo que significa que el comportamiento de la máquina viene totalmente determinado por el símbolo y el estado en cada instante. Puede tener tres acciones.

- Imprimir S_i, moverse hacia la izquierda un cuadrado (I) e ir al estado q_j.
- Imprimir S_i, moverse hacia la derecha un cuadrado (D) e ir al estado q_j.
- Imprimir S_i, no moverse (N) e ir al estado q_j.

Los movimientos de una máquina de Turing pueden ser descritos por la

quíntupla:

$$q_i S_j S_{i,j} M_i q_j \tag{3.1}$$

donde q_i es el estado actual, S_j el contenido del cuadrado leído, $S_{i,j}$ el nuevo contenido del cuadrado, M_j el movimiento y $q_{i,j}$ el siguiente estado de la máquina.

La tesis de Church-Turing no dice nada sobre la eficiencia de computación. Cuando un ordenador simula a otro, normalmente existe un coste adicional procedente de la simulación. Por ejemplo, consideramos dos ordenadores A y B y supongamos que C requiere T unidades de tiempo y la simulacíon de A usa $\mathcal{O}(T^2 2^T)$. Si C puede resolver un problema en $\mathcal{O}(n)$, A usará hasta $\mathcal{O}(n^2 2^n)$.

Con el fin de ampliar la tesis de Church-Turing es útil generalizar la defición ligeramente. Una máquina de Turing probabilística es una capaz de hacer una elección aleatoria binaria en cada paso, como si se tratase de lanzar una moneda al aire, donde las reglas de transición de estados se expanden para tener en cuenta estos bits. Existen problemas que pueden ser resueltos mediante una máquina de Turing probabilística pero no con una convencional, como el problema de encontar raíces cuadradas modulo un primo. El algoritmo para este problema sugiere probar si un $t \in \mathbb{F}_p$ aleatorio cumple determinadas condiciones, por lo que es en esa aleatoriedad donde es necesaria la computación probabilística ($\mathbb{F}_p$ es un campo finito con p elementos). [4]Problemas como estos, en los que no se ha encontrado un algoritmo convencional pero si probabilístico, sugieren que las máquinas de Turing probabilísticas tienen más potencia (aunque sigue siendo una pregunta abierta) por lo que existe otra tesis relacionada con esto. [2]

Tesis de Church-Turing fuerte: Una máquina de Turing probabilísitica puede simular cualquier modelo de computación realista. [2]

El problema fundamental de la tesis de Church-Turing fuerte es que aparentemente la física clásica no es suficientemente poderosa como para

simular la física cuántica de manera eficiente. El pricipio básico aún se cree verdadero, pero hay que cambiar el modelo de computación a uno que pueda simular dispositivos cuánticos eficientemente. Esto deriva en la tesis de Church-Turing cuántica fuerte. Más adelante se definirá la máquina de Turing cuántica.

Tesis de Church-Turing cuántica fuerte: Una máquina de Turing cuántica puede simular cualquier modelo de computación realista.[2]

3.5. Problemas de reconocimiento del lenguaje

Una clase de problemas básica son los problemas de 'decisión', es decir, problemas en los cuales la respuesta es 'sí' o 'no'. Muchos de los problemas interesantes pueden ser reformulados para pertenecer a esta clase. Por ejemplo, el problema de factorizar un número entero N en dos factores no tiviales puede ser reducido a un problema de decisión cambiando la pregunta a '¿Tiene el entero N un factor no trivial más pequeño que T?' donde T es una entrada adicional elegida. Estos problemas pueden ser tratados como problemas de reconocimiento de un lenguaje. [2] Con el fin de computar, es necesaria una manera de representar la información. La codificación unaria (representar un número n como una cadena de 1s de longitud n) es una manera mucho menos eficiente que usar símbolos de un alfabeto de tamaño dos, y pasar de esta a un alfabeto más grande solo aumenta el tamaño de la representación una cantidad constante. Por lo tanto se usa el alfabeto $\Sigma = \{0, 1\}$. Σ^* representa todas las cadenas finitas a partir de este alfabeto. Un lenguaje L es un subconjunto finito de Σ^*. Un algoritmo "resuelve el problema de reconocimiento de lenguaje para L"si acepta cualquier cadena $x \in L$ y rechaza las cadenas $y \notin L$. [2]

Como ejemplo, se considera el problema de decidir si un grafo dado es de 3 colores. Un grafo se considera de 3 colores si se puede asignar a cada vértice v un color $c(v) \in \{VERDE, ROJO, AZUL\}$ de manera

que los vértices unidos a otros por una arista no son del mismo color y se llama grafo 3-COLOREABLE. Por ejemplo, sea un grafo no orientado (grafo en que las relaciones entre vértices son simétricas, sin sentido) de 4 vértices, v_1, v_2, v_3, v_4. Este solo puede tener 6 aristas, las cuales son: $e_1 = \{v_1, v_2\}, e_2 = \{v_1, v_3\}, e_3 = \{v_1, v_4\}, e_4 = \{v_2, v_3\}, e_5 = \{v_2, v_4\}, e_6 = \{v_3, v_4\}$. Se puede representar un grafo tal con una cadena $x_1x_2x_3x_4x_5x_6$ de longitud 6 haciendo que $x_j = 1$ si el grafo contiene la arista e_j y 0 de lo contrario. Así, el grafo 101111 es de 3 colores puesto que $c(v_1) = VERDE, c(v_2) = ROJO, c(v_3) = AZUL, c(v_4) = VERDE$, como se ve en la Figura 3.5.1

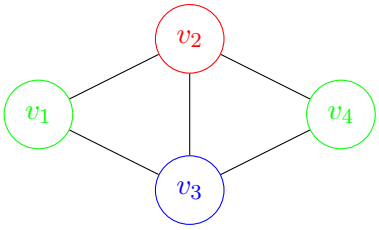

Figura 3.5.1: Este grafo está representado por la cadena 101111 y es de 3 colores.

Una vez definidos los lenguajes, se puede fraccionar los distintos problemas por complejidad.

La clase BPP (tiempo polinomial probabilístico con error acotado, 'bounded-error probabilistic polynomial time') consiste en los lenguajes L para los cuales existe un algoritmo clásico aleatorio A que se ejecuta en un tiempo, en el peor de los casos, polinomial tal que, para cada entrada $x \in \Sigma^*$

- si $x \in L$ la probabilidad de que A admita x es al menos $\frac{2}{3}$.
- si $x \notin L$ la probabilidad de que A admita x es como máximo $\frac{1}{3}$.

No hay nada de especial en el número $\frac{2}{3}$, cualquier constante $\frac{1}{2} + \delta$ con $\delta > 0$ valdría.

La clase BPQ (tiempo polinomial cuántico con error acotado, 'bounded-error quantum polynomial time') consiste en los lenguajes L para los cuales existe un algoritmo cuántico A que se ejecuta en un tiempo, en el peor de los casos, polinomial tal que, para cada entrada $x \in \Sigma^*$

- si $x \in L$ la probabilidad de que A admita x es al menos $\frac{2}{3}$.
- si $x \notin L$ la probabilidad de que A admita x es como máximo $\frac{1}{3}$.

Los algoritmos polinomiales son interesantes no solo porque son rápidos de ejecutar, si no porque la suma, producto y composición de polinomios sigue siendo uno. La mayoría de los cambios razonables en la implementación no depende de si un algoritmo es polinomial o no, por lo que distinguir si un algoritmo es polinomial o no, no dependerá de la implementación. Por ello, es conveniente tratar los algoritmos con complejidad polinomial como 'eficientes'. [2]

Algunos problemas pueden ser no eficientes a la hora de resolverlos pero eficientes a la hora de comprobarlos. El problema de decidir si un grafo es de tres colores es difícil, mientras que comprobar si una coloración de un grafo es de 3 colores es más sencillo. Es decir, existe un algoritmo polinómico, $COMPROBACIÓN\text{-}3\text{-}COLOREABLE(a, b)$ tal que $COMPROBACIÓN\text{-}3\text{-}COLOREABLE(x, y) = 1$ si y sólo si la coloración y es válida en el grafo x.

Esta propiedad inspira a la clase NP (tiempo polinomial no determinista, 'non-deterministic polynomial time') que consiste en los lenguajes L para los cuales existe un algoritmo de tiempo polinomial $A(a, b)$ tal que, para cada entrada $x \in \Sigma^*$

- Si $x \in L$ entonces existe una entrada y tal que $A(x, y)$ da el output 'aceptar'.
- Si $x \notin L$ entonces $A(x, y)$ da el output 'rechazar' para todo y.

y la longitud de y está acotado por un polinomio en la longitud de x.

Dentro de la clase NP hay una subclase interesante, llamada *NP-completa*. Si se es capaz de resolver un problema en esta subclase eficientemente, implica que se puede resolver eficientemente cualquier problema de la clase NP. Es decir, sea $L \in$ *NP-completa*. Para cualquier $L' \in NP$ existe un algoritmo clásico determinista de tiempo polinómico que computa una función $f : \{0,1\}^* \longrightarrow \{0,1\}^*$ de manera que $x \in L'$ si y sólo si $f(x) \in L$.

Dentro de esta clase *NP-completa* hay un problema interesante, que fue el primero encontrado dentro de esta clase, llamado el problema de satisfacibilidad booleana o 3-SAT. Un problema 3-SAT es especificado por una fórmula booleana Φ en una forma particular. Esta fórmula consiste en conjunciones (AND lógico) de términos, cada uno de los cuales es una disjunción (OR lógico) de tres variables booleanas (o su negación). Por ejemplo, una fórmula booleana en las variables $x_1, x_2, x_3, x_4, x_5, x_6$ cumpliendo esto sería:[2]

$$\Phi = (x_2 \vee \overline{x_3} \vee x_1) \wedge (x_6 \vee x_2 \vee \overline{x_1}) \wedge (\overline{x_4} \vee x_5 \vee \overline{x_6})$$

Un argumento satisfactorio de Φ es una serie de asignaciones de 0 y 1 a cada una de las variables tal que la fórmula se evalúa en 1. Intiutivamente, este problema está en *NP-completa* porque comprobar si cada asignación da como valor 1 se puede hacer en tiempo polinomial.

La clase $PSPACE$ consiste en los lenguajes L para los cuales existe un algoritmo clásico A que utiliza un espacio de memoria, en el peor de los casos, polinomial tal que, para cada entrada $x \in \Sigma^*$ el algoritmo acepta x si y solo si $x \in L$.

La Figura 3.5.2 muestra un diagrama con las relaciones conocidas de las clases de complejidad más conocidas. Claramente $P \subseteq NP$ y $P \subseteq BPP \subseteq BQP \subseteq PSPACE$ pero, hasta la fecha, aún no se ha demostrado que niguno de los contenidos sea estricto. Se cree que $P \neq NP$ y $NP \neq PSPACE$ y se

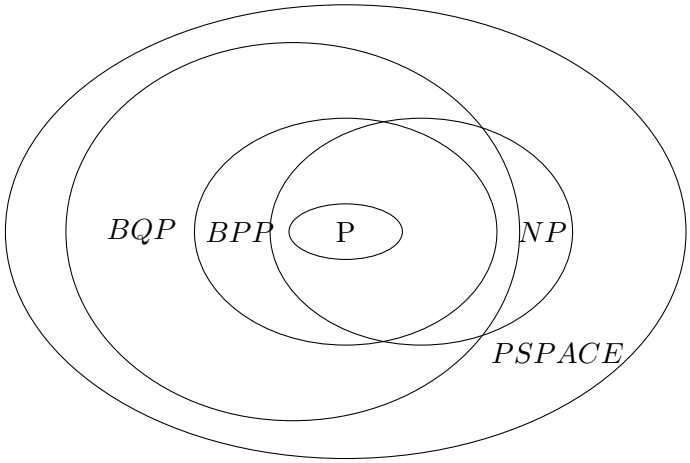

Figura 3.5.2: Diagrama que ilustra las relaciones conocidas entre las clases de complejidad más famosas.

espera que $BPQ \neq BPP$. Además, resolver el problema $P = NP$ es uno de los problemas abiertos más grandes en las matemáticas.

Capítulo 4

Computación cuática

El qubit es la unidad básica de la computación cuántica, así como lo es el bit de la computación clásica. Para formalizar la expresión del qubit es necesario un espacio de Hilbert, $\mathcal{H}$, de dimensión 2. Los vectores de la base son denotados como $\{|0\rangle, |1\rangle\}$ siendo evidentemente ortognonales. Un qubit es cualquier vector que se pueda escribir como combinación lineal compleja de los vectores de la base, es decir: [3]

$$|q\rangle = \alpha |0\rangle + \beta |1\rangle \qquad \alpha, \beta \in \mathbb{C} \tag{4.1}$$

Notar que una amplitud α puede descomponerse como un producto $e^{i\theta}|\alpha|$, donde $|\alpha|$ es la magnitud de α y θ es la fase. Un punto importante sobre los vectores es la fase global. Dado un vector $|\phi\rangle$ es equivalente el vector $e^{i\theta}|\phi\rangle$. Por ejemplo, el vector [2]

$$|0\rangle + |1\rangle$$

es equivalente a

$$e^{i\theta}|0\rangle + e^{i\theta}|1\rangle$$

Por otro lado, la fase relativa entre dos vectores ortogonales en superposición son físicamente significativas. El estado

$$|0\rangle + |1\rangle$$

es físicamente distinto al estado

$$e^{i\theta} |0\rangle + |1\rangle$$

Por lo tanto, la manera más genérica de describir un estado en un espacio de Hilbert de dimensión 2 es:

$$|\phi\rangle = cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\varphi} sen\left(\frac{\theta}{2}\right)|1\rangle \tag{4.2}$$

Este estado es a menudo representado como un punto sobre la superficie de la esfera de Bloch. Dos parámetros reales, θ y φ son suficientes para describir el estado, dado que está restringido a tener norma 1 y son equivalentes bajo cambio de fase global. Los puntos en la superficie de la esfera de Bloch pueden ser expresados en coodenadas cartesianas como el vector:

$$(x, y, z) = (sen\theta\, cos\varphi, sen\theta\, sen\varphi, cos\theta) \tag{4.3}$$

Un qubit también puede estar representado por un estado mixto y un punto en la superficie de la esfera de Bloch no sería adecuado para representarlo, si no que corresponden a puntos interiores. Si $\rho = \sum_i p_i |\phi_i\rangle \langle\phi_i|$ es el operador densidad y $|\phi_i\rangle$ tiene como vector de Bloch $(\alpha_{x,i}, \alpha_{y,i}, \alpha_{z,i})$, el vector de Bloch para el estado mixto ρ es:

$$\rho = \sum_i p_i(\alpha_{x,i}, \alpha_{y,i}, \alpha_{z,i}) = \left(\sum_i p_i\alpha_{x,i}, \sum_i p_i\alpha_{y,i}, \sum_i p_i\alpha_{z,i}\right) \tag{4.4}$$

Para representar la combinación de más de un qubit se utiliza el producto tensorial antes mencionado. Sean $\alpha|0\rangle + \beta|1\rangle$ y $\delta|0\rangle + \gamma|1\rangle$. El estado conjunto de estos dos qubits es:

$$(\alpha|0\rangle + \beta|1\rangle) \otimes (\delta|0\rangle + \gamma|1\rangle) =$$
$$= \alpha\delta(|0\rangle \otimes |0\rangle) + \alpha\gamma(|0\rangle \otimes |1\rangle) + \beta\delta(|1\rangle \otimes |0\rangle) + \beta\gamma(|1\rangle \otimes |1\rangle)$$

que pertenece al espacio de Hilbert $\mathcal{H} \otimes \mathcal{H}$. Para simplificar, se omite el producto tensorial y se meten ambos vectores en el mismo ket:

$$\alpha\delta(|0\rangle) + \alpha\gamma(|01\rangle) + \beta\delta(|10\rangle) + \beta\gamma(|11\rangle) \tag{4.5}$$

Este es un vector en un espacio de dimensión 4, generado por la base $\{|01\rangle, |01\rangle, |10\rangle, |11\rangle\}$. [3]

4.1. Modelo de álgebra lineal

En esta sección se desarrollará un modelo de matrices y vectores que, aunque no sea muy utilizado en la computación clásica, si que sirve de antecedente para la formulación estandar de la computación cuántica.

El estado en un punto dado de un circuito determinista (no probabilístico) puede ser especificado listando los valores de los bits en cada uno de los cables del circuito, el estado del cable será el valor del bit asociado a ese cable (0 o 1). Sin embargo, en un circuito probabilístico no es tan sencillo.

Sea un solo bit que tiene la probabilidad p_0 de estar en el estado 0 y p_1 de estar en el estado 1. Se puede sintetizar esta información en un vector de dos dimensiones:

$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \tag{4.6}$$

Una vez representados los cables, o bits, es conveniente representar las puertas lógicas que nos permitan operar con estos vectores. Sea la puerta NOT. El objetivo de esta puerta es modificar el estado del cable en cualquier situación. Nótese que el estado 0 puede representarse con probaiblidad $p_0 = 1$ y $p_1 = 0$ y análogamente el estado 1, por lo que se quiere:

$$NOT \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad NOT \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{4.7}$$

Lo que implica que se puede representar el operador NOT con la matriz:

$$NOT = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{4.8}$$

Por lo tanto:

$$NOT \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \tag{4.9}$$

Sea ahora un circuito compuesto por dos cables. El estado del primero en un punto dado es 0 con probabilidad $|\alpha_0|^2$ y 1 con probabilidad $|\alpha_1|^2$. El estado del segundo es 0 con probabilidad $|\beta_0|^2$ y 1 con probabilidad $|\beta_1|^2$. Las posibles combinaciones del estado combinado son $\{00, 01, 10, 11\}$, donde la cadena binaria ij indica que el primer cable esta en el estado i y el segundo cable está en el estado j. Así, las probabilidades vienen dadas por:

$$prob(ij) = |\alpha_i|^2 |\beta_j|^2 \tag{4.10}$$

Para representar el estado se hace uso del producto tensorial:

$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \otimes \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} = \begin{pmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_0 \\ \alpha_1\beta_1 \end{pmatrix} \tag{4.11}$$

También se puede representar puertas que actúan sobre dos bits, como la puerta $CNOT$. Esta puerta actúa sobre dos bits, etiquetados como control y objetivo. La acción de esta puerta es aplicar una puerta NOT sobre el bit objetivo en caso de que el bit control esté en el estado 0 y no haga nada en otro caso. La puerta $CNOT$ se puede representar como sigue:

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{4.12}$$

Es interesante ver como actúa si el primer bit está en el estado

$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \tag{4.13}$$

y el segundo en el estado

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{4.14}$$

El estado resultante al aplicar la puerta $CNOT$ será:

$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} \tag{4.15}$$

que es un estado entrelazado.

4.2. Modelo cuántico de circuitos

Previamente, en la Sección 3.1, se ha introducido la computación mediante circuitos. Este modelo puede ser generalizado para la computación cuántica. Aquí los 'cables' conducen los qubits y las puertas cuánticas actúan sobre los qubits. Una puerta cuántica de n qubits tiene de entrada n cables que transportan n qubits y tiene otros n cables con sus qubits de salida. Se representa esquemáticamente como un circuito convencional. Por conveniencia, se restringirá el desarrollo a puertas unitarias (que serán tambien reversibles). En la Figura 4.2.1 se muestra un ejemplo de circuito.

4.2.1. Puertas cuánticas

Previamente en este capítulo, se ha introducido la puerta NOT, también llamada la puerta de Pauli X, para un solo bit, siendo un operador unitario de dimensión 2. Una puerta cuántica U transforma el qubit $|\phi\rangle$ en el qubit $U|\phi\rangle$. En términos de la esfera de Bloch, la acción de U se puede ver como la rotación del vector de Bloch $|\phi\rangle$ al vector $U|\phi\rangle$. Se ha visto en la Sección 2.1.1 la función exponencial, entre otras. Si se aplica la función exponencial a los operadores de Pauli, se obtienen los operadores unitarios correspondientes a las rotaciones alrededor de los ejes x, y y z de la esfera de Bloch. Los

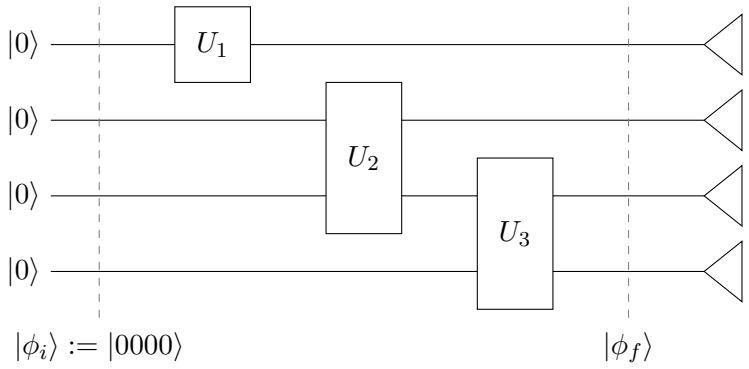

Figura 4.2.1: Un circuito cuántico. El estado de 4 qubits $|\phi_i\rangle := |0000\rangle$ entra en el circuito cuántico por la izquierda. Las cajas etiquetadas como U_1, U_2, U_3 representan las puertas cuánticas. El estado de 4 qubits (posiblemente entrelazado) después que las puertas sean aplicadas es $|\phi_f\rangle$. Los triángulos de la derecha representan las medidas de los qubits, en este caso, al haber un triángulo por cada qubit significa que son medidos por separado.

operadores de Pauli son: [2]

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (4.16)$$

Y las puertas de rotación estan definidas como sigue:

$$R_x(\theta) = e^{\frac{-i\theta X}{2}}; R_y(\theta) = e^{\frac{-i\theta Y}{2}}; R_z(\theta) = e^{\frac{-i\theta Z}{2}}; \quad (4.17)$$

Sea un estado de 1 qubit arbitrario definido por sus ángulos del vector de Bloch:

$$cos\left(\frac{\sigma}{2}\right)|0\rangle + e^{i\tau} sen\left(\frac{\sigma}{2}\right)|1\rangle$$

Aplicando el operador $R_z(\theta)$ se obtiene: [2]

$$cos\left(\frac{\sigma}{2}\right)|0\rangle + e^{i(\tau+\theta)} sen\left(\frac{\sigma}{2}\right)|1\rangle$$

salvo fase global. $R_z(\theta)$ ha cambiado el ángulo τ a $\tau + \theta$ lo que significa una rotación de θ sobre el eje z de la esfera de Bloch.

Teorema 4.1. *[2] Sea U una puerta unitaria que actúa sobre un solo bit. Existen números reales α, β, γ y δ tales que:*

$$U = e^{i\alpha} R_x(\beta) R_y(\gamma) R_z(\delta) \tag{4.18}$$

También se puede generalizar la descomposición de una puerta unitaria en función de dos ejes no-paralelos cualesquiera:

Teorema 4.2. *Sea U una puerta unitaria que actúa sobre un solo bit. Sean l y m dos ejes no paralelos de la esfera de Bloch. Existen números reales α, β, γ y δ tales que:*

$$U = e^{i\alpha} R_l(\beta) R_m(\gamma) R_l(\delta) \tag{4.19}$$

Proposición 4.3. *[2] Cualquier puerta unitaria U de un qubit puede escribirse de la siguiente manera:*

$$U = e^{i\alpha} AXBXC \tag{4.20}$$

donde A, B, C son operadores unitarios satisfaciendo $ABC = I$ y X es la puerta NOT.

Existen más puertas cuánticas conocidas. A continuación se dan más ejemplos:[3]

- La puerta *Toffoli*, de la que ya se ha hablado previamente, que mapea $|x, y, z\rangle$ a $|x, y, z \oplus xy\rangle$.
- La puerta de *fase*

$$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \tag{4.21}$$

- La puerta $CNOT$, también mencionada

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{4.22}$$

4.3. Puertas controladas

Dada una puerta U de 1 qubit, se puede definir la puerta $U-controlada$ de forma similar a como se ha hecho anteriormente. Se denota $c-U$ y se corresponde a una puerta de 2 quibits que actúa de la siguiente manera:

$$\begin{aligned} c-U\left|0\right\rangle\left|\phi\right\rangle &= \left|0\right\rangle\left|\phi\right\rangle \\ c-U\left|1\right\rangle\left|\phi\right\rangle &= \left|1\right\rangle U\left|\phi\right\rangle \end{aligned} \tag{4.23}$$

Esto puede generalizarse para cualquier circuito implementando una operación unitaria U. Sea C_U un circuito implementando una operación unitaria U. El objetivo es buscar un circuito para la operación $U-controlada$. La técnica básica es sustituir cada una de las pueras G del circuito por una puerta controlada $c-G$, todas al mismo qubit de control.

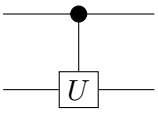

Figura 4.3.1: Puerta $c-U$

4.4. Conjuntos universales de puertas cuánticas

Hasta ahora se ha trabajado con puertas de solo 1 o 2 qubits. Un algoritmo interesante sería un operador unitario genérico actuando sobre n-qubits. El objetivo es elegir un número finito de puertas de manera que, al construir un circuito utilizando únicamente puertas de este conjunto, se puedan implementar operaciónes más complejas.

Cuando se usa un circuito de puertas cuánticas para implementar alguna operación unitaria deseada, en la práctica, es suficiente tener una implementación que aproxime la operación deseada con un nivel de precisión específico. Ahora bien, es necesario precisar la noción de *calidad de aproximación* de una transformación unitaria. Sea U la transformación unitaria deseada y

V la aproximación a esa transformación (también unitaria). El error en la aproximación es:

$$E(U,V) = \max_{|\phi\rangle} ||(U-V)\,|\phi\rangle\,|| \tag{4.24}$$

donde la norma es la norma euclídea $\||\phi\rangle\| = \sqrt{\langle\phi|\phi\rangle}$.Dada esta definición, cuando U puede ser aproximado bajo cierta precisión significa que, dada una tolerancia $\epsilon > 0$ se puede implementar V tal que $E(U,V) < \epsilon$.

Proposición 4.4. *Dados los operadores unitarios $U_1, ..., U_n, V_1, ..., V_n$ se cumple:*

$$E(U_nU_{n-1}...U_2U_1, V_nV_{n-1}...V_2V_1) = E(U_n,V_n)+E(U_{n-1},V_{n-1})+...+E(U_1,V_1) \tag{4.25}$$

Definición 4.1. *[2] Se dice que un conjunto de puertas es **universal** si para cualquier número entero $n \geq 1$, cualquier operador unitario de n-qubits puede aproximarse con una precisión arbitraria mediante un circuito cuántico usando solo puertas de ese conjunto.*

Definición 4.2. *[2] Se dice que una puerta de 2 qubits es una **puerta entrelazada** si para algún estado producto de entrada $|\phi\rangle\,|\psi\rangle$ la salida de la puerta no es un estado producto (es decir, los qubits de salida están entrelazados).*

Teorema 4.5. *[2] Un conjunto compuesto por cualquier puerta entrelazada de 2 qubits, junto con todas las puertas de 1 qubit, es universal.*

El Teorema 4.5 implica que la puerta $CNOT$ junto a todas las puertas de 1-qubit es un conjunto universal. Este teorema da conjuntos universales en un sentido más estricto que en la Definición 4.1, en tanto que se puede implementar cualquier operación unitaria de n-qubits de manera exacta. El problema de este teorema es que proporciona un conjunto infinito de puertas, y el objetivo era encontrar uno finito. Una manera de empezar es con las puertas de 1-qubit.

Definición 4.3. *[2] Se dice que un conjunto de puertas es* ***universal para puertas de 1-qubit*** *si cualquier puerta unitaria de 1-qubit se puede aproximar a una precisión arbitraria mediante un circuito cuántico utilizando solo puertas de ese conjunto.*

La puerta de Hadamard, H, se define de manera que actúa sobre la base computacional de la siguiente manera:

$$\begin{aligned} H|0\rangle &= \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ H|1\rangle &= \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned} \tag{4.26}$$

La matriz de Hadamard tiene la siguiente representación matricial con respecto a la base computacional:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{4.27}$$

Una propiedad de la matriz H es que es autoinversa, es decir, $H = H^{-1}$, por lo que

$$\begin{aligned} H(\tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle)) &= |0\rangle \\ H(\tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle)) &= |1\rangle \end{aligned} \tag{4.28}$$

Otra puerta de 1-qubit es la puerta de fase $\frac{\pi}{8}$, T, que actúa:

$$\begin{aligned} T|0\rangle &= |0\rangle \\ T|1\rangle &= e^{i\frac{\pi}{4}}|1\rangle \end{aligned} \tag{4.29}$$

que tiene la representación matricial:

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix} = \begin{pmatrix} e^{-i\frac{\pi}{8}} & 0 \\ 0 & e^{i\frac{\pi}{8}} \end{pmatrix} \tag{4.30}$$

O la puerta de fase S:

$$\begin{aligned} S|0\rangle &= |0\rangle \\ S|1\rangle &= i|1\rangle \end{aligned} \tag{4.31}$$

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \tag{4.32}$$

Teorema 4.6. *[2] El conjunto $\{H, T\}$ es universal para puertas de 1-qubit.*

Teorema 4.7. *[2] El conjunto $\{H, CNOT, T\}$ es un conjunto universal de puertas.*

Teorema 4.8. *[3] El conjunto $\{Toffoli, S, H\}$ es un conjunto universal de puertas.*

4.5. Medidas con circuitos cuánticos

En esta sección se indagará en como los circuitos cuánticos pueden ser usados para implementar medidas cuánticas, usando solo medidas en la base computacional y un conjunto universal de puertas (por simplicidad, se asumirá que las puertas pueden ser implementadas exactamente).

Dada una base ortonormal $|\phi_j\rangle$, se tiene un estado $|\psi\rangle$ escrito en la base:

$$|\psi\rangle = \sum_j \alpha_j |\phi_j\rangle \tag{4.33}$$

La medida de $|\psi\rangle$ con respecto a la base $\{|\phi_j\rangle\}$ se describe mediante los proyectores de cada subespacio 'j', $\{|\phi_j\rangle \langle\phi_j|\}$ y resultará en el espacio 'j' con probabilidad $|\alpha_j|^2$.

Se puede utilizar un circuito cuántico para implementar esta medida. Primero, se construye un circuito cuántico que implemente la transformación unitaria:

$$U |\phi_j\rangle = |j\rangle \tag{4.34}$$

(donde se supone que j esta escrito en binario mediante n bits y $|j\rangle$ es el estado de la base computacional de n qubit correspondiente). El operador U actúa cambiando de la base $\{|\phi_j\rangle\}$ a la base computacional. Dado el estado general $|\psi\rangle = \sum_j \alpha_j |\phi_j\rangle$, se utiliza el circuito para hacer el cambio de base U, y luego se hace la medida en la base computacional. Por último, se realiza el cambio de base inverso, U^{-1}, corriendo el circuito hacia atrás, con la inversa de todas las puertas. [2]

Como ejemplo, sea un espacio de Hilbert de dimensión 4 de 2 qubits, con la base ortonormal $\{|\beta_{00}\rangle, |\beta_{01}\rangle, |\beta_{10}\rangle, |\beta_{11}\rangle\}$, conocida como la base de Bell, donde:

$$\begin{aligned} |\beta_{00}\rangle &= \tfrac{1}{\sqrt{2}}|00\rangle + \tfrac{1}{\sqrt{2}}|11\rangle \quad |\beta_{01}\rangle = \tfrac{1}{\sqrt{2}}|01\rangle + \tfrac{1}{\sqrt{2}}|10\rangle \\ |\beta_{10}\rangle &= \tfrac{1}{\sqrt{2}}|00\rangle - \tfrac{1}{\sqrt{2}}|11\rangle \quad |\beta_{11}\rangle = \tfrac{1}{\sqrt{2}}|01\rangle - \tfrac{1}{\sqrt{2}}|10\rangle \end{aligned} \tag{4.35}$$

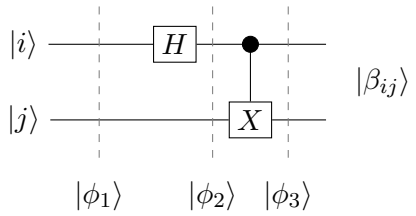

Figura 4.5.1: Un circuito para implementar el cambio de base, desde la computacional a la base de Bell (para el cambio de base inverso solo es necesario correr el circuito de forma inversa)

Se supone que el estado de entrada en la figura 4.5.1 es $|\phi_1\rangle = |00\rangle$. El estado se encuentra con la puerta de Hadamard en el primer qubit, por lo que cambia a:

$$|\phi_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\,|0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \tag{4.36}$$

Posteriormente pasa por c-X, (NOT controlada) y transforma al qubit en:

$$|\phi_3\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = |\beta_{00}\rangle \tag{4.37}$$

[2]

Análogamente, si $|\phi_1\rangle = |01\rangle$, al pasar por H en el primer qubit se obtiene

$$|\phi_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\,|1\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |11\rangle) \tag{4.38}$$

al pasar poseriormente por la c-NOT se obtiene

$$|\phi_3\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) = |\beta_{01}\rangle \tag{4.39}$$

Para $|\phi_1\rangle = |10\rangle$ se sigue que $|\phi_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\,|0\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |10\rangle)$. Aplicando la puerta *c-NOT* se obtiene $|\phi_3\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) = |\beta_{10}\rangle$.

Finalmente, para $|\phi_1\rangle = |11\rangle$ al aplicar H se obtiene $|\phi_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\,|1\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |11\rangle)$ y con la puerta *c-NOT* $|\phi_3\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) = |\beta_{11}\rangle$.

4.6. Diferencias entre computación cuántica y probabilística

Sea un circuito probabilístico simple. En la Figura 4.6.1 se muestran dos pasos de tal computación en un registro que puede estar en uno de los cuatro estados 0,1,2,3. La computación comienza con el estado 0 en cualquier caso. Los $p_{0,j}$ son las probabilidades de pasar del estado 0 la estado j en el primer paso y las $q_{j,k}$ son las probabilidades de pasar del estado j al estado k en el segundo paso. Si se desea saber cual es la probabilidad de que la computación acabe en un determinado estado hay que determinar la probabilidad de cada camino posible para llegar a este y hacer una suma de todos los caminos. Por ejemplo, para llegar al estado 3 hay cuatro posibles caminos. Iniciando en el 0, en el primer paso puede estar en el estado j donde $j \in \{0, 1, 2, 3\}$ y acabar en el estado 3 en el segundo paso. La probabilidad de tomar ese camino se obtiene multiplicando la probabilidad $p_{0,j}$ de la transición del 0 al j por la probabilidad $q_{j,3}$ de la transición del j al 3. Por lo tanto la probabilidad de acabar en el paso 3 es:

$$prob(reslultado\, final\, es\, 3) = \sum_{j=0}^{3} p_{0,j} q_{j,3} \tag{4.40}$$

Otra forma de ver este cálculo es suponer que el registro consta de dos qubits y dejar que las etiquetas 0, 1, 2, 3 se refieran a los cuatro estados básicos $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ respectivamente y ver cada una de las probabilidades como el cuadrado de la norma de la aplitud de probabilidad. Este

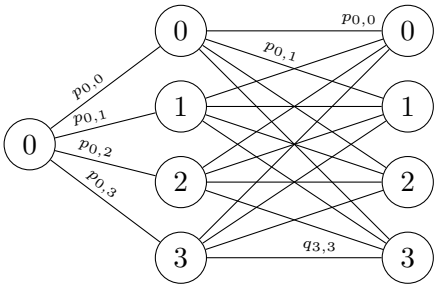

Figura 4.6.1: Una computación probabilística clásica actuando sobre un registro que puede estar en uno de los cuatro estados 0,1,2,3.

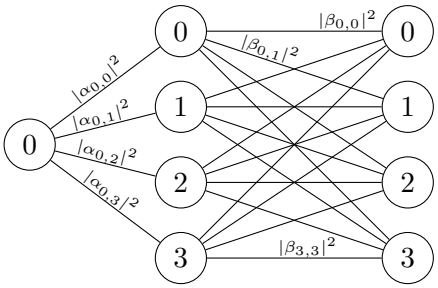

Figura 4.6.2: Una computación probabilística clásica vista desde un punto cuántico. Las probabilidades son sustituidas por el cuadrado de la norma de las amplitudes probabilísitcas de manera que $p_{0,j} = |\alpha_{0,j}^2|$ y $q_{j,k} = |\beta_{j,k}^2|$. Esto puede ser visto como una computación cuántica en la que se efectúa una medida en cada paso.

enfoque se puede observar en la Figura 4.6.2. Como antes, la probabilidad de obtener el estado 3 en el último paso es:

$$prob(reslultado\, final\, es\, 3) = \sum_{j=0}^{3} |\alpha_{0,j}|^2 |\beta_{j,3}|^2 = \sum_{j=0}^{3} |\alpha_{0,j}\beta_{j,3}|^2 \qquad (4.41)$$

En un algoritmo cuántico, no se haría una medida después de cada paso. De esta manera, las amplitudes de probabilidad podrán interferir. Una versión cuántica del algoritmo se ve en la Figura 4.6.1.

Esta vez, el cálculo de la probabilidad es distinto. Como no hay medida en el paso intermedio, no es posible saber cual de los caminos se siguen para

llegar al estado final. En este caso, en lugar de sumar las probabilidades de cada camino, se deben sumar las amplitudes de probabilidad y posteriormente hacer la medida, tomando la norma al cuadrado de la amplitud de probabilidad obtenida, es decir:

$$prob(reslultado\, final\, es\, 3) = \left| \sum_{j=0}^{3} \alpha_{0,j}\beta_{j,3} \right|^2 \tag{4.42}$$

Observación 2. *[2] No todos los algoritmos clásicos probabilísticos pueden ser simulados por algoritmos cuánticos de la manera que se muestra en esta sección. Para simular de la manera más general un algoritmo probabilístico con uno cuántico es necesario añadir qubits auxiliares.*

Esta observación viene del hecho de que dada una matriz unitaria $U = [u_{i,j}]$, la matriz dada por $S = [|u_{i,j}|^2]$ es una matriz estocástica, las puertas de la computación probabilística, pero no todas las matrices estocásticas son de esa manera. Para que una matriz sea estocástica se tiene que cumplir que todos los $a_{i,j} \geq 0$ y que $\sum_j a_{i,j} = 1$ para i fijo.

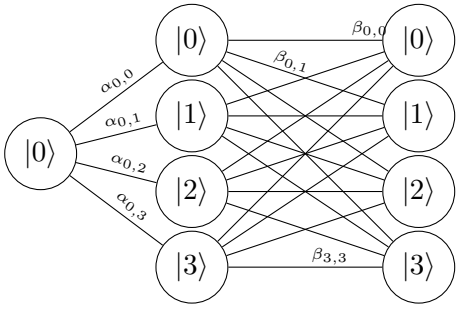

Figura 4.6.3: Una computación cuántica completa.

Para ilustrar como puede afectar la interferencia se considera el circuito mostrado en las Figura 4.6.4 y Figura 4.6.5. Ambos son circuitos con dos puertas de Hadamard (H), pero en el primero se efectua una medida al aplicar la primera.

El estado $|\phi_1\rangle$ nada más hacer la medida es:

$$|\phi_1\rangle = \begin{cases} |0\rangle & con\, probabilidad\, \frac{1}{2} \\ |1\rangle & con\, probabilidad\, \frac{1}{2} \end{cases} \tag{4.43}$$

El estado después de la segunda puerta de Hadamard es:

$$|\phi_2\rangle = \begin{cases} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) & con\, probabilidad\, \frac{1}{2} \\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) & con\, probabilidad\, \frac{1}{2} \end{cases} \tag{4.44}$$

En cualquier caso, la medida final dará $|0\rangle$ y $|1\rangle$ con igual probabilidad.

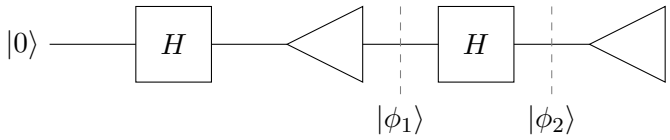

Figura 4.6.4: Circuito cuántico que no presenta interferencia

Sin embargo, en el circuito de la Figura 4.6.5, al no haber medida entre las dos puertas de Hadamard puede dar lugar a interferencias. El estado tras aplicar la primera puerta es:

$$|\varphi_1\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \tag{4.45}$$

A este estado se le aplica directamente otra puerta de Hadamard obteniendo:

$$\begin{aligned} |\varphi_2\rangle &= H\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{\sqrt{2}}H|0\rangle + \frac{1}{\sqrt{2}}H|1\rangle = \\ &= \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) + \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) = \\ &= \frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle + \frac{1}{2}|0\rangle - \frac{1}{2}|1\rangle = |0\rangle \end{aligned} \tag{4.46}$$

Figura 4.6.5: Circuito cuántico que presenta interferencia

Capítulo 5

Complejidad cuántica

Previamente se ha establecido que cualquier transformación unitaria puede ser aproximada usando las puertas de un conjunto universal, por ejemplo $\{H, CNOT, T\}$. Sin embargo, no se ha hablado sobre la eficiencia de esas aproximaciones. Lo más interesante sería poder implementar cualquier transformación unitaria en un tiempo polinomial, tanto en el número de qubits, n, como en $\frac{1}{\epsilon}$, donde ϵ es la precisión de la estimación de la transformación. [2]

La dificultad de implementar eficientemente una transformación unitaria no reside en la complejidad de simular puertas de 1 qubit arbitrarias a partir de una familia de puertas de 1 qubit. El teorema de Solovay-Kitaev afirma que se puede encontrar un conjunto $\mathcal{G}$ de puertas de 1 qubit a partir de la cual se puede aproximar cualquier puerta de 1 qubit en tiempo poli-logarítmico. Es decir, si se quiere aproximar una puerta de 1 qubit con un error menor que ϵ, se puede implementar en un número de puertas polinómico en $log(\frac{1}{\epsilon})$. [2]

Teorema 5.1 (Solovay-Kitaev)**.** *[2] Si $\mathcal{G} = \{R_l(\beta), R_m(\gamma)\}$ es una familia finita de puertas de 1 qubit satisfaciendo:*

1. *l y m son dos ejes no paralelos de la esfera de Bloch*

2. *$\beta, \gamma \in [0, 2\pi)$ son números reales tales que $\frac{\beta}{\pi}$ y $\frac{\gamma}{\pi}$ son no racionales*

3. *para cualquier puerta $g \in \mathcal{G}$, su inversa g^{-1} puede ser implementada con una secuencia finita de puertas de $\mathcal{G}$*

entonces cualquier puerta de 1 qubit puede ser aproximada con un error máximo de ϵ usando $\mathcal{O}(log^c(\frac{1}{\epsilon}))$ número de puertas de $\mathcal{G}$, donde c es una constante positiva.

5.1. Modelo de cajas negras

Una caja negra no es más que un circuito reversible que implementa cierta función f del cual no se puede obtener ninguna información del funcionamiento interno. El circuito cuántico que implementa f más general viene dado por la expresión [2]

$$U_f : |x\rangle |y\rangle \longmapsto |x\rangle |y \oplus f(x)\rangle \tag{5.1}$$

En el modelo de cajas negras, la entrada está proporcionada por una caja negra, $O_{\mathbf{X}}$ para acceder a la cadena desconocida $\mathbf{X} = X_1, ..., X_N$, donde las variables X_i son binarias. Por lo tanto:

$$O_{\mathbf{X}} : |j\rangle |b\rangle \longrightarrow |j\rangle |b \oplus X_j\rangle \tag{5.2}$$

El objetivo normalmente es computar una función $F(\mathbf{X})$ de la cadena $\mathbf{X}$. Una computación en el modelo de caja negra es aquella que calcula una función $F : \{0,1\}^N \longrightarrow \{0,1\}$ dado acceso a la caja negra $O_{\mathbf{X}}$. La computación puede realizar cualquier operación unitaria y realiza consultas a la caja negra. Sin estas consultas a $O_{\mathbf{X}}$ no se puede evaluar ningún $F(\mathbf{X})$ no trivial en tanto que no existe ninguna información sobre $\mathbf{X}$. El objetivo de la caja negra es obtener suficiente información sobre $\mathbf{X}$ usando $O_{\mathbf{X}}$ para poder computar $F(\mathbf{X})$ de forma fiable. [2]

La complejidad de consulta de un algoritmo es el número de consultas utilizadas por el algoritmo y, análogamente, la complejidad de consulta de un problema es el número de consultas necesarias para resolverlo.

Las cajas negras pueden ser reemplazadas por 'cajas blancas', que no son más que circuitos que implementan las cajas negras. En este proceso, la complejidad total del algoritmo puede ser acotada superiormente por $TB + A$, donde T es la complejidad de consulta del algoritmo de caja negra, B es la complejidad de realizar una consulta y A la complejidad de las operiaciones que no son de consulta realizadas por el algoritmo de caja negra. [2]

Por ejemplo, considerando una fórmula 3-SAT Φ (definida en la sección 3.5) en n variables $x_1, .., x_n, N = 2^n$ y sean los números $1, .., N$ que codifican las variables $x_1, .., x_n$. Se define la función f_Φ tal que $f_\Phi(y) = 1$ si la asignación $x_1 = y_1, ..., x_n = y_n$ satisface la fórmula Φ, es decir es igual a 1, y $f_\Phi(y) = 0$ en otro caso. Se puede reformular el problema 3-SAT de la siguiente manera. Se definen las N variables $X_1, .., X_N$ tales que $X_j = f_\Phi(j)$ y se resuelve el problema de búsqueda para $\mathbf{X} = X_1, .., X_N$.

5.2. Distinguibilidad del estado

En general, para probar que son necesarias T consultas se demuestra que con menos de estas T consultas el algoritmo no puede distinguir de manera fiable la caja negra $O_\mathbf{X}$ que satisface $F(\mathbf{X}) = 1$ de la caja negra $O_\mathbf{Y}$ que satisface $F(\mathbf{Y}) = 0$. Sea el algoritmo $\mathcal{A}$ que hace T consultas y sean $|\phi_\mathbf{X}\rangle, |\phi_\mathbf{Y}\rangle$ los estados producidos por $\mathcal{A}$ con los oráculos $O_\mathbf{X}, O_\mathbf{Y}$ respectivamente. Para que el algoritmo compute fiablemente $F(\mathbf{X})$ y $F(\mathbf{Y})$ es necesario poder distinguir los estados.

Teorema 5.2. *[2] Cualquier proceso que al ingresar $|\phi_\mathbf{Z}\rangle$ averigüe si $\mathbf{Z} = \mathbf{X}$ o $\mathbf{Z} = \mathbf{Y}$ acertará como máximo con una probabilidad $1 - \epsilon = \frac{1}{2} + \frac{1}{2}\sqrt{1 - \delta^2}$ donde $\delta = \langle\phi_\mathbf{X}|\phi_\mathbf{Y}\rangle$*

Distinguir dos estados cuánticos puros con el mínimo error

Input: Dos estados cuánticos, uno de ellos conocido, $|\phi_{\mathbf{X}}\rangle$ o $|\phi_{\mathbf{Y}}\rangle$ con la propiedad de $|\langle\phi_{\mathbf{X}}|\phi_{\mathbf{Y}}\rangle| = \delta$.

Output: Una conjetura, '**X**' o '**Y**'.

Problema: Optimizar la probabilidad $1 - \epsilon$ de que la conjetura sea correcta.

5.3. El problema de búsqueda: el algortimo cuántico de Grover.

Dada una función $f : \{0,1\}^n \longrightarrow \{0,1\}$ el problema de búsqueda es encontrar $x \in \{0,1\}^n$ tal que $f(x) = 1$. El problema de decisión consiste en determinar si existe o no una solución para el problema de búsqueda. Una solución al problema de búsqueda dara una solución al problema de decisión por lo que un límite inferior para en la complejidad el problema de decisión implica un límite inferior en la complejidad del problema de búsqueda. Se podría definir el problema como sigue: [2]

El problema de búsqueda

Input: Una caja negra U_f para computar una función desconocida $f : \{0,1\}^n \to \{0,1\}$.

Problema: Encontrar un imput $x \in \{0,1\}^n$ tal que $f(x) = 1$.

Inicialmente, por simplicidad, se supone que la función f tiene una única solución $x = w$. Se asume que se desea que el proceso encuentre la solución con probabilidad $\frac{2}{3}$ para cada función f. [1] [2]

[1] Como se dijo anteriormente, la elección de $\frac{2}{3}$ es arbitraria, y cualquier constante entre $\frac{1}{2}$ y 1 es suficiente.

Si solo está permitida una consulta, lo mejor que el algoritmo puede hacer es elegir una solución x_1 uniformemente al azar y usar la consulta para comprobar si $f(x_1) = 1$. Si x_1 es la respuesta correcta, imprime x_1. En otro caso elegir otra cadena $x_2 \in \{0,1\}^n \backslash \{x_1\}$ aleatoriamente e imprimir x_2. La probabilidad de que imprima la respuesta correcta es $\frac{2}{2^n}$. [2]

Con dos consultas, siguiendo con el proceso se utiliza la segunda para comprobar si $f(x_2) = 1$, en cuyo caso se imprime x_2. Si no, se elige otra cadena $x_3 \in \{0,1\}^n \backslash \{x_1, x_2\}$ aleatoriamente e imprimir x_3. La probabilidad en este caso es $\frac{3}{2^n}$. [2]

Si se continua el proceso, para k consultas, el proceso imprimirá el valor correcto $x = w$ con probabilidad $\frac{k+1}{2^n}$. Notar que, sin consultas, la probabilidad de acertar sería $\frac{1}{2^n}$ y por cada consulta se aumenta la probabilidad en $\frac{1}{2^n}$.

Ahora sea una versión cuántica del algoritmo naíf que resuelve el problema sin consultas. Este lo hace con probabilidad $\frac{1}{2^n}$, por lo que su versión cuántica lo resuelve con una amplitud de probabilidad de $\frac{1}{\sqrt{2^n}}$. Si hubiera alguna manera de amplificar la probabilidad $\frac{1}{\sqrt{2^n}}$ por cada consulta, solo serían necesarias $O(\sqrt{2^n})$ consultas para resolverlo. Grover encontró un algoritmo que consigue ese aumento de probabilidad. Esta es una descripción intuitiva del algoritmo, a continuación se precisa la idea. [2]

Se supone que existe un medio para reconocer una solución y por tanto se puede asumir que la caja negra U_f para f se implementa como sigue: [2]

$$U_f : |x\rangle\,|b\rangle \longmapsto |x\rangle\,|b \oplus f(x)\rangle \tag{5.3}$$

donde $|b\rangle$ es un único qubit. Si se establece $|b\rangle$ a $|0\rangle$ y, dado un valor de consulta x codificado en el primer registro, o registro de consulta como $|x\rangle$, se hace actuar U_f el resultado es: [2]

$$|x\rangle\,|0\rangle \xmapsto{U_f} |x\rangle\,|f(x)\rangle \tag{5.4}$$

y midiendo el qubit objetivo se consigue la respuesta a la consulta de caja

negra de f. Pero esto no es mejor que lo conseguido clasicamente. Para ganar 'ventaja cuántica' es necesario hacer uso de la superposición. [2]

Se puede preparar fácilmente el primer registro en una superposición de todos los posibles valores de consulta como $\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$, con $N = 2^n$. [2]

Esta suma se puede dividir en dos partes. La primera parte es una suma sobre los valores de x para los cuales $f(x) = 0$, es decir, los 'malos' x que no son solución del problema. Sea X_{malo} el conjunto con dichos x. La segunda parte es una suma sobre los valores de x para los cuales $f(x) = 1$, es decir, los 'buenos' x que si son solución. Sea X_{bueno} el conjunto con dichos x. Por conveniencia, se asume que solo hay una solución w, por lo que $X_{bueno} = \{w\}$. Se definen los estados [2]

$$\begin{aligned} |\varphi_{bueno}\rangle &= |w\rangle \\ |\varphi_{malo}\rangle &= \frac{1}{\sqrt{N-1}} \sum_{x \in X_{malo}} |x\rangle \end{aligned} \tag{5.5}$$

Se prepara el qubit objetivo de U_f en el estado $|0\rangle$ y el registro de consulta en una superposición de la forma:

$$\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle = \frac{1}{\sqrt{N}} |w\rangle + \sqrt{\frac{N-1}{N}} |\varphi_{malo}\rangle \tag{5.6}$$

Ahora con probabilidad $\frac{1}{N}$ la medida del qubit objetivo dará $|1\rangle$ y los qubits de consulta se quedarán en el buen estado $|w\rangle$. Aunque este algoritmo usa el principio de superposición, no hace empleo de la interferencia cuántica, por lo que puede ser simulado por uno clásico. El algoritmo cuántico del problema de búsqueda es un proceso iterativo que utiliza la interferencia cuántica [2]

Dado que se puede ver la acción de U_f como una puerta controlada actuando sobre el qubit objetivo y controlada por el registro de consulta, si se prepara el qubit objetivo en la superposición $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ el efecto del oráculo es: [2]

$$|x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \longmapsto (-1)^{f(x)} |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \tag{5.7}$$

Como el qubit objetivo es un estado propio, se puede omitir obteniendo:

$$U_f : |x\rangle \longmapsto (-1)^{f(x)} |x\rangle \tag{5.8}$$

Es conveniente reescribir U_f para que sea el operador de n qubits que actua como (5.8). También se define el operador de n qubits $U_{0^\perp}$ que actua como sigue: [2]

$$U_{0^\perp} : \begin{cases} |x\rangle \mapsto -|x\rangle, & x \neq 0 \\ |0\rangle \mapsto |0\rangle \end{cases} \tag{5.9}$$

Este operador aplica un cambio de fase de -1 a todos los estados ortogonales al $|00...0\rangle$. Denotando V_0 el espacio generado por este estado, entonces el espacio ortogonal a V_0 es el espacio generado por los vectores de la base $|x\rangle \neq |00...0\rangle$, denotado $V_0^\perp$. El operador $U_{0^\perp}$ aplica una fase de -1 a todos los vectores de $V_0^\perp$. [2]

Ahora se define un operador cuya función es aumentar la amplitud de $|\varphi_{bueno}\rangle = |w\rangle$. Este operador es $G = HU_{0^\perp}HU_f$ y se denomina la iteración de Grover, que viene dada por: [2]

1. Aplicar la caja negra U_f
2. Aplicar la puerta de Hadamard de n qubits H
3. Aplicar $U_{0^\perp}$
4. Aplicar la puerta de Hadamard de n qubits H

Una vez definida la iteración de Grover, se puede definir el algoritmo cuántico de búsqueda de Grover. [2]

El algoritmo cuántico de búsqueda de Grover

1. Empezar con el estado de n-qubit $|00...0\rangle$
2. Aplicar la puerta de Hadamard de n qubits para preparar el estado $\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$ con $N = 2^n$
3. Aplicar la iteración de Grover un total de $\left\lfloor \frac{4}{\pi}\sqrt{N} \right\rfloor$ veces.
4. Medir el estado resultante.

Se verá ahora un explicación breve de como realmente la iteración de Grover aumenta la amplitud de $|\varphi_{bueno}\rangle$. Sea [2]

$$|\varphi\rangle := H|00...0\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle = \frac{1}{\sqrt{N}} |w\rangle + \sqrt{\frac{N-1}{N}} |\varphi_{malo}\rangle \tag{5.10}$$

Considerando la actuación de $HU_{0^\perp}H$ sobre $|\varphi\rangle$, al ser H autoinverso, [2]

$$HU_{0^\perp}H : |\varphi\rangle \longmapsto |\varphi\rangle \tag{5.11}$$

Sea $V_\varphi^\perp$ el espacio ortogonal a $|\varphi\rangle$. Este espacio es el generado por los vectores $H|x\rangle$ con $x \neq 00...0$, y para estos vectores se tiene: [2]

$$HU_{0^\perp}H : H|x\rangle \longmapsto -H|x\rangle \tag{5.12}$$

por lo que este operador aplica una fase de -1 a los vectores ortogonales a $|\varphi\rangle$, por lo que se puede denotar $HU_{0^\perp}H = U_{\varphi^\perp}$ en analogía a (5.9) y $G = U_{\varphi^\perp}U_f$. [2]

Viendo el estado $|\varphi\rangle$ como combinación lineal de $|w\rangle$ y $|\varphi_{malo}\rangle$ y aplicando repetidamente los operadores U_f y $U_{\varphi^\perp}$ deja el estado del sistema en un subespacio bidimensional del espacio de N dimensiones generado por $|w\rangle$ y $|\varphi_{malo}\rangle$. Se definen unas nuevas bases de este subespacio para analizar el

algoritmo:[2]

$$\{|w\rangle, |\varphi_{malo}\rangle\} \tag{5.13}$$

y

$$\{|\varphi\rangle, |\overline{\varphi}\rangle\} \tag{5.14}$$

donde se define $|\overline{\varphi}\rangle$ como el estado ortogonal a $|\varphi\rangle$.

$$|\overline{\varphi}\rangle = \sqrt{\frac{N-1}{N}}\,|w\rangle - \frac{1}{\sqrt{N}}\,|\varphi_{malo}\rangle \tag{5.15}$$

Se define θ tal que

$$sin(\theta) = \frac{1}{\sqrt{N}}, \quad cos(\theta) = \sqrt{\frac{N-1}{N}} \tag{5.16}$$

Notar que:

$$|\varphi\rangle = sin(\theta)\,|w\rangle + cos(\theta)\,|\varphi_{malo}\rangle \tag{5.17}$$

$$|\overline{\varphi}\rangle = cos(\theta)\,|w\rangle - sin(\theta)\,|\varphi_{malo}\rangle \tag{5.18}$$

$$|w\rangle = sin(\theta)\,|\varphi\rangle + cos(\theta)\,|\overline{\varphi}\rangle \tag{5.19}$$

$$|\varphi_{malo}\rangle = cos(\theta)\,|\varphi\rangle - sin(\theta)\,|\overline{\varphi}\rangle \tag{5.20}$$

El algoritmo empieza en el estado $|\varphi\rangle$ dado por la Ecuación 5.17. El operador U_f da el estado

$$\begin{aligned} U_f\,|\varphi\rangle &= (-1)^1 sin(\theta)\,|w\rangle + (-1)^0 cos(\theta)\,|\varphi_{malo}\rangle = \\ &= -sin^2(\theta)\,|\varphi\rangle - sin(\theta)cos(\theta)\,|\overline{\varphi}\rangle + cos^2(\theta)\,|\varphi\rangle - cos(\theta)sin(\theta)\,|\overline{\varphi}\rangle = \\ &= cos(2\theta)\,|\varphi\rangle - sin(2\theta)\,|\overline{\varphi}\rangle \end{aligned} \tag{5.21}$$

El operador $U_{\varphi^\perp}$ actúa sobre este último estado

$$U_{\varphi^\perp}U_f\,|\varphi\rangle = cos(2\theta)\,|\varphi\rangle + sin(2\theta)\,|\overline{\varphi}\rangle = sin(3\theta)\,|w\rangle + cos(3\theta)\,|\varphi_{malo}\rangle \tag{5.22}$$

Es fácil demostrar por inducción que, tras k interacciones de la iteración de Grover empezando con el estado $|\varphi\rangle$, se obtiene el estado

$$\begin{aligned} (U_{\varphi^\perp}U_f)^k\,|\varphi\rangle &= cos(2k\theta)\,|\varphi\rangle - sin(2k\theta)\,|\overline{\varphi}\rangle = \\ &= sin((2k+1)\theta)\,|w\rangle + cos((2k+1)\theta)\,|\varphi_{malo}\rangle \end{aligned} \tag{5.23}$$

Con el fin de encontrar una alta probabilidad de obtener $|w\rangle$ se desea seleccionar k de manera que $sin((2k+1)\theta) \approx 1$, que significa que $(2k+1)\theta \approx \frac{\pi}{2}$ y por tanto $k \approx \frac{\pi}{4\theta} - \frac{1}{2} \approx \frac{\pi}{4}\sqrt{N}$. Sea $\tilde{k}$ tal que $(2\tilde{k}+1)\theta = \frac{\pi}{2}$. Sea $k = \lfloor \tilde{k} \rfloor$. Notar que $(2k+1)\theta = \frac{\pi}{2} + \epsilon$ con $|\epsilon| \in \mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$. De este modo, $sin(\frac{\pi}{2} + \epsilon) = cos(\epsilon) \geq 1 - \frac{\epsilon^2}{2} \in 1 - \mathcal{O}(\frac{1}{N})$ [2]

Teorema 5.3. *[2] Sea f una función con exactamente una solución. Sea $k = \lfloor \tilde{k} \rfloor$, para $\tilde{k} = \frac{\pi}{4\theta} - \frac{1}{2}$. Aplicar el algoritmo cuántico para el problema de búsqueda con k aplicaciones de la iteración de Grover encontrará una solución para $f(x) = 1$ con probabilidad de al menos $1 - \mathcal{O}(\frac{1}{N})$.*

La suposición de tener una sola solución no es necesaria. Se puede conseguir la misma probabilidad para funciones con más de una solución con un proceso similar, añadiendo qubits auxiliares. [2]

Para que el algoritmo encuentre solución con tal probabilidad son necesarias k iteraciones, por lo que es necesario conocer el valor de θ, es decir, de N. Para los casos en los que N es desconocido existe un algoritmo basado en estimar la amplitud de $sin(\theta)$ que también encuentra solución en $\mathcal{O}\left(\frac{1}{\theta}\right)$. [2]

5.4. Límites inferiores para el problema de búsqueda: el método híbrido.

Hasta ahora se han demostrado límites inferiores para el problema de búsqueda, es decir, si el espacio de posibles soluciones tiene N elementos, se puede encontrar una solución en, como máximo, $\mathcal{O}(\sqrt{N})$ consultas a la caja negra U_f. Lo natural es pensar si existe algún otro algoritmo que resuelva el problema con menos consultas. En esta sección se mostrará que ningún algoritmo de caja negra puede resolver el problema de búsqueda con menos de $\Omega(\sqrt{N})$ consultas. [2]

Definición 5.1. *[2] Sea $\boldsymbol{X}_x$ la cadena con un 1 en la posición x y 0 en el resto, es decir, $\boldsymbol{X}_x = 1$ y $\boldsymbol{X}_y = 0$ $\forall y \neq x$. Sea $S = \{\boldsymbol{X}_x : x = 1, ..., N\}$ el conjunto con todas las cadenas $\{0,1\}^N$ en las que hay exactamente un 1.*

Teorema 5.4. *[2] Cualquier argoritmo cuántico con error acotado que determinará por cada $\boldsymbol{X} \in S \cup \{\boldsymbol{0}\}$ si existe j tal que $X_j = 1$ debe hacer $\Omega(\sqrt{N})$ consultas a $O_{\boldsymbol{X}}$.*

De este teorema se obtienen algunos corolarios.

Corolario 5.5. *[2] Sea $T \subseteq \{0,1\}^N$ satisfaciendo $S \cup \{\boldsymbol{0}\} \subseteq T$. Cualquier algoritmo cuántico con error acotado que determinará para cada $\boldsymbol{X} \in T$ si existe j tal que $X_j = 1$ debe hacer $\Omega(\sqrt{N})$ consultas.*

Corolario 5.6. *[2] Sea $T \subseteq \{0,1\}^N$ satisfaciendo $S \subseteq T$. Cualquier algoritmo cuántico con error acotado que encontrará un j tal que $X_j = 1$ para cada $\boldsymbol{X} \in T$ debe hacer $\Omega(\sqrt{N})$ consultas.*

5.5. Límites inferiores generales en la caja negra.

En esta sección se describirán métodos para computar cualquier función F de N variables binarias. [2]

Si una función F está definida para todos los posibles valores de entrada $\{0,1\}^N$ se dice función total. En otro caso se dice función parcial. El problema de evaluar una función parcial también se le puede llamar como problema de promesa, en tanto que se está prometiendo que la entrada a la función tiene una forma específica. [2]

Antes de probar límites inferiores en el modelo de caja negra, se mostrará que los algoritmos cuánticos dan una ventaja polinomial como máximo con respecto a los algortimos cuánticos para funciones totales. [2]

Definición 5.2. *[2] La complejidad de consulta determinista $D(F)$ de F es el mínimo número de consultas a $O_{\boldsymbol{X}}$ requerido por un proceso clásico determinista para computar $F(\boldsymbol{X})$ para cualquier $\boldsymbol{X} \in \{0,1\}^N$.*

El análogo cuántico de $D(F)$ es la complejidad cuántica de consulta exacta $Q_E(F)$ de F.

Definición 5.3. *[2] La complejidad cuántica de consulta exacta $Q_E(F)$ de F es el mínimo número de consultas a $O_{\boldsymbol{X}}$ requerido por un algoritmo cuántico que computa correctamente $F(\boldsymbol{X})$ con probabilidad 1 para cualquier $\boldsymbol{X} \in \{0,1\}^N$.*

La complejidad cuántica de consulta exacta no es tan natural como su contraparte clásica en tanto que la probabilidad de éxito no suele ser exactamente 1. Una cantidad más relevante es la complejidad cuántica de consulta de error bilateral $Q_2(F)$ de F. [2]

Definición 5.4. *[2] La complejidad cuántica de consulta de error bilateral $Q_2(F)$ de F es el mínimo número de consultas a $O_{\boldsymbol{X}}$ requerido por un algoritmo cuántico que, para cualquier entrada $\boldsymbol{X} \in \{0,1\}^N$, produce un valor en $\{0,1\}$ que con probabilidad al menos $\frac{2}{3}$ es igual a $F(\boldsymbol{X})$.*

Teorema 5.7. *[2] Si F es una función total Booleana, entonces $D(F) \leq 2^{12} Q_2(F)^6$.*

Se dice que una función F es simétrica si cualquier permutación de $\mathbf{X}$ no cambia el valor de $F(\mathbf{X})$. En otras palabras, F solo depende del número de 1's en $\mathbf{X}$ y no de su posición.

Teorema 5.8. *[2] Si F es una función simétrica Booleana, entonces $D(F) \in \mathcal{O}(Q_2(F)^2)$.*

¿Qué significan estos teoremas para la computación cuántica? Si la mejor estrategia determinista clásica requiere, en el peor de los casos, $T = D(F)$ consultas para evaluar $F(\mathbf{X})$, el Teorema 5.7 dice que cualquier algoritmo cuántico requiere al menos $\frac{T^{\frac{1}{6}}}{4}$ consultas para computar F y si ésta es simétrica, se requieren $\Omega(\sqrt{T})$. [2]

El Teorema 5.7 es el que nos dice que la complejidad cuántica de consulta es como máximo polinomialmente mejor que la complejidad clásica de consulta en funciones totales.

5.6. Método polinomial

En esta sección se mostrará como un circuito cuántico que hace T consultas a una cadena $\mathbf{X}$ tendrá amplitudes que son polinomios de grado T en las variables $X_1, ..., X_N$. Si $T = 0$ las amplitudes son independientes de las variables y el circuito computa una función constante.

Lema 5.9. *[2] Sea $\mathcal{N}$ un circuito cuántico que utiliza m qubits y hace T consultas a un oráculo $O_{\boldsymbol{X}}$. Entonces existen polinomios multilineales de N variables complejas $p_0, p_1, ..., p_{2^m-1}$, cada uno de grado máximo T, tales que el estado final del circuito esta en la superposición*

$$\sum_{y=0}^{2^m-1} p_y(\boldsymbol{X}) \left|y\right\rangle \tag{5.24}$$

para cualquier oráculo $O_{\boldsymbol{X}}$

El siguiente colorario se sigue del hecho que si la amplitud de un vector de la base es un polinomio $\alpha(\mathbf{X})$ de orden a lo sumo T en las variables $X_1, ..., X_N$ entonces la probabilidad de medir ese estado vendrá dada por $\alpha(\mathbf{X})\alpha(\mathbf{X})^*$, un polinomio de grado a lo sumo $2T$. [2]

Corolario 5.10. *[2] Sea $\mathcal{N}$ un circuito cuántico que hace T consultas a un oráculo $O_{\boldsymbol{X}}$ y $\mathcal{B}$ una base de estados. Entonces existe un polinomio real multilineal P de grado a lo sumo $2T$ que es igual a la probabilidad de observar un estado de la base $\mathcal{B}$ después de aplicar el circuito $\mathcal{N}$ usando el oráculo $O_{\boldsymbol{X}}$.*

5.6.1. Aplicaciones a límites inferiores.

Primeramente se definirán las cantidades $deg(F)$ y $\widetilde{deg}(F)$ relacionados con la función de N variables F.

Definición 5.5. *[2] Un polinomio de N variables $p : \mathbb{R}^n \longrightarrow \mathbb{R}$ representa a F si $p(\boldsymbol{X}) = F(\boldsymbol{X})$ para todo $\boldsymbol{X} \in \{0,1\}^N$.*

Lema 5.11. *[2] Toda función de N variables $F : \{X_1, ..., X_N\} \longrightarrow \{0,1\}$ tiene un único polinomio multilineal $p : \mathbb{R}^n \longrightarrow \mathbb{R}$ que la representa*

Definición 5.6. *[2] El grado del polinomio p que representa F se denota $deg(F)$.*

Por ejemplo, la función OR se representa con $1 - \prod_{j=1}^{N}(1 - X_j)$ por lo que $deg(OR) = N$. En la práctica sin embargo será suficiente tener un polinomio p que aproxime la función F para cada $\mathbf{X} \in \{0,1\}^N$. Por ejemplo $OR(X_1, X_2) \approx \frac{2}{3}(X_1 + X_2)$.[2]

Definición 5.7. *[2] Un polinomio de N variables $p : \mathbb{R}^n \longrightarrow \mathbb{R}$ aproxima F si $|p(\boldsymbol{X}) - F(\boldsymbol{X})| \leq \frac{1}{3}$ para todo $\boldsymbol{X} \in \{0,1\}^N$.*

Definición 5.8. *[2] El grado mínimo de un polinomio p que aproxima F se denota $\widetilde{deg}(F)$.*

Teorema 5.12. *Si F es una función Booleana, entonces $Q_E(F) \geq \frac{deg(F)}{2}$ y $Q_2(F) \geq \frac{\widetilde{deg}(F)}{2}$.*

5.7. Sensibilidad de bloque

Intiutivamente, uno podría pensar que las funciones que son muy sensibles a los cambios en casi cualquiera de los bits de una cadena $\mathbf{X}$ requerirá sondear mas bits de $\mathbf{X}$ que cualquier otra función que es más indiferente a ciertos cambios. La manera rigurosa de ilustrar este concepto es la sensibilidad de bloque. [2]

Definición 5.9. *[2] Sea* $F : \{0,1\}^N \longrightarrow \{0,1\}$ *una función,* $\boldsymbol{X} \in \{0,1\}^N$ *y* $B \subseteq \{1,2,...,N\}$ *un subconjunto de índices.*

Sea $\boldsymbol{X}^B$ *la cadena obtenida de* $\boldsymbol{X}$ *invirtiendo los valores de las variables en* B*.*

La función F *es sensible a* B *si* $F(\boldsymbol{X}) \neq F(\boldsymbol{X}^B)$*.*

La sensibilidad de bloque $bs_{\boldsymbol{X}}(F)$ *de* F *en* $\boldsymbol{X}$ *es el máximo número* t *para el cual existen* t *subconjuntos disjuntos de índices,* $B_1, .., B_t$ *tales que* F *es sensible a cada* B_i *en* $\boldsymbol{X}$*.*

La sensibilidad de bloque en F*,* $bs(F)$ *es el máximo de* $bs_{\boldsymbol{X}}(F)$ *sobre todos los* $\boldsymbol{X} \in \{0,1\}^N$*.*

Teorema 5.13. *[2] Si* F *es una función Booleana, entonces* $Q_E(F) \geq \sqrt{\frac{bs(F)}{8}}$ *y* $Q_2(F) \geq \sqrt{\frac{bs(F)}{16}}$

Capítulo 6

El problema del subgrupo oculto.

En este capítulo se hablará del problema del subgrupo oculto, $HSP(G)$, sobre un grupo finito G. En el HSP, se tiene acceso a una caja negra que computa una función $f : G \longrightarrow \{0,1\}^*$ que es constante en las clases laterales[1] de un subgrupo oculto $H \leq G$. El problema consiste en encontrar el subgrupo H mediante, por ejemplo, una lista de sus generadores[2]. [3]

Teorema 6.1. *[3]* $HSP(\mathbb{Z}_2^n) \in BQP^f$

Recordando, la clase BQP es la clase de comlpejidad de problemas que pueden ser resueltos por un ordenador cuántico en tiempo polinomial, con un error de $\frac{1}{3}$. La clase BQP entonces es la clase de problemas resueltos por un ordenador cuántico con un número polinomial de iteraciones, en el cual,

---

[1]Dado un grupo G, $H \leq G$ un subgrupo y $g \in G$, se define la clase lateral derecha de H (respectivamente izquierda) al conjunto $Hg := \{hg | h \in H\}$ [5]

[2]Dado un grupo G y un subgrupo $H \leq G$ se dice $\langle H \rangle := \{h_1^{i_1} ... h_n^{i_n} | h_j \in H, i_j = \pm 1\}$

cada iteración viene dada por la función f.

El $HSP(\mathbb{Z}_2)^n$ es equivalente al problema de Simon. Es decir, el problema del subgrupo oculto implementado en un circuito cuántico es una versión cuántica del problema de Simon. En este problema, se tiene acceso de caja negra a una función $f : \{0,1\}^n \longrightarrow \{0,1\}^n$. Además se asegura que la función cumple uno de los dos siguientes enunciados:[3]

1. f es 1-a-1.

2. existe una "*cadena secreta*" $s \neq 0^n$ tal que, para todo $x \neq y$, se tiene $f(x) = f(y)$ si y solo si $y = x \oplus s$.

El problema es encontrar cual de los dos enunciados se cumple. Notar que el enunciado 1 es equivalente a "f esconde el subgrupo trivial $H = \{0^n\}$" y el enunciado 2 corresponde a "f esconde el subgrupo $H = \{0^n, s\}$ de orden 2" . [6]

El algoritmo cuántico diseñado para este problema permite resolverlo en $\mathcal{O}(n)$ consultas a f, y un total de $n^{\mathcal{O}(1)}$ computaciones en total. Este algoritmo es como sigue: [3]

1. Se prepara una superposición con igual amplitud sobre todas las posibles 2^n entradas a f

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

2. Consultar a f para obtener

$$\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \, |0^n \oplus f(x)\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \, |f(x)\rangle$$

   En realidad, no es necesario saber el valor de $f(x)$, sino lo que la computación $f(x)$ hace con el qubit original $|x\rangle$. Es decir, no es necesario saber como es f, solo saber si es del tipo 1 o 2.

3. Medir $|f(x)\rangle$ (verdaderamente no es necesario este paso, pero es útil para clarificar). Si la función es 1-a-1 (inyectiva), lo que quedará en la salida será un estado de la base computacional, $|x\rangle$. Si la función es 2-a-1, lo que quedará en la salida será una superposición de dos entradas, x e y con $f(x) = f(y)$ y por lo tanto $y = x \oplus s$:

$$\frac{|x\rangle + |y\rangle}{\sqrt{2}}$$

La función siempre cumple uno de los dos casos 1 o 2, o bien es inyectiva, y entonces solo hay un x para cada $f(x)$ o bien existe s tal que $f(x) = f(y) \Leftrightarrow y = x \oplus s$. Al medir el registro $f(x)$ se dan dos casos, uno en el que solo hay un x para cada $f(x)$ y entonces es 1-1 o el caso en el que hay varios x para una misma imagen, que entoces, por las suposiciones, se cumple lo segundo. El objetivo es saber qué es esa s. Con este objetivo se formula la siguiente pregunta, ¿qué medida sobre la superposición dará la suficiente información sobre s? Claramente, una medida sobre los estados de la base dará con igual probabilidad x e y, por lo que es inútil. Y tampoco se puede medir el estado dos veces. Se podría repetir el proceso, pero con una alta probabilidad daría un par (x, y) diferente por lo que el número de repeticiones crecería exponencialmente antes de obtener información de s. Por lo tanto, hay que probar algo diferente. Este paso en si es innecesario, por eso se llega a que no sirve de nada medir el registro $|f(x)\rangle$.

4. Una vez visto en un paréntesis lo que supone medir $|f(x)\rangle$, evidentemente no se realiza la medida. El siguiente paso real por tanto es aplicar la puerta de Hadamard H a cada uno de los n qubits en el registro $|x\rangle$ y sólamente después medir ese registro sobre la base computacional. Esto mapeará el qubit $|x\rangle$ a

$$\frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} (-1)^{z \cdot x} |z\rangle$$

por lo tanto, mapeará $\frac{1}{\sqrt{2}}(|x\rangle + |y\rangle)$ a

$$\frac{1}{\sqrt{2^{n+1}}} \sum_{z \in \{0,1\}^n} (-1)^{x \cdot z + y \cdot z} |z\rangle$$

Las únicas medidas sobre z no nulas serán aquellas para las cuales las contribuciones a la aplitud de x e y interfieran no destructivamente, es decir, sean ambas positivas o negativas, lo que se traduce en que

$$x \cdot z \equiv y \cdot z \qquad (mod\, 2)$$

o lo que es lo mismo

$$(x \oplus y) \cdot z \equiv 0 \qquad (mod\, 2)$$

para que z pueda ser observado. Como $y = x \oplus s \Rightarrow s = x \oplus y$. Por tanto $s \cdot z \equiv 0\ (mod\, 2)$. En conclusión, no se obtiene s en la medida, pero se obtiene un z aleatorio tal que $s \cdot z \equiv 0\ (mod\, 2)$, o lo que es lo mismo, una ecuación de congurencias para s.

5. Ahora lo único que queda es repetir los pasos 1 a 4, suponiendo que nos encontramos en el caso 2, hasta que se obtiene la suficiente información para saber s unívocamente. Cada iteración se obtiene una ecuación de congruencias

$$s \cdot z_1 \equiv 0 \quad (mod\, 2), \qquad s \cdot z_2 \equiv 0 \quad (mod\, 2) \quad ...$$

Solo se necesita seguir hasta las únicas posibles soluciones sean 0^n y s. Esto se alcanza tras $\mathcal{O}(n)$ repeticiones.

Para ver la mejoría de un algoritmo cuántico, la solución a este problema en su versión clásica utiliza $\Theta(\sqrt{2^n})$ consultas para obtener la solución. [3]

Inspirado por este algoritmo, Kitaev concluyó que $HSP(G)$ puede ser resuelto en tiempo polinomial para cualquier grupo abeliano[3] G. [3]

[3]Grupo abeliano = Grupo conmutativo[5]

Teorema 6.2. *[3] Para cualquier grupo finito G, $HSP(G) \in BQP^f$*

Inspirados en el éxito en grupos abelianos, se ha intentado diseñar algoritmos cuánticos que permitan resolver el problema para algunos grupos no abelianos. Desafortunadamente, esta búsqueda ha tenido éxito limitado, y todos los grupos para los que se ha conseguido son grupos cercanos a ser abelianos. [3]

El interés que hay por el HSP reside en que hay gran cantidad de problemas que se pueden reducir a éste. Por ejemplo, el isomorfismo de grafos se puede reducir al HSP sobre el grupo simétrico de orden n, S_n. También el problema de aproximar el vector de red más corto puede reducirse, bajo ciertas suposiciones, al HSP sobre el grupo diédrico de orden N, D_N. Ambos se desarrollarán más adelante. [3]

Estos dos problemas son dos problemas que no se conoce que estén en P, pero tienen fuertes razones teóricas para no estar en NP-completo y por lo tanto pueden estar en una zona intermedia entre P y NP. [3]

La reducción del problema de isomorfismo de grafos al $HSP(S_n)$ es sencilla. Sean dos grafos A y B con n vértices. Sea f una función sobre S_{2n} hacia grafos con $2n$ vértices tal que para cada permutación σ, $f(\sigma)$ es el grafo obtenido al realizar la permutación σ a la unión disjunta de A y B. Determinar si el subgrupo oculto de f puede intercambiar los vértices de A y B responde a si A y B son isomorfos. [3]

Mientras tanto, una red es un conjunto de vectores de $\mathbb{R}^n$ cerrado por combinación lineal entera. El objetivo del problema de aproximar el vector de red más corto es encontrar un vector no nulo de una red L que sea, como máximo, un factor $\sqrt{n}$ más largo que el vector más corto no nulo en L. Este problema está estrechamente relacionado con $HSP(D_n)$ y encontrar un algoritmo cuántico para este problema rompería los esquemas de la criptografía de clave pública. [3]

¿Por qué podría esperarse un algortimo polinomial para resolver HSP

en grupos no abelianos? Desde la perspectiva de la complijidad de consulta (modelo de cajas negras) la computación cuántica es suficientemente poderosa. De hecho, utilizando $\mathcal{O}(log^2|G|)$ consultas cualquier algoritmo cuántico puede extraer suficiente información para resolver $HSP(G)$. Este algoritmo toma tiempo exponecial en el proceso posterior a las consultas a f, pero no requiere ninguna consulta más.[3]

Teorema 6.3. *[3] Para todo grupo G, $HSP(G)$ se puede resolver con $\mathcal{O}(log^2|G|)$ consultas cuánticas a f.*

6.1. Complejidad de circuitos.

Definición 6.1. *Dado una transformación unitaria de n qubits U, se denota $\mathcal{C}_\epsilon(U)$ al mínimo tamaño necesario de un circuito cuántico (sobre, por ejemplo, $\{H, P, Toffoli\}$) que implementa U con una precisión de entrada ϵ (por ejemplo $\epsilon = 2^{-n}$).*

También se puede definir la complejidad cuántica de circuitos de transformaciones unitarias relativa a un oráculo:

Definición 6.2. *Se denota $\mathcal{C}_\epsilon^A(U)$ al mínimo tamaño de un circuito cuántico con puertas de caja negra A que aproxima U con una precisión de ϵ.*

El problema de síntesis unitaria. *¿Es verdad que para cualquier transformación unitaria de n qubits U existe un oráculo A tal que $\mathcal{C}_\epsilon^A(U) \leq n^{\mathcal{O}(1)}$?*

Este problema apareció por primera vez en un paper de 2006 de Aaronson y Kuperberg [7]. Conjeturaron que su respuesta era no: es decir, existen transformaciones unitarias que no pueden ser implementadas en BQP^A. Sin embargo, lo mejor que consiguieron probar en esa dirección fue lo siguiente:

Teorema 6.4. *[7] Existen transformaciones unitarias de n qubits U tales que, para todos los oráculos A, un algoritmo de tiempo polinomial no puede*

implementar U con solo una consulta a A, asumiendo que el algoritmo debe implementar alguna matriz unitaria sobre los n qubits en cuestión independientemente de lo que A produzca.

Notar que, pese a que $C_\epsilon(U)$ y $C_\epsilon^A(U)$ son magnitudes de tamaño y por ende de complejidad espacial, tiene sentido hablar de complejidad temporal. Al fijar la dimensión de la transformación U, de n qubits, está fijo también el número de puertas posibles en cada instante, dependiente de n. En el caso del conjunto mencionado, $\{H, P, Toffoli\}$, al ser puertas de dos qubits, es posible colocar como máximo $\frac{n}{2}$ puertas en cada instante. Por lo tanto, hablar de profundidad está directamente relacionado con el tiempo de ejecución.

6.2. Complejidad de estado cuántico

Está sección tratará sobre la complejidad de preparar un estado cuántico, en contraste con implementar una transformación unitaria.

Definición 6.3. *[3] Dado un estado de n qubits $|\psi\rangle$, se denomina $\mathcal{C}_\epsilon(|\psi\rangle)$ al mínimo tamaño de un circuito cuántico (sobre el conjunto $\{H, P, Toffoli\}$) que mapea $|0\rangle^{\otimes m}$ para algún $m > n$ a algún estado ρ tal que*

$$||\rho - |\psi\rangle\langle\psi| \otimes |0...0\rangle\langle 0...0|\,||_{tr} \leq \epsilon \tag{6.1}$$

donde $||\rho||_{tr} = Tr(\sqrt{\rho})$ para cualquier estado ρ es la distancia de traza. [3]

Si se permite basura en los qubits auxiliares (y solo se considera la distancia de traza entre los primeros n qubits de ρ y $|\psi\rangle\langle\psi|$), entonces la medida se denota $\mathcal{C}_\epsilon^*(|\psi\rangle)$. [3]

Para circuitos generales, no es sabido si hay separación $\mathcal{C}$ y $\mathcal{C}^*$ o es siempre posible quitar la basura. Es posible quitar la basura en situaciones como esta:

$$|x\rangle\,|basura(f(x))\rangle\,|f(x)\rangle$$

haciendo una copia del bit $f(x)$

$$|x\rangle\,|basura(f(x))\rangle\,|f(x)\rangle\,|f(x)\rangle$$

y deshaciendo la computación de f para conseguir $|x\rangle\,|0...0\rangle\,|0\rangle\,|f(x)\rangle$. Sin embargo, en el caso de computar un estado cuántico $|\psi_x\rangle$:

$$|x\rangle\,|basura(\psi_x)\rangle\,|\psi_x\rangle$$

no se sabe como quitar la basura en tanto que, debido al Teorema de no Clonación 2.4, no se puede hacer una copia del estado $|\psi_x\rangle$. [3]

Observación 3. *[3] Al igual que en la complejidad de circuitos, se puede probar con un argumento de conteo que casi todos los estados de n-qubits $|\psi\rangle$ satisfacen $\mathcal{C}_\epsilon(|\psi\rangle) = 2^{\Omega(n)}$.*

Observación 4. *[3] Cualquier estado con alta complejidad debe estar entrelazado. Un estado separable de n qubits $|\psi\rangle$ cumple que $\mathcal{C}_\epsilon(|\psi\rangle) = \mathcal{O}(n)$. El contrario de esta afirmación es falso, complejidad y entrelazamiento no son lo mismo. Por ejemplo, el estado*

$$\left(\frac{|00\rangle + |11\rangle}{\sqrt{2}}\right)^{\otimes n} \tag{6.2}$$

tiene entrelazamiento máximo pero su complejidad es trivial (sólo $\mathcal{O}(n)$).

Proposición 6.5. *[3] Si $\langle\psi|\phi\rangle = 0$ entonces*

$$\mathcal{C}(\alpha\,|\psi\rangle + \beta\,|\phi\rangle) \leq \mathcal{O}(\mathcal{C}(|\psi\rangle) + \mathcal{C}(|\phi\rangle) + n) \tag{6.3}$$

Demostración. [3] Se puede preparar fácilmente

$$\alpha\,|0\rangle\,|\psi\rangle + \beta\,|1\rangle\,|\phi\rangle \tag{6.4}$$

(Se están ignorando los qubits auxiliares). Sean U, V los operadores unitarios usados para construir $|\psi\rangle$ y $|\phi\rangle$ respectivamente. El estado anterior es

$$\alpha\,|0\rangle\,U\,|0\rangle^{\otimes n} + \beta\,|1\rangle\,V\,|0\rangle^{\otimes n} \tag{6.5}$$

Aplicando el operador U^{-1}

$$\alpha \left|0\right\rangle \left|0\right\rangle^{\otimes n} + \beta \left|1\right\rangle U^{-1}V \left|0\right\rangle^{\otimes n} \tag{6.6}$$

De acuerdo con el supuesto

$$0 = \langle\psi|\phi\rangle = \langle 0|^{\otimes n} U^{\dagger}V \left|0\right\rangle^{\otimes n} = \langle 0|^{\otimes n} U^{-1}V \left|0\right\rangle^{\otimes n} \tag{6.7}$$

Por tanto la amplitud de $\left|0\right\rangle^{\otimes n}$ en $U^{-1}V \left|0\right\rangle^{\otimes n}$ es 0. Sea W el operador que implementa la función reversible

$$(a_0, a_1, ..., a_n) \longrightarrow (a_0 \oplus OR(a_1, ..., a_n), a_1, ..., a_n) \tag{6.8}$$

Aplicando W a 6.6 se obtiene

$$\alpha \left|0\right\rangle \left|0\right\rangle^{\otimes n} + \beta \left|0\right\rangle U^{-1}V \left|0\right\rangle^{\otimes n} \tag{6.9}$$

Aplicando U

$$\alpha \left|0\right\rangle U \left|0\right\rangle^{\otimes n} + \beta \left|0\right\rangle V \left|0\right\rangle^{\otimes n} = \alpha \left|0\right\rangle \left|\psi\right\rangle + \beta \left|0\right\rangle \left|\phi\right\rangle \tag{6.10}$$

que es el estado deseado (con un qubit auxiliar $\left|0\right\rangle$). □

6.3. Complejidad de estado vs. complejidad unitaria.

En la sección 6.1 se planteó en el problema de síntesis unitaria para alguna transformación unitaria. ¿Es verdad que para cualquier transformación unitaria de n qubits U existe un oráculo A tal que $\mathcal{C}_{\epsilon}^{A}(U) \leq n^{\mathcal{O}(1)}$? En el escenario de complejidad de estado la respuesta es sí.

Proposición 6.6. *[3] Para todo estado cuántico $\left|\psi\right\rangle$ de n qubits existe una función oráculo $A : \{0,1\}^* \longrightarrow \{0,1\}$ tal que $\mathcal{C}_{\epsilon}^{A}(\left|\psi\right\rangle) \leq n^{\mathcal{O}(1)}$.*

Esta proposición ilustra la diferencia entre complejidad de estado y complejidad de transformaciones unitarias. Alguna gente afirma que ambos deben ser equivalentes debido al llamado isomorfismo de Choi-Jamiolkowski, en el cual, el estado cuántico de máximo entrelazamiento

$$|\psi_U\rangle = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} |i\rangle\, U\, |i\rangle \tag{6.11}$$

codifica toda la información sobre la transformación unitaria U. Sin embargo, el problema es que podría haber formas más sencillas de preparar $|\psi_U\rangle$ que no sea preparar $|i, i\rangle$ y luego aplicar U al segundo registro. Para ilustrar esto, sea $\sigma : \{0,1\}^N \longrightarrow \{0,1\}^N$ una función fácilmente aplicable pero difícil de invertir (como alguna función implementada mediante un circuito no reversible) y sea el estado

$$|\phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle\, |\sigma(x)\rangle \tag{6.12}$$

Este estado es fácil de preparar si el estado $|x\rangle$ se puede preparar fácilmente en el primer registro y luego, condicionado a eso, preparar el segundo en $|\sigma(x)\rangle$. Por el contrario, si es necesario empezar con el estado $\sum_x |x\rangle\, |x\rangle$ y luego mapearlo a $|\phi\rangle$ aplicando una transformación unitaria al segundo registro sin acceder al primero el proceso es más complicado. Se necesitaría poder pasar de manera reversible de $|x\rangle$ a $|\sigma(x)\rangle$ y luego invertir σ.[3]

Capítulo 7

Sombras clásicas

Los parámetros necesarios para describir un sistema cuántico no son alcanzables directamente, sino que es necesario realizar una medida. Una medida informativa puede es destructiva (colapso de la función de onda) y solo proporciona probabilidades (regla de Born). Con el fin de solventar estos problemas, se ha intentado hacer descripciones clásicas como la tomografía de red neuronal, basada en entrenar una red con medidas de sistemas cuánticos o la tomografía del estado cuántico. [8] [9]

Sin embargo, estos enfoques para describir un estado cuántico de manera clásica necesitan que los sistemas cuánticos tengan propiedades adecuadas. En cambio, es posible predecir ciertas carácterísticas en lugar de hacer una completa descripción clásica. En mecánica cuántica, las propiedades son a menudo funciones lineales de la matriz de densidad ρ, como los valores esperados $\{o_i\}$ de las observables $\{O_i\}$:[8]

$$o_i = tr(O_i\rho) \qquad 1 \leq i \leq M \tag{7.1}$$

La tarea de predecir propiedades sin necesariamente caracterizar el estado cuántico recibe el nombre de tomografía de sombras. [8]

Una sombra clásica, S_ρ, es creada aplicando repetidamente un simple proceso: aplicar una transformación unitaria $\rho \mapsto U\rho U^\dagger$, y luego medir to-

dos los qubits en la base computacional. El número de veces que se aplica este proceso es el **tamaño** de la sombra clásica. La transformación U es elegida al azar de una familia de unitarias. Cada familia da lugar a un procedimiento diferente. Se consideran circuitos aleatorios de Clifford de qubits de longitud n y productos tensoriales de circuitos aleatorios de Clifford de qubits únicos.[8]

La familia de Clifford viene dada por las puertas $\{S, H, CNOT\}$. Esta familia no es universal para la computación cuántica, pero es conocida por ser eficientmente simulable por un ordenador clásico. Sin embargo, añadir cualquier puerta que no sea de Clifford a esta familia produce un conjunto aproximadamente universal, es decir, la mayoría de operaciones se podrían llevar a cabo. [10]

Notar que ρ es la matriz de densidad de un sistema cuántico fijo. Para formalizar la definición de sombra clásica:

Definición 7.1 (Medida primitiva)**.** *[8] Se puede aplicar un conjunto de unitarios $\rho \mapsto U\rho U^\dagger$, donde U es elegido al azar de una familia de unitarios $\mathcal{U}$. Después se puede medir el estado rotado en la base computacional $\{|b\rangle : b \in \{0,1\}^n\}$. Además se asume que esta colección es tomográficamente completa, es decir, para cada $\sigma \neq \rho$ existe $U \in \mathcal{U}$ tal que $\langle b| U\sigma U^\dagger |b\rangle \neq \langle b| U\rho U^\dagger |b\rangle$.*

Basado en esta primitiva, se realiza repetidamente un proceso de medición aleatorio: se rota aleatoriamente el estado $\rho \mapsto U\rho U^\dagger$ y se realiza una medida en la base computacional. Después de la medida se aplica la inversa de U al estado de la base resultante. Este proceso colapsa ρ en:[8]

$$U^\dagger \left|\hat{b}\right\rangle \left\langle\hat{b}\right| U \quad \text{donde} \quad Pr[\hat{b} = b] = \langle b| U\rho U^\dagger |b\rangle \quad \text{(Regla de Born)} \quad (7.2)$$

Pese a la notación $\left|\hat{b}\right\rangle$ representa una cadena clásica de n bits, es decir, es un vector de n dimensiones, por lo tanto, la matriz $U^\dagger \left|\hat{b}\right\rangle \left\langle\hat{b}\right| U$ es una matriz $n \times n$ clásica. Por así decirlo, $U^\dagger \left|\hat{b}\right\rangle \left\langle\hat{b}\right| U$ es una *fotografía* o muestra

del estado ρ, es decir, una de las posibilidades de medir el estado cuántico en la base computacional.

Esta muestra aleatoria contiene información sobre ρ en esperanza:[8]

$$\mathbb{E}\left[U^\dagger \left|\hat{b}\right\rangle \left\langle\hat{b}\right| U\right] = \mathbb{E}_{U\sim\mathcal{U}} \sum_{b\in\{0,1\}^n} \langle b| U\rho U^\dagger |b\rangle U^\dagger |b\rangle \langle b| U = \mathcal{M}(\rho) \tag{7.3}$$

Para cualquier familia de unitarios $\mathcal{U}$, esta relación constituye un canal cuántico $\rho \mapsto \mathcal{M}(\rho)$. La completitud tomográfica asumida en la definición 7.1 asegura que $\mathcal{M}$, vista como un mapeo lineal, tenga inversa única y por tanto:[8]

$$\hat{\rho} = \mathcal{M}^{-1}\left(U^\dagger \left|\hat{b}\right\rangle \left\langle\hat{b}\right| U\right) \tag{7.4}$$

Este $\hat{\rho}$ es denominado sombra clásica. Está diseñado para que la esperanza de $\hat{\rho}$ sea el propio ρ, $\mathbb{E}[\hat{\rho}] = \rho$.[8]

7.1. Sombras clásicas usadas para predecir.

Las sombras clásicas son adecuadas para predecir funciones lineales en el estado desconocido ρ de la forma de la ecuación (7.1). Para conseguir el objetivo, se sustituye la sombra clásica por el estado. Como las sombras clásicas son aleatorias, esto produce una variable aleatoria que produce la predicción correcta en esperanza, es decir:[8]

$$\hat{o}_i = tr(O_i\hat{\rho}) \qquad\qquad \mathbb{E}[\hat{o}_i] = tr(O_i\rho) \tag{7.5}$$

Lema 7.1. *[8] Fijado O y sea $\hat{o} = tr(O\hat{\rho})$ donde $\hat{\rho}$ es una sombra clásica. La varianza de $\hat{o}$ viene dada por*

$$Var[\hat{o}] = \mathbb{E}[(\hat{o} - \mathbb{E}[\hat{o}])^2] \leq \left\| O - \frac{tr(O)}{2^n} \right\|_{sombra} \tag{7.6}$$

La norma $\|\cdot\|_{sombra}$ solo depende de la medida primitiva

$$\|O\|_{sombra} = \max_{\sigma:estado} \left(\mathbb{E}_{U\sim\mathcal{U}} \sum_{b\in\{0,1\}^n} (\langle b| U\sigma U^\dagger |b\rangle \langle b| U\mathcal{M}^{-1}(\sigma)U^\dagger |b\rangle)^2 \right)^{1/2} \tag{7.7}$$

Este lema es la base de la predicción fiable de funciones lineales. Una sombra clásica puede predecir cualquier función lineal $o_i = tr(O_i\rho)$ en esperanza, por lo que, tomando varias muestras de estas sombras la convergencia al estado deseado se acelerará. Por lo tanto, se construyen $\hat{\rho_1}, ..., \hat{\rho_N}$ sombras clásicas y sea

$$\hat{o}_i(N, 1) = \frac{1}{N} \sum_{i=1}^{N} tr(O_i\hat{\rho}_i) \tag{7.8}$$

Cada sumando es una sombra clásica independiente con esperanza ρ y varainza acotada según el lema 7.1. El problema de este método es que, para conseguir una probabilidad de fallo δ (tener un outlier[1]) se necesitan $N = \frac{\mathrm{Var}[\hat{o}_i]}{\delta\epsilon^2}$ número de muestras. La dependencia con ϵ, la precisión de la aproximación (cercana a 1), es buena, pero la dependencia con δ (cercana a 0) es particularmente mala. Para intentar evitar este problema se utiliza el concepto de *mediana de medias*. En lugar de utilizar todas las muestras para construir una media, se construyen K medias muestrales independientes y se toma su mediana:[8]

$$\hat{o}_i(N, K) = \text{mediana}\{\hat{o}_i^{(1)}(N, 1), ..., \hat{o}_i^{(K)}(N, 1)\} \tag{7.9}$$

donde

$$\hat{o}_i^{(k)}(N, 1) = \frac{1}{N} \sum_{j=N(k-1)+1}^{Nk} tr(O_i\hat{\rho}_j)$$

para $1 \leq k \leq K$. Esta técnica requiere NK muestras, pero es mucho más robusta respecto a outliers. De hecho, $|\hat{o}_i(N, K) - tr(O_i\rho)| > \epsilon$ si y sólo si más de la mitad de las medias se desvían individualmente más de ϵ, es decir, $\left| \frac{1}{N} \left(\sum_{j=N(k-1)+1}^{Nk} tr(O_i\hat{\rho}_j) \right) - tr(O_i\rho) \right| > \epsilon$. La probabilidad asociada

[1]Un outlier en una muestra es una observación muy lejana con respecto al resto de observaciones. [11]

con tener un outlier decrece exponencialmente con el número de medias muestrales K. Esto es una mejora exponencial sobre la estimación de media muestral en términos de probabilidad de fallo. Esto se recoge en el siguiente resultado.[8]

Teorema 7.2. *[8] Fijada una medida primitiva $\mathcal{U}$, una colección $O_1, ..., O_M$ de operadores hermíticos $2^n \times 2^n$ y los parámetros de precisión $\epsilon, \delta \in [0, 1]$. Sea*

$$K = 2\log\left(\frac{2M}{\delta}\right) \quad y \quad N = \frac{34}{\epsilon^2} \max_{1\leq i\leq M} \left\| O_i - \frac{tr(O_i)}{2^n}\mathbb{I} \right\|^2_{sombra} \tag{7.10}$$

donde $\|\cdot\|_{sombra}$ es la norma definida en (7.7). Entonces una colección de NK sombras clásicas independientes permite predecir con precisión todas las características mediante el método de mediana de medias:

$$|\hat{o}_i(N, K) - tr(O_i\rho)| \leq \epsilon \qquad \forall 1 \leq i \leq M \tag{7.11}$$

con probabilidad al menos $1 - \delta$.

Por lo tanto, el número total de repeticiones de medidas para predecir una colección de M funciones lineales $tr(O_i\rho)$ es:

$$N_{tot} = \mathcal{O}\left(\frac{\log(M)}{\epsilon^2} \max_{1\leq i\leq M} \left\| O_i - \frac{tr(O_i)}{2^n}\mathbb{I} \right\|^2_{sombra}\right) \tag{7.12}$$

Esta complejidad solo crece de forma logarítmica con el número de funciones objetivo M. Además, no depende explícitamente de la dimensión del problema 2^n. Sin embargo si que depende de la medida primitiva a través de la norma $\|\cdot\|_{sombra}$. [8]

Estos límites se han tomado para cualquier medida primitiva. Si se elige la familia de unitarios de la que procede se pueden ajustar estas acotaciones. Un primer ejemplo sería el de medidas aleatorias de Clifford.

Proposición 7.3. *[8] Sea una medida primitiva tomada de un circuito de Clifford, es decir, cada rotación $\rho \mapsto U\rho U^\dagger$ es tomada del circutito de Clifford de n qubits ($Cl(2^n)$). La sombra clásica asociada es*

$$\hat{\rho} = (2^n + 1)U^\dagger \left|\hat{b}\right\rangle \left\langle\hat{b}\right| U + \mathbb{I} \tag{7.13}$$

donde $\hat{b} \in \{0,1\}^n$ *es la salida de la medida en la base computacional (del estado* $U\rho U^\dagger$*). Más aún la norma* $\|\cdot\|_{sombra}$ *está acotada por:*

$$tr(O_0^2) \leq \|O_0\|_{sombra}^2 \leq 3tr(O_0^2) \tag{7.14}$$

para cualquier O_0 *hermítico* $2^2 \times 2^n$ *con traza nula.*

Convirtiendo el operador O en traza nula se tiene que

$$\left\| O - \frac{tr(O)}{2^n} \right\|_{sombra}^2 \leq 3tr(O^2) \tag{7.15}$$

Esto combinado con la equación (7.12) asegura que:

$$N_{Clifford} = \mathcal{O}\left(\frac{log(M)\, \text{máx}_i\, tr(O_i^2)}{\epsilon^2} \right) \tag{7.16}$$

Aquí el número de medidas es independiente de la dimensión 2^n.[8]

Aunque se cree que los circuitos de Cliford son más manejables que circuitos cuánticos generales, siguen conteniendo puertas que entrelazan qubits, como la $CNOT$. Estas puertas son aún desafiantes. Para solventar esto, se puede utilizar las medidas aleatorias de Pauli, en las que se asume que solo se aplican puertas de Clifford a un qubit, es decir, $U = U_1 \otimes ... \otimes U_n \sim \mathcal{U} = Cl(2)^{\otimes n}$. Esto es equivalente a asumir que se puede aplicar medidas de Pauli arbitrarias, es decir, medir cada qubit en las bases X, Y, Z respectivamente. Estas medidas se descomponen fácilmente en productos tensoriales ($U\left|\hat{b}\right\rangle = \bigotimes_{j=1}^{n} U_j \left|b_j\right\rangle$ para $b = (b_1, ..., b_n) \in \{0,1\}^n$).

Proposición 7.4. *Sea una medida primitiva tomada de una medida de la base de Pauli, es decir, cada rotación* $\rho \mapsto U\rho U^\dagger$ *es un producto tensorial* $U_1 \otimes ... \otimes U_n$ *de puertas Clifford de un solo qubit seleccionado aleatoriamente* $U_1, ..., U_n \in Cl(2)$*. La sombra clásica asociada viene dada por:*

$$\hat{\rho} = \bigotimes_{j=1}^{n} \left(3U_j^\dagger \left|\hat{b}_j\right\rangle \left\langle \hat{b}_j \right| U_j - \mathbb{I} \right) \quad \textit{donde } \hat{b} = \hat{b}_1 \otimes ... \otimes \hat{b}_n \quad \textit{y } \hat{b}_1, ..., \hat{b}_n \in \{0,1\}^n \tag{7.17}$$

Más aún la norma definida en (7.7) respeta la localidad. Si O es un operador hermítico actuando sobre un producto tensorial de qubits simples y solo actúa sobre k de ellos, por ejemplo, $O = \tilde{O} \otimes \mathbb{I}^{\otimes(n-k)}$; entonces $\|O\|_{sombra} = \left\|\tilde{O}\right\|_{sombra}$ donde $\left\|\tilde{O}\right\|_{sombra}$ es la misma norma pero para sistemas de k qubits.

Una vez conseguida la forma de la sombra clásica, es interesante ver como es la norma $\|\cdot\|_{sombra}$.

Proposición 7.5. *Sea O una observable actuando solo sobre k qubits como en la proposición 7.4, por ejemplo $O = \tilde{O} \otimes \mathbb{I}^{(n-k)}$. Entonces:*

$$\|O\|^2_{sombra} \leq 4^k \|O\|^2_{\infty} \tag{7.18}$$

donde $\|\cdot\|_{\infty}$ es la norma espectral. Esta misma acotación funciona para $\left\|O - \frac{tr(O)}{2^n}\mathbb{I}\right\|^2 \leq 4^k \|O\|^2_{\infty}$.

Esta proposición combinada con la ecuación (7.12) asegura que

$$N_{Pauli} = \mathcal{O}\left(\frac{log(M)4^k}{\epsilon^2}\right) \tag{7.19}$$

En conclusión, el algoritmo que recoge la mediana de las medias de sombras clásicas y que cumple con las coaracterísticas de complejidad desarrolladas hasta ahora, esquematicamente, sería:

Algoritmo de mediana de medias basado en sombras clásicas

1. Preparar $\mathbf{S}(\rho; N) = [\hat{\rho}_1, ..., \hat{\rho}_N]$ las sombras clásicas.

2. Dividir las sombras en K partes iguales y sea

$$\hat{\rho}_{(k)} = \frac{1}{\lfloor N/K \rfloor} \sum_{i=(k-1)\lfloor N/K \rfloor + 1}^{k\lfloor N/K \rfloor} \hat{\rho}_i \tag{7.20}$$

   para construir los K estimadores.

3. Para cada observable O_i mostrar $\hat{o}_i(N, K) = \text{mediana}\{tr(O_i\hat{\rho}_{(1)}), ..., tr(O_i\hat{\rho}_{(K)})\}$.

Capítulo 8

Conclusiones

En este trabajo de fin de grado se ha explorado el concepto de la complejidad en la computación cuántica desde sus bases, la mecánica cuántica. Para ello, en el capítulo 2 se han introducido los conceptos básicos de la mecánica cuántica, incluyendo la amplitud de probabilidad y la superposición, conceptos claves en la computación cuántica.

En el capítulo 3 se ha hecho una introducción a la computación y la complejidad, explicando y definiendo los circuitos como herramientas para operar con bits. También se han definido las máquinas de Turing y algunas de las clases de complejidad existentes.

Posteriormente en el capítulo 4 se han presentado los principios de la computación cuántica, el qubit, y se ha intentado extrapolar los conceptos de la computación clásica a la cuántica. También se ha establecido la diferencia entre la computación probabilística y la cuántica, fundamentada en las propiedades intrínsecas de la computación cuántica como la superposición.

En el capítulo 5 se han propuesto la solución a ciertos problemas con algoritmos cuánticos, haciendo hincapié en el problema de búsqueda. También se ha intentado buscar límies inferiores en el modelo de cajas negras.

En el capítulo 6 se ha indagado en el problema del subgrupo oculto, investigando un algoritmo cuántico para su solución y comprobando su com-

plejidad. También se ha discutido las complejidades de estado y de circuito.

Por último, en el capítulo 7 se ha introducido el concepto de sombra clásica y se ha estudiado su uso para predecir funciones lineales, examinando un algoritmo y su complejidad.

Bibliografía

[1] M. Le Bellac, Quantum physics, Cambridge University Press, 2011.

[2] P. Kaye, R. Laflamme, M. Mosca, An introduction to quantum computing, OUP Oxford, 2006.

[3] S. Aaronson, A. Bouland, L. Schaeffer, Lecture notes for the 28th mcgill invitational workshop on computational complexity (2016).

[4] E. Bach, J. O. Shallit, Algorithmic number theory: Efficient algorithms, Vol. 1, MIT press, 1996.

[5] F. B. Mora, Introducción a la teoría de grupos, Universidad Autónoma del Estado de Hidalgo, 2004.

[6] P. Koiran, V. Nesme, N. Portier, The quantum query complexity of the abelian hidden subgroup problem, Theoretical computer science 380 (1-2) (2007) 115–126.

[7] S. Aaronson, G. Kuperberg, Quantum versus classical proofs and advice, in: Twenty-Second Annual IEEE Conference on Computational Complexity (CCC'07), IEEE, 2007, pp. 115–128.

[8] H.-Y. Huang, R. Kueng, J. Preskill, Predicting many properties of a quantum system from very few measurements, Nature Physics 16 (10) (2020) 1050–1057.

[9] M. Cramer, M. B. Plenio, S. T. Flammia, R. Somma, D. Gross, S. D. Bartlett, O. Landon-Cardinal, D. Poulin, Y.-K. Liu, Efficient quantum state tomography, Nature communications 1 (1) (2010) 149.

[10] S. Zhou, Z.-C. Yang, A. Hamma, C. Chamon, Single T gate in a Clifford circuit drives transition to universal entanglement spectrum statistics, SciPost Phys. 9 (2020) 087.

[11] D. Ghosh, A. Vogt, Outliers: An evaluation of methodologies, in: Joint statistical meetings, Vol. 12, 2012, pp. 3455–3460.