Sección de Matemáticas
Universidad de La Laguna

Adrián Daniel Pérez Galván

Alternativa al Algoritmo de Grover

Alternative to Grover's Algorithm

Trabajo Fin de Grado
Grado en Matemáticas
La Laguna, Septiembre de 2022

Dirigido por
Pino Teresa Caballero Gil

Pino Teresa Caballero Gil
Ingeniería Informática y de Sistemas
Universidad de La Laguna
38200 La Laguna, Tenerife

Agradecimientos

Quiero agradecer a mi familia y amigos por apoyarme y ayudarme a llegar hasta aquí. Especialmente quiero agradecerle a Aridane, Aitana y Patricia su ayuda a lo largo de toda la carrera, puesto que sin ellos aún seguiría en segundo. También quiero agradecer a los profesores Nacho, Teresa y Evelia por despertar en mi la pasión hacia las matemáticas y por su excelente labor docente. Por último, pero no por ello menos importante, quiero agradecer a mi tutora, Pino, por ayudarme y animarme a perseguir los temas que me interesaban. Sin ella no conocería la computación cuántica y este trabajo no existiría.

Adrián Daniel Pérez Galván
La Laguna, 9 de septiembre de 2022

Resumen · Abstract

Resumen

Este trabajo tiene como objetivo presentar una introducción a la computación cuántica desde cero, incluyendo la motivación de crear esta teoría, la notación y la modelización matemática. También se presentan algunos algoritmos cuánticos con una ligera explicación de su funcionamiento.
Además se realiza una profundización del algoritmo de Grover, en la que se explica cuál es su funcionamiento y qué importancia tiene. Por último, se propone una modificación del algoritmo de Grover que reduce la cantidad de evaluaciones de la puerta Oráculo.

Palabras clave: *Computación Cuántica – Algoritmo de Grover – Teoría de la Complejidad Computacional –*

Abstract

This paper serves as an introduction of quantum computing from scratch, including the inspiration of developing this theory, the notation and the mathematical model. I also show some quantum algorithms and explain their behaviors.
Furthermore, in this paper I dive into the Grover's algorithm, where I give a detail explanation of how it works, while also exposing what is its relevance.
Lastly, I propose a modification in Grover's algortihm that reduces the number of evaluations of the Oracle.

Keywords: *Quantum Computing – Grover's Algorithm – Computational Complexity Theory*

Contenido

1

Introducción

1.1. Motivación

Este Trabajo fue motivado por un intento de usar el algoritmo de Grover para criptoanalizar el sistema de McEliece. En ese intento surgió la idea propuesta en el Capítulo 3. Una vez se vio la posibilidad de realizar una mejora sobre el algoritmo de Grover, se decidió intentar obtener una mejora exponencial.

1.2. Objetivo

Este Trabajo se propone cumplir dos objetivos, uno divulgativo y otro académico.

Uno de los objetivos es realizar una introducción formal y completa a la computación cuántica y al algoritmo de Grover. La intención es que un lector completamente ajeno a la computación cuántica pueda adquirir una buena base mediante este Trabajo, solo necesitando conocimientos previos de matemáticas básica y álgebra lineal.

El otro objetivo es presentar un descripción clara de la idea que motiva este Trabajo, además de demostrar que podría tener futuro. De esta forma, la intención es animar a otras personas a trabajar en esta dirección con la esperanza de llegar a un resultado final.

1.3. Organización de la memoria

Esta memoria está organizada en tres capítulos.

El primero es una introducción a la computación cuántica desde el punto de vista matemático. Comienza con unas nociones sobre física cuántica que motivan todo el desarrollo de esta teoría y, a partir de ese punto, no se vuelve a mencionar la física. A lo largo del capítulo se definen todos los conceptos que conforman la teoría de la computación cuántica, y se van desarrollando acompañando al lector

de forma que, al terminar el capítulo, el lector tenga todas la herramientas que necesita para entender algoritmos cuánticos complejos.

El segundo capítulo es un análisis exhaustivo del algoritmo de Grover, en el que se explica con profundidad todo lo que hace falta saber para entender completamente dicho algoritmo, y así poder trabajar con él. Además, esta explicación se realiza con el objetivo de facilitar la comprensión todo lo posible, tomando la perspectiva de alguien que desarrolla el algoritmo, lo que justifica la motivación de cada paso.

En el último capítulo, una vez el lector está equipado con los conocimientos de los dos capítulos anteriores, se propone la idea que motiva este Trabajo. Esta es una idea para mejorar el algoritmo de Grover y se desarrolla siguiendo la línea de pensamiento original, consiguiéndose avances poco a poco hasta que, finalmente, se demuestra que se puede conseguir la mejora deseada.

2

Introducción a la Computación Cuántica

El campo al que pertenece este Trabajo de Fin de Grado es el de la computación cuántica, que es un área del conocimiento que partiendo de las ciencias de la información se enfoca en las peculiaridades de la física cuántica para abrir con ellas un mundo de posibilidades [1] [2]. Para entender la motivación detrás de usar estas propiedades como las bases de una nueva forma de entender la computación, primero hay que saber cuáles son dichas propiedades.

A continuación se da un repaso divulgativo sobre las dos propiedades cuánticas más importantes para la computación cuántica: la superposición cuántica y el entrelazamiento cuántico.

2.1. Superposición cuántica

Supongamos que tenemos un sistema, por ejemplo una caja de madera cerrada que contiene una pelota. Se sabe que la pelota estará en alguna posición con respecto a la caja, y tendrá alguna velocidad, es decir, la pelota estará en un estado concreto. Es posible que su estado sea posicionada en una de las esquinas de la caja con velocidad cero, o quizás su estado sea posicionada en el centro de la caja, en el aire, con una velocidad vertical porque está rebotando.

Quizás no se sepa exactamente dónde está la pelota, o cuál es su velocidad, pero se puede tener la certeza de que la pelota está en un solo lugar y tiene una sola velocidad. Dicho con otras palabras, la pelota tiene un único estado, que está completamente determinado.

Esa certeza no es algo que se pueda tener con cualquier sistema imaginable. Bajo ciertas circunstancias, algunos sistemas se comportan bajo las leyes de la física cuántica, lo que provoca que el estado de sus componentes no esté determinado. Si el ejemplo de la caja y la pelota siguiera este comportamiento,

el estado de la pelota no estaría determinado. Podría estar simultáneamente en una esquina quieta o rebotando en el centro de la caja.

Es muy importante dejar claro que esta indeterminación no viene de la persona que está interesada en la pelota, sino que es la pelota que no se "decide" entre uno de los dos estados. La propia naturaleza de la pelota es existir en un nuevo estado, uno formado por los dos estados en los que puede estar. Este nuevo estado, llamado superposición cuántica, no es uno en el que la pelota esté posicionada en el punto medio de el centro de la caja y de la esquina, es un estado en el que hay una probabilidad de que la pelota esté en el centro y otra probabilidad de que esté en la esquina. Es solo cuando alguien abre la caja y mira dentro que la pelota se "decide" y "elige" uno de los dos estados, es decir, el estado en superposición colapsa a uno de los dos estados posibles.

Sobre esta propiedad se cimienta toda la computación cuántica, donde la unidad básica de información es el cúbit, que no solo permite dos estados 0 y 1 como el bit clásico, sino que permite una infinidad de estados en superposición de ambos.

2.2. Entrelazamiento cuántico

El entrelazamiento cuántico es un efecto muy destacable de la superposición cuántica. En el ejemplo, la pelota tenía dos posibles posiciones, en la esquina y en el centro de la caja. Además, tenía dos posibles velocidades, quieta y con una velocidad vertical. Supongamos que la pelota está en el estado de superposición de antes, es decir, en superposición entre quieta en la esquina y moviéndose en el centro.

Nótese que, al mirar dentro de la caja, no es posible que la pelota colapse a cualquier combinación de las posibles posiciones con las posibles velocidades. Por ejemplo, la pelota no podría estar quieta en el centro de la caja, o moviéndose en la esquina. La superposición en la que se encuentra la pelota sólo permite los dos estados con los que se define la superposición.

Quizás a primeras no parece muy destacable, pero este caso esconde una propiedad muy interesante. Supongamos que dentro de la caja hay una cámara de fotos que permite sacar una foto al interior de la caja, revelando la posición de la pelota.

Si se saca una foto cuando la pelota está en superposición, la posición de la pelota colapsará a una de las dos posibilidades, y supóngase que colapsa a la

esquina. En este caso, sólo se ha roto la superposición de la posición, pero como la pelota no puede estar en la esquina moviéndose, sé con certeza que la pelota estará quieta. En caso de que pudiera colapsar sólo la velocidad y obtuviese que la pelota está moviéndose, también quedaría determinada la posición, que sería en el centro. Es decir, la posición y la velocidad dependen una de la otra y se afectan mutuamente. Este fenómeno se denomina entrelazamiento cuántico, y se dice que la velocidad y la posición están entrelazadas.

Este fenómeno diferencia la computación cuántica de la computación clásica, puesto que está demostrado que si un algoritmo cuántico no usa esta propiedad puede ser simulado por un algoritmo clásico en tiempo polinomial, [3] [4].

2.3. Modelización matemática de los cúbits

En esta sección se introducen los conceptos y definiciones más importantes para la modelización de los cúbits. Esta modelización se basa en el álgebra lineal, lo que proporciona dos ventajas. La primera es que en el álgebra lineal se encuentran todas las herramientas que se necesitan para poder realizar una modelización completa. La segunda es que, debido a que el álgebra lineal está tan bien estudiada, servirá como unos cimientos muy sólidos sobre los que construir la teoría de la computación cuántica.

Para definir la unidad básica de información en computación cuántica, el cúbit, se toma inspiración del bit clásico. En la computación clásica, la unidad básica de información es el bit, que tiene dos estados 0 y 1.

En la computación cuántica se define el cúbit como un objeto que puede tener dos estados, de forma similar al bit clásico, pero además, es posible que el cúbit también exista en un estado de superposición entre ambos, permitiendo que tenga comportamientos según las leyes de la física cuántica. Para denotar los estados equivalentes a 0 y 1 del cúbit usaremos la notación $|\cdot\rangle$ de Dirac, quedando los estados del cúbit de la forma $|0\rangle$ y $|1\rangle$, respectivamente.

Así, modelizaremos los estados de un cúbit $|0\rangle$ y $|1\rangle$ como los vectores $\binom{1}{0}$ y $\binom{0}{1}$, respectivamente, que forman una base del espacio vectorial $\mathbb{C}^2$. Tomamos $\mathbb{C}$ como cuerpo puesto que representa mejor los estados de un cúbit en la realidad. Esto nos permite modelizar un estado en superposición como una combinación lineal de $|0\rangle$ y $|1\rangle$.

Definición 2.1. *Definiremos un* ***cúbit*** *como un espacio de Hilbert H complejo bidimensional.*

A los vectores unitarios de un cúbit los denominaremos ***estados****.*

A los vectores de la base canónica $\binom{1}{0}$ y $\binom{0}{1}$, denotados por $|0\rangle$ y $|1\rangle$, los denominaremos ***estados básicos****.*

Diremos que un vector $|x\rangle$ de H es un ***estado de superposición*** *de un cúbit si es una combinación lineal de la forma*

$$|x\rangle = \alpha_0\tbinom{1}{0} + \alpha_1\tbinom{0}{1} \equiv \alpha_0|0\rangle + \alpha_1|1\rangle\,, \quad \alpha_0, \alpha_1 \in \mathbb{C}$$

donde se verifica $|\alpha_0|^2 + |\alpha_1|^2 = 1$.

A α_0 y α_1 los denominaremos ***amplitud*** *de $|0\rangle$ y $|1\rangle$, respectivamente.*

Nótese que todos los espacios de Hilbert H complejos bidimensionales serán isomorfos a $\mathbb{C}^2$.

Con esta notación, observar un cúbit en un estado de superposición $|x\rangle$ lo colapsará al estado $|0\rangle$ con probabilidad $|\alpha_0|^2$ y al estado $|1\rangle$ con probabilidad $|\alpha_1|^2$.

Esta definición se puede extender a n cúbits tomando $\mathbb{C}^{2^n}$ como espacio.

Definición 2.2. *Definiremos un* ***sistema de n cúbits*** *como un espacio de Hilbert H^{2^n} complejo de dimensión 2^n.*

A los vectores unitarios de H^{2^n} los denominaremos ***estados****.*

A los vectores de la base canónica los denominaremos ***estados básicos****.*

Si $a \in \mathbb{N}$ tal que $0 \leq a \leq 2^n - 1$, denotaremos por $|a\rangle$ al vector de la base canónica que tiene el 1 en la posición a, siendo la primera posición el 0. A menos que se indique lo contrario, al usar esta notación consideraremos que a está escrito en binario.

Diremos que un vector $|x\rangle \in H^{2^n}$ es un ***estado es superposición*** *de n cúbits si es una combinación lineal de la forma*

$$\alpha_0 \underbrace{|0\cdots0\rangle}_{n} + \cdots + \alpha_{2^n-1}\underbrace{|1\cdots1\rangle}_{n} \quad \text{con } \alpha_i \in \mathbb{C}$$

tal que $|\alpha_0|^2 + \cdots + |\alpha_{2^n-1}|^2 = 1$.

A la condición que deben verificar los coeficientes α_i la denominaremos ***condición de normalización****.*

Al coeficiente α_i lo denominaremos ***amplitud*** *del estado $|i\rangle$.*

Ejemplo 2.3. Tomando un sistema de 2 cúbits, tenemos que los estados básicos son

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

Y los estados de superposición son $\alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle$, asegurándonos de que cumplen la condición de normalización.

Con esta notación se define el operador $\langle x|$, que será esencial en los siguientes capítulos.

Definición 2.4. *Dado un sistema de n cúbits H^{2^n}, y dado un vector unitario $v \in H^{2^n}$, denotaremos por $\langle v|$ al operador:*

$$\langle v||x\rangle = \langle v, x\rangle, \ \forall |x\rangle \text{ estado de } H^{2^n}. \tag{2.1}$$

Definiremos ahora la operación que nos permitirá relacionar los estados básicos de un sistema de varios cúbits con los de un solo cúbits y poder trabajar con ellos de manera más eficaz.

Definición 2.5. *El **producto de Kronecker** entre $\mathbb{C}^n$ y $\mathbb{C}^m$ es*

$$\begin{aligned}\otimes : \mathbb{C}^n \times \mathbb{C}^n &\longrightarrow \mathbb{C}^{n+m}\\ (u, v) &\longmapsto (u_1 \cdot v, \cdots, u_n \cdot v)\end{aligned}$$

donde $(u_1, \cdots, u_n)$ son las componentes del vector u.

Usando esta operación, se puede expresar cada estado básico $|x\rangle$ de un sistema de n cúbits como el producto de Kronecker de cada cúbit que lo forma, donde cada dígito en la notación de Dirac $|\cdot\rangle$ representa un cúbit.

Proposición 2.6. *Si $|x\rangle$ es un estado básico de un sistema de n cúbits y x_m es el $m-$ésimo dígito binario de x, de izquierda a derecha, entonces*

$$|x\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle.$$

Ejemplo 2.7. Para un sistema de $n = 2$ cúbits, el estado básico $|00\rangle$ verifica:

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

Definición 2.8. *Decimos que un sistema de n cúbits está en* ***estado producto*** *si su estado se puede expresar como el producto de Kronecker de los estados de los cúbits que lo forman. En el caso contrario decimos que están en* ***estado de entrelazamiento*** *o también que el sistema está* ***entrelazado****.*

Ejemplo 2.9. Por la proposición 2.6 tenemos que los estados básicos son estados productos. Un ejemplo de estado no básico que está en estado producto es el siguiente:

$$\frac{1}{2}|00\rangle - \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle = \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right)$$

Un ejemplo de estado de entrelazamiento es el estado:

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}} \quad .$$

A este estado de dos cúbits se le denomina par EPR (por Einstein-Podolsky-Rosen) o estado de Bell.

Otros estados importantes de dos cúbits son los estados :

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

Posiblemente, el estado mas importante en computación cuántica sea el estado de n cúbits denominado **superposición equiprobable**, que se define como:

$$|\alpha\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle.$$

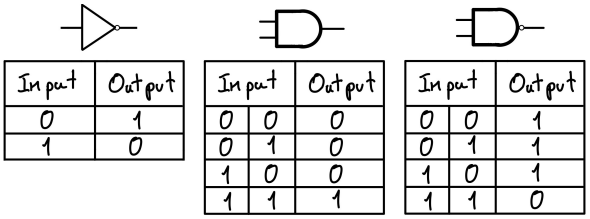
Nótese que $|+\rangle$ es el caso particular de $|\alpha\rangle$ para $n = 1$ cúbit.

2.4. Puertas cuánticas

En computación clásica se manipula la información almacenada en un conjunto de bits usando **puertas lógicas** como la puerta unitaria NOT o la puerta binaria AND.

Una puerta particularmente importante en computación clásica es la puerta NAND, que surge al aplicar una puerta NOT al resultado de una puerta AND.

En la siguiente imagen se ve cómo se representan y cómo actúan las puertas NOT, AND y NAND, respectivamente.



Input	Output
0	1
1	0

Input		Output
0	0	0
0	1	0
1	0	0
1	1	1

Input		Output
0	0	1
0	1	1
1	0	1
1	1	0

La puerta NAND es importante por su propiedad de **universalidad**, que indica que cualquier circuito lógico clásico se puede obtener como concatenación de dichas puertas.

Estas puertas se modelizan como aplicaciones de la forma:

$$\begin{aligned} f : \{0,1\}^n &\longrightarrow \{0,1\}^m \\ (a_1, \cdots, a_n) &\longmapsto f(a_1, \cdots, a_n) \end{aligned}$$

En computación cuántica, se manipula la información almacenada en un conjunto de cúbits usando **puertas cuánticas**. Estas puertas son una generalización de las puertas lógicas clásicas. Para definir cómo actúa una puerta f en el caso de la computación clásica le damos un valor a cada uno de los estados del sistema, que son finitos. En el caso cuántico hace falta dar un valor los estados básicos, que son equivalentes a los estados clásicos, pero también hay que dar valores a los estados de superposición, que son infinitos.

Para lidiar con estos estados, se le exige a las puertas cuánticas que sean lineales, de forma que solo hay que definir el valor de los estados básicos. Es decir, serán funciones lineales de la forma $p : \mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$. Y por lo tanto, las puertas cuánticas quedarán caracterizadas por una matriz A de la forma

$$\begin{aligned} p : \mathbb{C}^{2^n} &\longrightarrow \mathbb{C}^{2^n} \\ x &\longmapsto Ax \end{aligned}$$

Además, si tenemos un estado de un conjunto de cúbits y le aplicamos una puerta p, el resultado tiene que ser un estado válido, es decir, debe verificar la condición de normalización, lo que implica que la matriz A tiene que ser **unitaria**.

Definición 2.10. *Llamaremos puerta cuántica de n cúbits de entrada y m de salida a las aplicaciones de la forma:*

$$\begin{aligned} p : \mathbb{C}^{2^n} &\longrightarrow \mathbb{C}^{2^m} \\ x &\longmapsto Ax \end{aligned}$$

donde A será una matriz unitaria de mxn con coeficientes complejos.

En esta notación, denotaremos por A a la puerta p.

Esto implica que todas las puertas cuánticas son reversibles, en oposición a las puertas clásicas. A pesar de que esta propiedad tiene consecuencias muy interesantes, se salen del objetivo de este Trabajo de Fin de Grado, asé que no las veremos aquí.

A continuación definiremos algunas de las puertas cuánticas más importantes.

Definición 2.11. *El "equivalente" cuántico a la puerta NOT para un cúbit, se llama puerta* X *y está definida por*

$$\begin{aligned} |0\rangle &\longrightarrow |1\rangle \\ |1\rangle &\longrightarrow |0\rangle \end{aligned}$$

caracterizada por la matriz $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

Definición 2.12. *La puerta Z está definida de forma matricial como*

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

Esta puerta cambia el signo de la amplitud del estado $|1\rangle$.

Definición 2.13. *La puerta* ***Hadamard*** *H está definida de forma matricial como*

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

Esta última tiene el efecto de transformar el estado $|0\rangle$ al estado $|+\rangle$ y el estado $|1\rangle$ al estado $|-\rangle$ definidos en (4.1). Esta puerta se utiliza en muchos algoritmos cuánticos, ya que nos permite poner los cúbits en un estado de superposición. De hecho, lo más común es empezar los algoritmos cuánticos en el estado $|0\cdots0\rangle$ y aplicándoles la puerta $H^{\otimes n}$ para que el algoritmo empiece en el estado $|\alpha\rangle$.

También existen puertas cuánticas de varios cúbits. Una de los más importantes es CNOT ("controlled NOT"), cuyo efecto es transformar los estados $|x\ y\rangle$ a $|x\ (y+x)_2\rangle$ donde $x, y \in \{0,1\}$.

Podemos describir su funcionamiento del siguiente modo. El primer cúbit se llama control y el segundo objetivo. Si el control vale 0 el objetivo no cambia y si el control vale 1, negamos el objetivo. En notación matricial:

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

Esta puerta inspira la definición de la siguiente familia de puertas cuánticas:

Definición 2.14. *Sea U una puerta de n cúbits. Denominaremos* ***Controlled-U*** *a una nueva puerta, de $n+1$ cúbits, donde el primer cúbit se denomina* ***control****. Esta puerta actúa de la siguiente forma:*

- *Si el control es 0 deja el sistema en el mismo estado.*
- *Si el control es 1 aplica U al resto de cúbits.*

Ejemplo 2.15. La puerta CNOT es la puerta **Controlled-U** donde $U = Z$.

Una puerta muy importante es la puerta que nos permite observar un cúbit, colapsándolo a uno de los dos estados básicos. Normalmente, tras medir un cúbit no se vuelve a operar con el para que pase a un estado en superposición. Además, como un cúbit en un estado básico es similar a un bit clásico, consideraremos que el cúbit pasará a ser un bit clásico.

Definición 2.16. *La puerta* ***Medida*** *es la puerta que colapsa un cúbit en estado $\alpha_0|0\rangle + \alpha_1|1\rangle$ al estado $|0\rangle$ con probabilidad $|\alpha_0|^2$ y al estado $|1\rangle$ con probabilidad $|\alpha_1|^2$.*

Faltan por ver dos resultados que permiten trabajar cómodamente con las puertas cuánticas. La siguiente proposición indica cómo obtener la puerta cuántica que resulta de aplicar dos puertas cuánticas una detrás de otra. La siguiente permite expresar cuál es la puerta cuántica que resulta de aplicar a un cúbit una puerta cuántica y otra puerta distinta a otro cúbit.

Figura 2.1. Representación de la puerta medida

Proposición 2.17. *Sea $|x\rangle$ un estado de n cúbits y sean A y B dos puertas cuánticas de n cúbits. Entonces, aplicar A a $|x\rangle$ y, después, aplicar B al resultado es equivalente a aplicar a $|x\rangle$ la matriz $C = BA$.*

Proposición 2.18. *Sean $|x_i\rangle$ k estados de n_i cúbits cada uno, y sean A_i k puertas de n_i cúbits cada una. Entonces se verifica que el estado resultante de aplicar A_i a $|x_i\rangle$ es:*

$$(A_1|x_1\rangle) \otimes \cdots \otimes (A_k|x_k\rangle) = (A_1 \otimes \cdots \otimes A_k)(|x_1\rangle \otimes \cdots \otimes |x_n\rangle).$$

2.5. Circuitos cuánticos

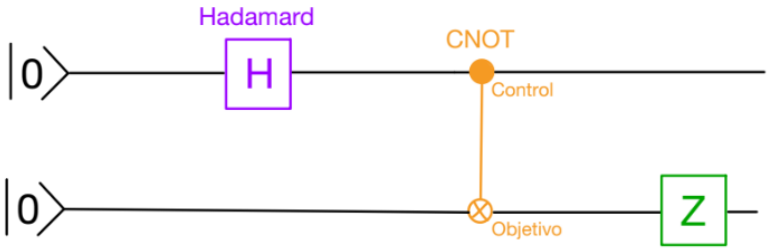
Los circuitos clásicos son concatenaciones de puertas lógicas, que no permiten traducir los algoritmos clásicos a un lenguaje que puedan entender los computadores clásicos.

De forma análoga, los circuitos cuánticos serán concatenaciones de puertas cuánticas. Los representaremos como un diagrama que consta de cables horizontales (uno por cúbit) y se lee de izquierda a derecha. Sobre esos cables colocamos las puertas cuánticas.

En caso de que nuestro sistema incorpore bits clásicos además de cúbits, representaremos los clásicos con un cable doble.

La mejor forma de entender esta representación es con un ejemplo.

Figura 2.2. Ejemplo de circuito cuántico



Ejemplo 2.19. En la imagen 2.19 tenemos representado un circuito de dos cúbits con un estado inicial $|00\rangle$ al que se le aplica puertas cuánticas.

El circuito empieza realizando una puerta Hadamard al primer cúbit, lo que resulta en

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle,$$

después se aplica la puerta binaria, CNOT, que nos da como resultado

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

y finalmente se aplica la puerta unitaria Z al segundo cúbit,

$$\frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle.$$

En resumen, este circuito transforma el estado básico $|00\rangle$ al estado de entrelazamiento

$$\frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle.$$

2.5.1. Teleportación cuántica

Veamos un caso práctico de un circuito cuántico. Este circuito es muy interesante porque demuestra una aplicación de la computación cuántica que no es reproducible en computación clásica.

Supongamos que Alice y Bob comparten un par EPR, es decir dos cúbits en estado $(\sqrt{2})^{-1}(|00\rangle + |11\rangle)$. En esta situación, Alice posee uno de los cúbits mientras que Bob posee el otro. Como la distancia entre los cúbits no es un factor para determinar su estado conjunto, Alice y Bob pueden estar tan separados como quieran que los cúbits seguirán entrelazados.

Supongamos que Alice tiene otro cúbit en estado $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ y necesita enviarle dicho estado a Bob. Bob no sabe cuál es el estado del cúbit que le quiere enviar Alice, y hay que tener cuidado con observar los cúbits puesto que podríamos perder los estados cuando los cúbits colapsen.

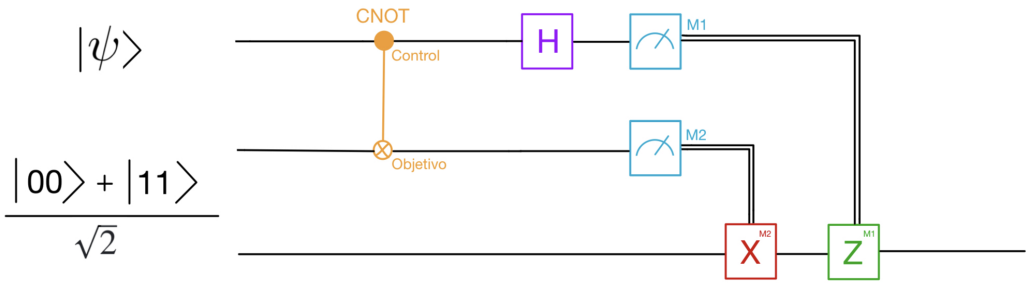
Una solución a este problema sería utilizar **teleportación cuántica**, el proceso para enviar información cuántica a través de un canal clásico. En nuestro caso, esta información cuántica no es nada más que las amplitudes α_0 y α_1 del cúbit que quiere mandar Alice $|\psi\rangle$.

Proceso de teleportación cuántica

Partimos de un estado inicial donde tenemos 3 cúbits, el cúbit que quiere enviarle Alice a Bob $|\psi\rangle$ y el par EPR que comparten Alice y Bob.

El circuito que teletransporta el estado del cúbit $|\phi\rangle$ al cúbit de Bob es el siguiente:

Figura 2.3. Circuito de la teleportación cuántica



Con este circuito cuántico, Bob podrá recibir el cúbit de Alice a través de 2 bits clásicos. Veamos paso a paso lo que sucede en el circuito.

Como el cúbit de Alice $|\psi\rangle$ y el par EPR no están entrelazados, el estado inicial del sistema será:

$$|\psi\rangle \otimes \left(\frac{|00\rangle + |11\rangle}{\sqrt{2}}\right) = \frac{1}{\sqrt{2}}\left(\alpha_0|000\rangle + \alpha_0|011\rangle + \alpha_1|100\rangle + \alpha_1|111\rangle\right).$$

Se aplica la puerta CNOT con control el primer cúbit y objetivo el segundo cúbit,

$$\frac{1}{\sqrt{2}}\left(\alpha_0|000\rangle + \alpha_0|011\rangle + \alpha_1|110\rangle + \alpha_1|101\rangle\right) = \frac{1}{\sqrt{2}}\left(\alpha_0|0\rangle\left(|00\rangle + |11\rangle\right) + \alpha_1|1\rangle\left(|10\rangle + |01\rangle\right)\right)$$

A continuación se aplica una Hadamard al primer cúbit,

$$\frac{1}{\sqrt{2}}\Big(\alpha_0|+\rangle\left(|00\rangle + |11\rangle\right) + \alpha_1|-\rangle\left(|10\rangle + |01\rangle\right)\Big) =$$
$$\frac{1}{\sqrt{2}}\Big(\alpha_0\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(|00\rangle + |11\rangle\right) + \alpha_1\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)\left(|10\rangle + |01\rangle\right)\Big).$$

Operando se puede llegar a al siguiente expresión, donde se agrupa los dos primeros cúbits que son los cúbits de Alice.

$$\frac{1}{2}\Big[|00\rangle\,(\alpha_0|0\rangle+\alpha_1|1\rangle)+|01\rangle\,(\alpha_0|1\rangle+\alpha_1|0\rangle)+|10\rangle\,(\alpha_0|0\rangle-\alpha_1|1\rangle)+|11\rangle\,(\alpha_0|1\rangle-\alpha_1|0\rangle)\Big].$$

A continuación se miden los dos primeros cúbits, los de Alice. Al hacer esto, Alice obtendrá dos bits clásicos que enviará a Bob y además pondrá el tercer cúbit, el de Bob, en uno de los estados de forma que Alice obtiene el estado $|xy\rangle$ el bit de Bob estará en el estado que acompaña a $|xy\rangle$.

Si nos fijamos, el estado del bit de Bob es muy parecido al estado que queríamos transmitir. Además, como Alice sabe en qué estado están sus bits, también sabe cómo estará el cúbit de Bob, de forma que puede decirle a Bob qué debe hacer para corregir su cúbit y obtener el estado objetivo.

La manera de actuar será la siguiente:

- Si el primer bit es 1 se aplica una puerta Z al cúbit de Bob, cambiando el signo de la amplitud de $|1\rangle$. En caso contrario no se aplicará la puerta Z.
- Si el segundo bit es 1 se aplica una puerta X, que intercambiando las amplitudes de $|0\rangle$ y $|1\rangle$. En caso contrario no se aplicará la puerta X.

Así, el cúbit EPR inicial de Bob estará en el mismo estado que el cúbit $|\psi\rangle$ que quería mandar Alice a Bob. Efectivamente, hemos teletransportado el cúbit de Alice.

2.6. Algoritmos cuánticos

2.6.1. Introducción a algoritmos cuánticos

En esta sección vamos a introducir el paralelismo cuántico y algunos algoritmos cuánticos más sencillos como el algoritmo de Deutsch y finalizaremos, nombrando y sin detallar, con el que es posiblemente el algoritmo cuántico más famoso, el algoritmo de Shor.

Antes de empezar, es importante señalar que, a fecha de la publicación de este Trabajo, los ordenadores cuánticos apenas están empezando a desarrollarse y aún hace falta superar muchos obstáculos de ingeniería antes de poder ejecutar algoritmos que sean realmente de utilidad [5] [6].

Para empezar, fijémonos en una propiedad de los computadores cuánticos. Sea f una aplicación de la forma $f(x) : \{0,1\}^n \to \{0,1\}$ y sea U_f su implementación en un computador cuántico.

Si aplicamos U_f al estado $|\alpha\rangle|0\rangle$ obtenemos el estado :

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

que es una superposición de todas las evaluaciones de f. De esta forma evaluamos f en todos los posibles valores de x con una sola aplicación de U_f.

Definición 2.20. *Se define el* ***paralelismo cuántico*** *como la capacidad de un ordenador cuántico para evaluar una función para todos los valores de x de forma simultánea.*

Proposición 2.21. *Dada f de la forma $f(x) : \{0,1\}^2 \to \{0,1\}$ y dado un par de cúbits en estado $|x\ y\rangle$ podemos construir una cadena de puertas cuánticas que realice la transformación $|x\ y\rangle \to |x\ y + f(x)\rangle$. El primer cúbit se llama control y el segundo objetivo. Esta transformación se denota $\bigcup_f$ y es unitaria. Si el objetivo es cero, el estado final será $|x\ f(x)\rangle$.*

Demostración. Vamos a demostrar que $\bigcup_f$ es unitaria. Tenemos cuatro posibles funciones f:

$$f_1(x) = \begin{cases} 0 \text{ si } x = 1 \\ 1 \text{ si } x = 0 \end{cases} \quad f_2(x) = \begin{cases} 1 \text{ si } x = 1 \\ 0 \text{ si } x = 0 \end{cases}$$

$$f_3(x) = 0 \qquad\qquad f_4(x) = 1$$

Las matrices asociadas a la transformación $\bigcup_f$ a cada una de ellas son

$$\bigcup_{f_1} = \begin{pmatrix} 0&1&0&0 \\ 1&0&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{pmatrix} \quad \bigcup_{f_2} = \begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \end{pmatrix}$$

$$\bigcup_{f_3} = \begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{pmatrix} \quad \bigcup_{f_4} = \begin{pmatrix} 0&1&0&0 \\ 1&0&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \end{pmatrix}$$

Como vemos, las cuatro matrices son matrices permutativas, por lo que sabemos que son unitarias.

Además, tenemos el siguiente resultado.

Proposición 2.22. *Dado un circuito clásico para calcular f existe un circuito cuántico de eficiencia comparable que calcula $\bigcup_f$.*

Proposición 2.23. *Dada una transformación de la forma*

$$|x\ 0\rangle \to |x\ f(x)\rangle$$
$$|x\ 1\rangle \to |x\ 1 + f(x)\rangle$$

se tiene que $|x\rangle|-\rangle \to (-1)^{f(x)}|x\rangle|-\rangle$ *donde* $|x\rangle$ *es un estado básico.*

Demostración.

$$|x\rangle|-\rangle = |x\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) = \left(\frac{|x\ 0\rangle - |x\ 1\rangle}{\sqrt{2}}\right) \xrightarrow{\bigcup_f} \left(\frac{|x\ f(x)\rangle - |x\ 1 + f(x)\rangle}{\sqrt{2}}\right)$$

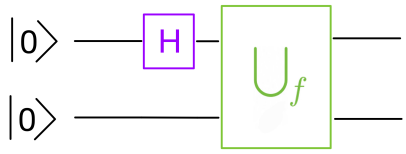
Si $f(x) = 0$,

$$\left(\frac{|x\ 0\rangle - |x\ 1\rangle}{\sqrt{2}}\right) = (-1)^0|x\rangle|-\rangle.$$

Si $f(x) = 1$,

$$\left(\frac{|x\ 1\rangle - |x\ 0\rangle}{\sqrt{2}}\right) = (-1)^1|x\rangle|-\rangle.$$

Ejemplo 2.24. En este circuito cuántico,



es fácil comprobar que obtenemos el estado final

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

En general podemos preparar $n+1$ cúbits en estado $|0\rangle$ y aplicar a los n primeros cúbits $H^{\otimes n}$, esto es, aplicar una puerta Hadamard a cada cúbit siendo n el número de cúbits. Y finalmente aplicando $\bigcup_f$ a los $n+1$ cúbits se obtiene el estado

$$\frac{1}{\sqrt{2^n}}\sum_x |x\ f(x)\rangle.$$

Realizando este procedimiento conseguimos una evaluación simultánea de todos los posibles estados. El problema es que al medir colapsará a uno solo de los estados básicos, obteniendo solo una evaluación. Para que esto no sea un inconveniente tenemos que ser capaces de obtener información de los valores de $f(x)$ a partir del estado de superposición.

2.6.2. Algoritmo de Deutsch y algoritmo de Deutsch-Jozsa

A continuación veremos dos algoritmos cuánticos. Empezaremos por el algoritmo de Deutsch, uno de los más sencillos que trataremos con detalle, y seguiremos con su generalización, el algoritmo de Deutsch-Jozsa.

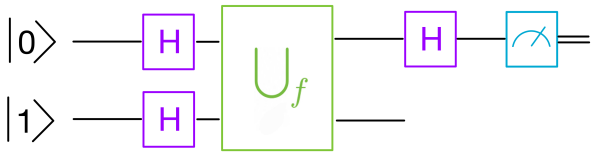
Algoritmo de Deutsch

El problema que resuelve el algoritmo de Deutsch es el siguiente:

Definición 2.25. *Sea* $f(x) : \{0,1\}^2 \to \{0,1\}$ *una aplicación cualquiera. El objetivos es determinar si* $f(0) = f(1)$ *o* $f(0) \neq f(1)$.

En computación clásica, el algoritmo sería parecido a: return($f(0) == f(1)$). Es decir, habría que evaluar la función f dos veces. En el caso cuántico, el algoritmo de Deutsch solo necesita de una evaluación de $\bigcup_f$ para obtener la solución. El circuito es el siguiente:

Figura 2.4. Circuito del algoritmo de Deutsch



Veamos paso a paso el circuito. Primero se aplica $H^{\times 2} = H \times H$ a $|0\ 1\rangle$ y obtenemos

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

A continuación se aplica $\bigcup_f$, utilizando la proposición 2.23 obtenemos

$$\frac{1}{\sqrt{2}}(-1)^{f(0)}|0\rangle|-\rangle + \frac{1}{\sqrt{2}}(-1)^{f(1)}|1\rangle|-\rangle = \frac{1}{\sqrt{2}}\left((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle\right)|-\rangle.$$

Luego se aplica una puerta Hadamard al primer cúbit, quedando el sistema en el estado:

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

esta expresión se puede escribir:

$$\begin{cases} \pm|0\rangle|-\rangle \text{ si } f(0) = f(1) \\ \pm|1\rangle|-\rangle \text{ si } f(0) \neq f(1) \end{cases}$$

Y, puesto que si $f(0) = f(1)$ entonces $f(0) + f(1) = 0$ y si $f(0) \neq f(1)$ entonces $f(0) + f(1) = 1$, la expresión anterior es equivalente a:

$$\pm|f(0) + f(1)\rangle|-\rangle$$

Por último, mediremos el primer cúbit obteniendo $f(0) + f(1)$. Si medimos el valor 0 entonces $f(0) = f(1)$ y si medimos el valor 1 entonces $f(0) \neq f(1)$.

El siguiente algoritmo es una generalización del algoritmo de Deutsch.

Algoritmo de Deutsch-Jozsa

El Algoritmo de Deutsch-Jozsa resuelve el mismo problema pero generalizado a n bits.

Definición 2.26. *Sea $f : \{0,1\}^n \to \{0,1\}$ una aplicación tal que f es constante o f es equilibrada, es decir, f lleva la misma cantidad de elementos a 0 como los que lleva a 1. El objetivo es determinar si f es constante o equilibrada.*

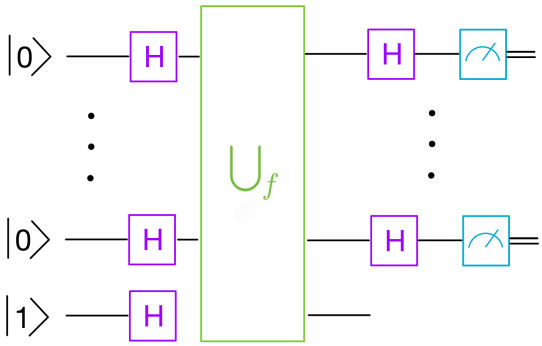
Con un algoritmo clásico, para obtener la solución de forma determinista necesitaremos, como máximo, $2^{n-1} + 1$ evaluaciones. Con el algoritmo de Deutsch-Jozsa solo necesitaremos 1 evaluación de $\bigcup_f$ y n medidas.

Este algoritmo es un ejemplo de un algoritmo cuántico que resuelve un problema de complejidad exponencial a uno polinomial.

El circuito es el siguiente:

Demostrar que este algoritmo funciona se sale del objetivo de este Trabajo de Fin de Grado, pero no es difícil si se usa la demostración del algoritmo de Deutsch.

Figura 2.5. Circuito del algoritmo de Deutsch-Jozsa



3

Algoritmo de Grover

3.1. Planteamiento del problema

El algoritmo objetivo de este Trabajo Fin de Grado es el algoritmo de Grover, desarrollado por Lov K. Grover en 1996, véase [7] [8].

Este algoritmo resuelve el siguiente problema:

Definición 3.1. *Supongamos que tenemos un conjunto A de N elementos y una aplicación $f : A \to \{0, 1\}$ que no sabemos cómo funciona (como si fuera una caja negra). Lo único que sabemos es que existe un único elemento $x_0 \in A$ tal que $f(x_0) = 1$, por lo que $\forall x \in A - \{x_0\},\ f(x) = 0$. El objetivo del problema es encontrar x_0.*

Este problema consiste en encontrar entre un conjunto de candidatos (A) un elemento con una propiedad (f).

Por como está planteado el problema no queda otra opción que ir probando elemento a elemento. En un caso real, donde sí sabemos cómo funciona f, no parece muy inteligente la estrategia de la fuerza bruta. Aún así, muchas veces deducir x_0 a partir de f es todo un desafío, o sencillamente no es computacionalmente eficiente.

Ejemplo 3.2. Un caso particular del problema sería encontrar la única raíz entera del polinomio $p(x) = x^3 + 4x^2 + 2x + 4$ sabiendo que debe estar en el intervalo $[-5, 0]$.

Para poder aplicar el algoritmo en un computador hay que traducir los elementos de A a su lenguaje, por lo que se identificarán los elementos de A con los 2^n estados de un sistema de n bits, con $n \in \mathbb{N}$ tal que $N \leq 2^n$. Además, para simplificar el problema, asumiremos que $N = 2^n$. Estos cambios dan lugar al problema descrito en la Definición 3.1 modificado.

Definición 3.3. *Sean* $f : \{0,1\}^n \to \{0,1\}$ *y* $x_0 \in \{0,1\}^n$ *tal que* $f(x_0) = 1$ *y* $f(x) = 0\ \forall x \neq x_0$. *El objetivo del problema es encontrar* x_0.

En computación clásica este es un problema con una solución muy simple. Tan solo hay que ir tomando elementos de $\{0,1\}^n$ y evaluándolos hasta llegar a x_0. Este algoritmo tiene una complejidad $O(2^n)$, puesto que nunca podrás asegurar que no necesites evaluar todos los elementos.

En computación cuántica el algoritmo de Grover hace uso de la superposición cuántica para evaluar todos los elementos aplicando f solo una vez. Esto permite realizar el algoritmo en $O(2^n)$ aplicaciones de f, con el mismo orden de aplicaciones de puertas básicas.

La idea detrás del algoritmo de Grover es, partiendo de la superposición equiprobable, usar f para "marcar" el estado objetivo $|x_0\rangle$. Una vez "marcado", se aumenta un poco la norma de su amplitud a_0, de forma que se aumenta la probabilidad de que el sistema colapse a $|x_0\rangle$. Este proceso se puede repetir hasta que a_0 sea cercana a 1, dando altas probabilidades de obtener $|x_0\rangle$ al medir.

Como los estados distintos a $|x_0\rangle$ tienen la misma importancia para nosotros, le exigiremos al algoritmo que las amplitudes de todos los estados distintos a $|x_0\rangle$ sean iguales, y las denotaremos por a.

Además, como $||e^{i\theta}z|| = ||z||$, no obtenemos ningún beneficio de trabajar con coeficientes complejos, así que también exigiremos que el algoritmo trabaje solo con coeficientes reales. De esta forma, el estado del sistema en cualquier paso se podrá escribir como:

$$a_0|x_0\rangle + a \sum_{x=0,\ x\neq x_0}^{2^n-1} |x\rangle, \text{ donde } a_0, a \in \mathbb{R} : \ ||a_0||^2 + (2^n - 1)||a|| = 1. \tag{3.1}$$

Puede parecer que exigirle tanto al algoritmo sea ponerse obstáculos en el camino, pero debido a cuál es nuestro objetivo estas condiciones simplificarán los cálculos del algoritmo.

La notación 3.1 inspira el siguiente lema:

Lema 3.4. *Sea*

$$|\varphi\rangle = \frac{1}{\sqrt{2^n - 1}} \sum_{x=0,\ x\neq x_0}^{2^n-1} |x\rangle.$$

Entonces se verifican:

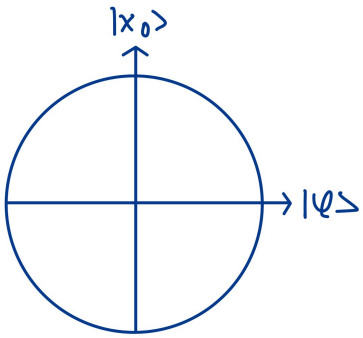
- $|\varphi\rangle$ *es un estado válido de* n *cúbits.*

- $|x_0\rangle$ *y* $|\varphi\rangle$ *son linealmente independientes.*
- *Si* V *es el espacio de Hilbert real con base* $\{|x_0\rangle, |\varphi\rangle\}$ *entonces los estados de la forma 3.1 son los elementos de* V *con norma* 1.
- $|\alpha\rangle \in V$.

Este lema no nos proporciona ninguna herramienta, pero nos enfoca el problema desde una perspectiva ventajosa. Ahora, al trabajar solo con estados de la forma 3.1, pasamos de trabajar en un espacio vectorial complejo de n dimensiones a trabajar en un espacio vectorial real de 2 dimensiones.

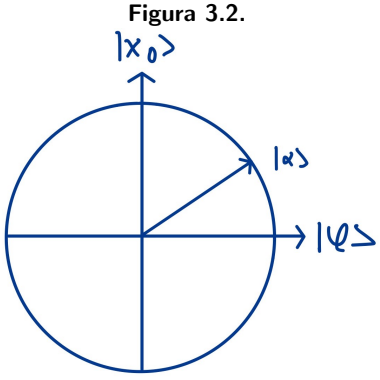
Además, como solo nos interesan los elementos de norma 1 podemos identificar nuestros estados con la circunferencia de radio 1:

Figura 3.1.



Bajo esta perspectiva, $|\alpha\rangle$ se encuentra en

Figura 3.2.



Una vez introducidos 3.1 y la figura 3.1 podemos adentrarnos en el algoritmo.

3.2. Puerta Oráculo

Lo primero es conseguir que f "marque" $|x_0\rangle$. Para ello hay que implementar f de alguna forma. Lo que propone el algoritmo de Grover es encontrar una puerta cuántica, llamada Oráculo y denotada O_f, que se comporte de la siguiente forma:

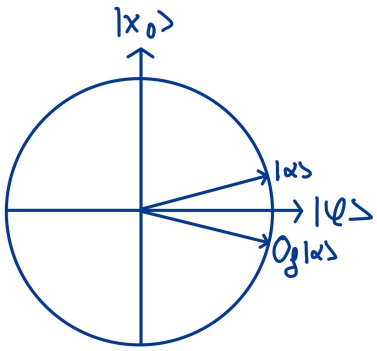
$$O_f|x\rangle = \begin{cases} |x\rangle & \text{si } x \neq x_0 \\ -|x\rangle & \text{si } x = x_0 \end{cases}$$

Nótese que el único efecto que tiene O_f es negar la amplitud del estado $|x_0\rangle$. Esta será nuestra forma de "marcar" $|x_0\rangle$. De esta forma, aplicar O_f a 3.1 resultará en el estado

$$a_0|x_0\rangle + a \sum_{x=0,\ x\neq x_0}^{2^n-1} |x\rangle.$$

En la perspectiva de la figura 3.1, la Oráculo es una simetría respecto al vector $|\varphi\rangle$, que coincide con el eje de abscisas. Es decir, aplicar O_f a $|\alpha\rangle$ daría el resultado:

Figura 3.3.



La aplicación de O_f será sobre la superposición equiprobable y, como veremos más adelante, aumentar la norma de a_0 provocará que a_0 vuelva a ser positiva, por lo que tras cada aumento habrá que volver a aplicar O_f.

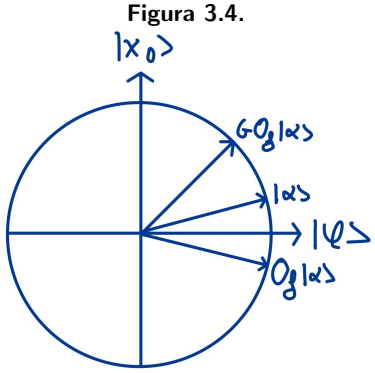
La implementación de f en un computador cuántico puede ser todo un desafío, pero es un desafío al que hay que enfrentarse para poder resolver el problema descrito en la Definición 3.3. Además, esta implementación dependerá de cada caso particular, de forma que no se puede construir sin saber cuál es el problema. Nosotros asumiremos que nuestra O_f existe y la conocemos.

3.3. Inversión sobre la media

La forma que elegiremos para aumentar la norma de a_0 será, bajo la perspectiva de la figura 3.1, realizar una simetría con respecto a $|\alpha\rangle$. Denotaremos por G a la puerta que realice dicha simetría.

Ejemplo 3.5. Si, partiendo de $|\alpha\rangle$, aplicamos O_f y después G obtenemos el estado:

Figura 3.4.



Los pasos del algoritmo que apliquen la puerta G se denominan inversión sobre la medida, y a la puerta G se le suele denominar difusor de Grover, inversión sobre la media o, sencillamente, puerta de Grover.

Como podemos ver, el estado $G\ O_f|\alpha\rangle$ está más cerca del eje de $|x_0\rangle$ que el estado $|\alpha\rangle$, lo que implica que la puerta $G\ O_f$ ha aumentado a_0 a costa de a. Además, como ya había nombrado, aplicar $G\ O_f$ nos ha dejado en un estado en el que $|x_0\rangle$ ya no está "marcado".

Veamos ahora cuál es exactamente el operador G. Justo lo que necesitamos es el siguiente lema, que es un resultado conocido de geometría afín.

Lema 3.6. *Sea H un espacio de Hilbert y $u, v \in H$. Entonces, la reflexión de u con respecto a v será*

$$(R_v)u = 2\langle v, u\rangle v - u.$$

En nuestro caso, u sería un estado $|x\rangle$ cualquiera y v sería el estado $|\alpha\rangle$. La ecuación quedaría:

$$G|x\rangle = 2\langle\alpha, x\rangle|\alpha\rangle - |x\rangle = \left(2|\alpha\rangle\langle\alpha| - I\right)|x\rangle$$

Por tanto, el operador G se definiría como:

$$G = 2|\alpha\rangle\langle\alpha| - I.$$

De forma análoga podemos escribir:

$$O_f = 2|\varphi\rangle\langle\varphi| - I.$$

El siguiente teorema permitirá facilitar la implementación de la puerta G en máquinas cuánticas.

Teorema 3.7. *La siguiente igualdad es cierta.*

$$G = H^{\otimes n}\left(2|0\rangle\langle 0| - I\right)H^{\otimes n}$$

Demostración. Por la propiedad distributiva podemos escribir

$$H^{\otimes n}\left(2|0\rangle\langle 0| - I\right)H^{\otimes n} = 2H^{\otimes n}|0\rangle\langle 0|H^{\otimes n} - H^{\otimes n}IH^{\otimes n}$$

Como H es su propia inversa y $H^{\otimes n}|0\rangle = |\alpha\rangle$ la expresión anterior es igual a

$$2|\alpha\rangle\langle 0|H^{\otimes n} - I$$

Nuestro objetivo es demostrar que dicha expresión es igual a

$$2|\alpha\rangle\langle\alpha| - I.$$

Por tanto, el problema se reduce a demostrar que el operador $\langle\alpha|$ es igual al operador $\langle 0|H^{\otimes n}$. Veamos que operan de la misma forma.

Sea $|x\rangle$ un estado. Entonces:

$$\langle 0|(H^{\otimes n}|x\rangle) = \langle 0, H^{\otimes n}x\rangle = \langle (H^{\otimes n})^* 0, x\rangle = \langle \alpha, x\rangle = \langle \alpha||x\rangle$$

donde $\left(H^{\otimes n}\right)^*$ denota la matriz hermítica de $H^{\otimes n}$.

El tercer miembro es resultado de la propiedad antilineal del producto escalar. El cuarto se debe a que H es su propia hermítica, así que no cambia.

3.4. Pseudocódigo del algoritmo de Grover

Quizás, tras aplicar G por primera vez obtenemos una probabilidad $||\alpha_0||^2$ deseable. Por desgracia, esto no es un caso que se pueda generalizar, así que habrá que aplicar G varias veces, aplicando O_f antes de cada G. Cada aplicación aumentará $||\alpha_0||^2$ hasta que llegue a su máximo, momento en el que dejaremos de aplicar G. Con este último razonamiento estamos preparados para diseñar el

pseudocódigo del algoritmo de Grover.

El primer paso será poner el sistema en el estado $|\alpha\rangle$. Después, se aplicarán las puertas O_f y G, en ese orden, tantas veces como haga falta para maximizar $||\alpha_0||^2$. Una vez esté maximizada, solo habrá que medir todo el sistema, obteniendo el estado objetivo $|x_0\rangle$ con una probabilidad igual a $||\alpha_0||^2$.

Llegados a este punto, solo queda una cosa por aclarar.¿Cuál es la cantidad k de iteraciones que debe realizar el algoritmo para maximizar $||\alpha_0||^2$? Para calcular k haremos uso de los siguientes resultados.

Teorema 3.8. *Sea $|x\rangle$ un estado de la forma 3.1 con $a_0, a > 0$. Además, sean θ el ángulo entre $|\alpha\rangle$ y $|\varphi\rangle$ y w el ángulo entre $|x\rangle$ y $|\varphi\rangle$, donde $\theta, w \in (-\pi, \pi]$.*

Entonces, el ángulo entre $G\,O_f|x\rangle$ y $|\varphi\rangle$ será $w + 2\theta$.

Demostración. Geométricamente, la demostración es trivial.

Corolario 3.9. *Sean $k \in \mathbb{N}$, θ el ángulo entre $|\alpha\rangle$ y $|\varphi\rangle$ y ψ el ángulo entre $(G\,O_f)^k\,|\alpha\rangle$ y $|\varphi\rangle$. Entonces $\psi = (2k+1)\theta$.*

En la notación del corolario 3.9, $\operatorname{sen}\psi$ es la amplitud de $|x_0\rangle$ tras aplicar $G\,O_f$ k veces. Queremos que esa amplitud sea lo más cercana a 1 posible, lo que se traduce a buscar k tal que

$$\psi = (2k+1)\theta = \frac{\pi}{2}$$

Antes de continuar, veamos el siguiente lema.

Lema 3.10. *Sea $|\alpha\rangle$ el estado en superposición equiprobable de un sistema de n cúbits y sea θ el ángulo entre $|\alpha\rangle$ y $|\varphi\rangle$. Entonces se verifica que:*

$$\cos\theta = \langle\alpha, \varphi\rangle = \sqrt{\frac{2^n - 1}{2^n}}.$$

Demostración. Tenemos que

$$|\alpha\rangle = \frac{1}{\sqrt{2^n}}|x_0\rangle + \frac{1}{\sqrt{2^n - 1}} \sum_{x=0,\, x\neq x_0}^{2^n-1} |x\rangle = \frac{1}{\sqrt{2^n - 1}}|x_0\rangle + \frac{\sqrt{2^n - 1}}{\sqrt{2^n}}|\varphi\rangle.$$

Por lo que:

$$\langle\alpha, \varphi\rangle = \frac{1}{\sqrt{2^n - 1}}\langle x_0, \varphi\rangle + \frac{\sqrt{2^n - 1}}{\sqrt{2^n}}\langle\varphi, \varphi\rangle = 0 + \frac{\sqrt{2^n - 1}}{\sqrt{2^n}}1.$$

Como $\cos\theta^2 + \operatorname{sen}\theta^2 = 1$ entonces tenemos que

$$\operatorname{sen}\theta = \sqrt{\frac{1}{2^n}}.$$

Además, para valores grandes de n, θ será pequeño por lo que podemos escribir

$$\operatorname{sen}\sqrt{\frac{1}{2^n}} \cong \sqrt{\frac{1}{2^n}}$$

es decir,

$$\theta = \operatorname{arc\,sen}\sqrt{\frac{1}{2^n}} \cong \sqrt{\frac{1}{2^n}}.$$

Aplicando este resultado a la ecuación 3.4 queda

$$(2k+1)\sqrt{\frac{1}{2^n}} = \frac{\pi}{2}$$

es decir,

$$k = \frac{\pi}{4}\sqrt{2^n} - \frac{1}{2}$$

Para valores grandes de n se puede ignorar el $1/2$ de forma que podemos decir que la cantidad k de iteraciones que hay que realizar para maximizar $||\alpha||^2$ es

$$k \cong \frac{\pi}{4}\sqrt{2^n},$$

lo que implica una complejidad $O(\sqrt{2^n})$. Esto es una mejora cuadrática con respecto al algoritmo clásico.

4

Algoritmo de Grover Mejorado

4.1. Inspiración y propuesta

La idea del funcionamiento de Grover es, partiendo de la superposición equiprobable, identificar el elemento deseado para después aumentar su amplitud. La identificación la realiza la puerta llamada Oráculo, que "marca". El estado deseado cambiando de signo su amplitud, quedando el sistema en el estado:

$$\begin{aligned}&\frac{1}{\sqrt{2^n}}\big(|0\ 0\cdots 0\ 0\rangle + |0\ 0\cdots 0\ 1\rangle + \cdots \\ &\cdots - |a_0\ a_1\cdots a_{n-1}\rangle + \cdots + |1\ 1\cdots 1\ 0\rangle + |1\ 1\cdots 1\ 1\rangle\big)\end{aligned} \tag{4.1}$$

Para el caso de 2 cúbits el algoritmo funciona de forma notable puesto que tras solo una iteración devuelve el valor deseado con probabilidad 1.

Esto propulsa el objetivo de intentar llevar el problema de búsqueda en n cúbits al caso de 2 cúbits. La primera idea sería agrupar los n cúbits en grupos de 2 cúbits. De esta forma surge el inconveniente de que el Oráculo entrelaza los n cúbits, pero para aplicar el algoritmo a cada pareja de cúbits necesitamos que no exista entrelazamiento entre parejas, es decir, que sean independientes entre parejas, pero que cada pareja se mantenga entrelazada con el estado que nos interesa "marcado".

Así, nuestro objetivo es, dado un sistema de n cúbits en superposición equiprobable con el estado de $|a_0\ a_1\cdots a_{n-1}\rangle$ "marcado", encontrar una puerta cuántica que lleve el sistema de ese estado al estado en el que cada pareja de cúbits está preparada para poder aplicar el algoritmo de Grover a cada una, es decir, que el sistema esté en el estado

$$\frac{1}{\sqrt{2^n}}\bigotimes_{i=0}^{\frac{n}{2}-1}\left(-|a_{2i}\ a_{2i-1}\rangle + \sum_{x=0,\ x\neq a_{2i}a_{2i-1}}^{3}|x\rangle\right), \tag{4.2}$$

donde $\otimes$ representa el producto de Kronecker.

Nótese que al aplicar el algoritmo de Grover en la primera pareja de 4.2, que se corresponde con el índice $i = 0$, obtenemos el estado de 2 cúbits $|a_0\ a_1\rangle$ con probabilidad 1, y además se corresponde con los dos primeros bits de nuestro valor deseado. Esto se puede repetir con el resto de parejas para obtener la solución a nuestro problema.

Además, de esta forma sólo se evalúa el Oráculo una vez.

Hay varios caminos que llevan desde 4.1 hasta 4.2, así que elegiré uno de ellos. El camino que he elegido es el primero que se me ocurrió pero ha demostrado tener propiedades interesantes que veremos más adelante.

El camino que tomé es uno que toma m cúbits entrelazados y los divide en dos conjuntos de cúbits entrelazados pero no entre los grupos. Uno de los grupos tendrá entrelazados una cantidad de cúbits igual a la mayor potencia de 2 menor que m. El otro grupo tendrá entrelazados el resto de cúbits.

Ejemplo 4.1. Supongamos que nos encontramos con un conjunto de 12 cúbits entrelazados. Como la mayor potencia de 2 menor de 12 es 8, uno de los grupos tendrá 8 cúbits entrelazados mientras que el otro grupo tendrá $12 - 8 = 4$ cúbits entrelazados.

Este camino presenta una propiedad ventajosa, puesto que las puertas cuánticas que se usen a lo largo del proceso dependerán exclusivamente de la cantidad de cúbits, por lo que no solo no va a haber que tomar decisiones sobre cómo actuar sino que siempre serán las mismas puertas, por lo que solo hay que calcularlas una vez.

Una vez elegida la ruta de actuación, veamos si existe una matriz unitaria para cada m, denotémosla por M_n tal que realice la operación que nos interesa. Para ello, fijémonos primero en el caso más sencillo, el de 4 cúbits.

Nuestro objetivo sería encontrar una matriz M_4 que lleve, por ejemplo, el estado en superposición:

$$\frac{1}{4}\big(-|0\ 0\ 0\ 0\rangle + |0\ 0\ 0\ 1\rangle + |0\ 0\ 1\ 0\rangle + \cdots + |1\ 1\ 1\ 1\rangle\big)$$

al estado en superposición:

$$\frac{1}{4}\big(-|0\ 0\rangle + |0\ 1\rangle + |1\ 0\rangle + |1\ 1\rangle\big) \otimes \frac{1}{4}\big(-|0\ 0\rangle + |0\ 1\rangle + |1\ 0\rangle + |1\ 1\rangle\big),$$

que es equivalente al estado:

$$\begin{aligned}
&|0\ 0\ 0\ 0\rangle - |0\ 0\ 0\ 1\rangle - |0\ 0\ 1\ 0\rangle - |0\ 0\ 1\ 1\rangle \\
-&|0\ 1\ 0\ 0\rangle + |0\ 1\ 0\ 1\rangle + |0\ 1\ 1\ 0\rangle + |0\ 1\ 1\ 1\rangle \\
-&|1\ 0\ 0\ 0\rangle + |1\ 0\ 0\ 1\rangle + |1\ 0\ 1\ 0\rangle + |1\ 0\ 1\ 1\rangle \\
-&|1\ 1\ 0\ 0\rangle + |1\ 1\ 0\ 1\rangle + |1\ 1\ 1\ 0\rangle + |1\ 1\ 1\ 1\rangle
\end{aligned}$$

Nótese la peculiaridad de que, en este ejemplo, el estado objetivo es el $|0\ 0\ 0\ 0\rangle$ y la expresión 4.1 es equivalente a tomar el estado en superposición equiprobable y, considerando la colocación de la expresión 4.1, cambiar de signo toda la primera fila, que es la fila en la que se encuentra el estado $|0\ 0\ 0\ 0\rangle$, y después cambiar de signo toda la primera columna, que es la columna en la que se encuentra el estado $|0\ 0\ 0\ 0\rangle$.

Resulta que esta propiedad no es una coincidencia, sino que es una propiedad que deriva de la similitud del producto de Kronecker con la propiedad distributiva. Esta propiedad es válida no solo para cualquier estado objetivo $|a_0\ a_1\ a_2\ a_3\rangle$, sino también para cualquier cantidad de cúbits m.

Esta propiedad facilitará los cálculos de las matrices M_m.

Volviendo a nuestro caso particular, M_4 debe satisfacer las $2^4 = 16$ ecuaciones:

$$M_4 \begin{pmatrix} -1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} c_{0,0} \\ c_{0,1} \\ c_{0,2} \\ \vdots \\ c_{0,15} \end{pmatrix}, \cdots, M_4 \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ -1 \end{pmatrix} = \begin{pmatrix} c_{15,0} \\ c_{15,1} \\ c_{15,2} \\ \vdots \\ c_{15,15} \end{pmatrix} \tag{4.3}$$

donde $c_{a,b}$ es el coeficiente del estado básico $|b\rangle$ después de aplicar M_4 al estado $\big(|0\rangle + \cdots + |a-1\rangle - |a\rangle + |a+1\rangle + \cdots + |15\rangle\big)$.

Podemos tomar cada ecuación de 4.3 como la ecuación correspondiente a cada columna de la ecuación matricial:

$$M_4 \begin{pmatrix} -1 & 1 & \cdots & 1 \\ 1 & -1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & -1 \end{pmatrix} = \begin{pmatrix} c_{0,0} & c_{1,0} & \cdots & c_{15,0} \\ c_{0,1} & c_{1,1} & \cdots & c_{15,1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{0,15} & c_{1,15} & \cdots & c_{15,15} \end{pmatrix} \tag{4.4}$$

que traspuesta queda

$$\begin{pmatrix} -1 & 1 & \cdots & 1 \\ 1 & -1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & -1 \end{pmatrix} M_4^t = \begin{pmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,15} \\ c_{1,0} & c_{1,1} & \cdots & c_{1,15} \\ \vdots & \vdots & \ddots & \vdots \\ c_{15,0} & c_{15,1} & \cdots & c_{15,15} \end{pmatrix} \tag{4.5}$$

Nótese que las diferentes M_m deberán satisfacer ecuaciones similares así que introduzco la notación A_m y C_m, donde A_m es la matriz que acompaña a M_n será la matriz de coeficientes, de forma que la ecuación 4.5 queda:

$$A_4 \cdot M_4^t = C_4$$

Además, motivados por la propiedad que surge en la ecuación 4.1, podemos definir los coeficientes de C_m para todo m.

Proposición 4.2. *Sean $f_1(a,b)$ y $f_2(a,b)$ dos funciones tal que:*

$$f_1(a,b) = \begin{cases} 1 \text{ si } a \equiv b(mod4) \\ 0 \text{ si } a \not\equiv b(mod4) \end{cases} \qquad f_2(a,b) = \begin{cases} 1 \text{ si } \lfloor a/4 \rfloor = \lfloor b/4 \rfloor \\ 0 \text{ si } \lfloor a/4 \rfloor \neq \lfloor b/4 \rfloor \end{cases}$$

donde $\lfloor x \rfloor$ es la "floor function".

Entonces los coeficientes de C_m son

$$c_{a,b} = (-1)^{f_1(a,b)+f_2(a,b)}.$$

4.2. Demostración de la regularidad de A_m y C_m

Llegados a este punto, si demostramos que A_4 es regular entonces tenemos que existe solución para 4.5 y es única. Si además tenemos que C_4 es regular entonces la solución de 4.5 será regular, por lo que M_4 será una puerta cuántica válida.

Como primer paso, todas las A_m con $m \geq 4$ son regulares puesto que son conocidas matrices cíclicas con determinante no nulo.

Tan solo faltaría demostrar que las diferentes C_m son regulares, pero debido a 4.2 van a satisfacer una ecuación de recursividad que será muy útil para la demostración.

Antes de deducir dicha ecuación, fijémonos en dos propiedades de $c_{a,b}$. Si $a, b \in \mathbb{N}$ tal que $f_2(a,b) = 1$ entonces $\lfloor a/4 \rfloor = \lfloor b/4 \rfloor$, lo que significa que a y b son números cercanos. Siendo más específicos, $\exists k \in \mathbb{N}$ tal que $a, b \in [k, k+4]$.

En nuestro caso esta propiedad se traduce en que en el coeficiente $c_{a,b}$ de C_m, $f_2(a,b) = 1$ si $c_{a,b}$ está cerca de la diagonal. En concreto, se satisface la siguiente propiedad.

Proposición 4.3. *Consideremos una división de C_m en bloques, de tal forma que cada bloque sea de tamaño 4×4. Entonces $c_{a,b}$ verificará que $f_2(a,b) = 1$ sí y solo si $c_{a,b}$ está en la diagonal de bloques.*

Así, si C_m la dividimos en 4 bloques de $m/4 \times m/4$ de la forma:

$$C_m = \begin{pmatrix} C^1 & C^2 \\ C^3 & C^4 \end{pmatrix}$$

f_2 solo afectará en coeficientes de C^1 y C^4, por lo que en C^2 y C^3 solo hay que tener en cuenta f_1, lo que impulsa el siguiente cálculo.

Proposición 4.4. *La siguiente igualdad se verifica:*

$$f_1(a + 4k, b) = f_1(a,b) = f_1(a, b + 4k)\ \forall k \in \mathbb{N}. \tag{4.6}$$

Esta proposición quiere decir que f_1 es una función 4-periódica tanto en a como en b, por lo que en C^2 y C^3 cada 4 filas o columnas se repetirán los coeficientes. El primer coeficiente de C^2 es $c_{0,2^{m-1}}$ y el primero de C^3 es $c_{2^{m-1},0}$. Esto quiere decir que $f_1(0, 2^{m-1}) = f_2(2^{m-1}, 0) = 1$ por lo que $c_{0,2^{m-1}} = c_{2^{m-1},0} = -1$.

Así, podemos afirmar que el comportamiento de f_1 en C^2 y C^3 es el mismo, y como C^2 y C^3 están completamente determinadas por f_1 tenemos la propiedad $C^2 = C^3$.

Con esto ya estamos preparados para deducir la recursividad.

Es fácil ver que para $a, b \in \mathbb{N}$ tal que $0 \leq a, b \leq 2^{m-1} - 1$, el coeficiente $c_{a,b}$ de C_m es igual al coeficiente $c_{a,b}$ de C_{m-1}, por lo que C_{m-1} está contenida en C_m.

Además, se verifican las propiedades

$$f_1(a + 2^{m-1}, b + 2^{m-1}) = f_1(a,b) \tag{4.7}$$
$$f_2(a + 2^{m-1}, b + 2^{m-1}) = f_2(a,b) \tag{4.8}$$

Así que podemos afirmar el siguiente lema:

Lema 4.5. *La matriz C_m se puede expresar de la forma:*

$$C_m = \begin{pmatrix} C_{m-1} & B_{m-1} \\ B_{m-1} & C_{m-1} \end{pmatrix}$$

donde B_m es una matriz por bloques cuadrada donde todos los bloques son:

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

Con esta ecuación recursiva solo nos falta una herramienta para calcular el determinante de C_m, que es el siguiente lema:

Lema 4.6. *Sea M una matriz de la forma:*

$$M = \begin{pmatrix} M_1 & M_2 \\ M_2 & M_1 \end{pmatrix}$$

Entonces

$$det(M) = det(M_1^2 - M_2^2) = det(M_1 - M_2)det(M_1 + M_2).$$

Ahora sí, estamos preparados para demostrar que C_m es regular.

Teorema 4.7. *Las matrices C_m tienen determinante no nulo $\forall m \in \mathbb{N}$.*

Demostración. Gracias a los lemas 4.5 y 4.6 sabemos que

$$det(C_m) = det(C_{m-1} - B_{m-1})det(C_{m-1} + B_{m-1}). \tag{4.9}$$

Por como están definidas todas las matrices C_i podemos escribir

$$C_{m-1} - B_{m-1} = \begin{pmatrix} C_{m-2} & B_{m-2} \\ B_{m-2} & C_{m-2} \end{pmatrix} - \begin{pmatrix} B_{m-2} & B_{m-2} \\ B_{m-2} & B_{m-2} \end{pmatrix} = \begin{pmatrix} C_{m-2} - B_{m-2} & 0 \\ 0 & C_{m-2} - B_{m-2} \end{pmatrix}$$

$$C_{m-1} + B_{m-1} = \begin{pmatrix} C_{m-2} & B_{m-2} \\ B_{m-2} & C_{m-2} \end{pmatrix} - \begin{pmatrix} B_{m-2} & B_{m-2} \\ B_{m-2} & B_{m-2} \end{pmatrix} = \begin{pmatrix} C_{m-2} + B_{m-2} & 2B_{m-2} \\ 2B_{m-2} & C_{m-2} + B_{m-2} \end{pmatrix}$$

de forma que

$$det(C_{m-1} - B_{m-1}) = det^2(C_{m-2} - B_{m-2}) \tag{4.10}$$

$$det(C_{m-1} + B_{m-1}) = det(C_{m-2} - B_{m-2})det(C_{m-2} + 3B_{m-2}) \tag{4.11}$$

Finalmente, realizando este proceso $m-4$ veces, es decir, hasta el caso base de C_4 podemos decir que C_m es regular sí y solo si $\forall n \in [-1, \infty)$, $det(C_4+nB_4) \neq 0$.

Podemos hacer uso de Python para calcular dicho determinante en función de n. Para el caso $n = 0$ no se puede calcular el determinante de forma simbólica, así que lo calcularé aparte. El determinante $det(C_4 + nB_4)$ es un cociente entre dos polinomios de grado muy alto. Tras un rápido estudio de dicho cociente

se comprueba que las únicas raíces deben estar cerca del 0. Como parte final del código se evalúa el determinante en los valores $[-10, 10]$*, comprobando que no se anula en ningún entero.*

El código usado es:

```
import numpy as np
from numpy.linalg import det, inv
import math
import sympy as sym

n = sym.symbols("n")

C_4 = 1/4*np.ones(shape=(16,16),dtype=np.float64)
B_4 = 1/4*np.ones(shape=(16,16),dtype=np.float64)
for i in range(16):
    for j in range(16):
      if i % 4 == j % 4:
        matrix_1[i,j]*=-1
        matrix_2[i,j]*=-1
      if math.floor(i/4)==math.floor(j/4):
        matrix_1[i,j]*=-1

print(det(C_4))

C=sym.Matrix(C_4)
B=sym.Matrix(n*B_4)

f=sym.det(C+B)

for i in range(-10, 10):
  print(f.subs(n, i))
```

Tras comprobar que todos los determinantes calculados son no nulos podemos afirmar que C_m *es regular.*

4.3. Caso con cantidad impar de cúbits

Todos estos cálculos tienen un pequeño fallo, puesto que una cantidad de cúbits n impar no se puede agrupar en parejas. Este fallo no es grave puesto que

se puede arreglar haciendo un ligero cambio al algoritmo.

El cambio consistirá en mantener entrelazados una cantidad impar de cúbits, de forma que el resto sí serán una cantidad par y se les podrá aplicar el algoritmo planteado en este capítulo. Si actuamos así surge la duda de qué hacer con la cantidad de cúbits impar.

Propongo dos alternativas, y dependiendo de cada una variarán la cantidad impar que agruparemos y lo que haremos con ellos.

La primera alternativa es agrupar tres cúbits e ignorarlos. Al realizar el algoritmo acabaremos con 3 dígitos indeterminados de la solución, los correspondientes a los 3 dígitos ignorados. De esta forma tendremos 8 posibles candidatos a solución, con la certeza de que $|x_0\rangle$ se encuentra entre ellos. Tan solo hace falta comprobar los ocho candidatos para encontrar nuestra solución.

La otra alternativa es mas elaborada. En esta elegiremos cinco cúbits, y de entre esos cinco debemos elegir uno en particular, que denotaremos por $|x^*\rangle$ y deberemos hacer dos parejas con los otros cuatro. Estas elecciones se deben mantener constantes a lo largo del todo el proceso. Primero, realizaremos el algoritmo agrupando $|x^*\rangle$ con una pareja, dejando la otra pareja sola. De esta forma sabremos cual es el estado que nos interesa de la pareja aislada. Después volveremos a aplicar el algoritmo pero cambiando los papeles de las parejas. Ahora, agruparemos con $|x^*\rangle$ la pareja que estaba aislada y dejaremos sola a la pareja que antes entrelazamos con $|x^*\rangle$, de forma que ahora obtenemos el estado objetivo de la pareja que nos faltaba.

Tras estas dos iteraciones, conocemos cuáles son todos los dígitos de la solución menos el correspondiente a $|x^*\rangle$, lo que nos deja con dos candidatos como posible solución, siendo la solución uno de ellos. Ahora solo falta comprobar los candidatos para encontrar la solución.

Además, dadas estas dos opciones es fácil saber en que situación es más eficiente una que a otra. La primera opción realiza seis comprobaciones más que la segunda, mientras que la segunda ejecuta el algoritmo una vez más que la primera. Sabiendo esto, si comprobar seis veces si un candidato es una solución o no es más costoso que una aplicación del algoritmo entonces es más eficiente la segunda. Si una aplicación del algoritmo es más costoso que seis comprobaciones, entonces la primera es más eficiente.

Es decir, trabajar con una cantidad impar de cúbits no será un problema puesto que tenemos dos formas de lidiar con la inconveniencia y hasta disponemos de un criterio para elegir la más eficiente.

5

Conclusiones

En este Trabajo de Fin de Grado se ha demostrado que existe una colección de matrices regulares $\{M_n\}$ que, tras aplicarlas como corresponden, transforman el estado 4.1 en el estado 4.2.

Esto indica la existencia de un algoritmo que, en principio, resuelve el problema descrito en la Definición 3.3 realizando una única evaluación de O_f. En este Trabajo no se propone una implementación de las puertas M_n que lo conforman, pero no es descabellado pensar que existe una implementación que no necesita aplicar la puerta Oráculo, lo que podría suponer un incremento en la eficiencia con respecto al algoritmo de Grover.

Además, cabe la posibilidad de que las puertas M_n se puedan implementar en tiempo polinomial con respecto a n. En caso de que esto fuera posible, se podría implementar un algoritmo que, quitando el paso de aplicar el Oráculo y la puerta de Grover G, se ejecute en tiempo $O(p(n))$ siendo n la cantidad de cúbits y p un polinomio, lo que implica tiempo $O(p(log(N)))$ siendo N la cantidad de elementos. Esto supondría un incremento exponencial con respecto al algoritmo de Grover.

5.1. Consecuencias

En caso de que exista alguna de dichas implementaciones del algoritmo, podría aumentarse considerablemente su eficiencia, permitiendo resolver el problema objetivo en un tiempo aún menor. Esto ya es interesante de por sí, pero se vuelve especialmente interesante al dirigir la atención hacia la criptografía.

En caso de que existiera un algoritmo que pudiese invertir una función en tiempo logarítmico, dicho algoritmo se podría usar para romper los sistemas de cifrado de clave pública y clave secreta mas conocidos, dejando mermada toda

la seguridad que se usa hoy en día.

Además, para que un sistema de cifrado resista dicho algoritmo hipotético, encontrar la clave por fuerza bruta necesitaría que fuese del orden $O(2^{2^n})$, es decir, doblemente exponencial. Esto es algo que no se le puede pedir a cualquier sistema.

5.2. Optimalidad del Algoritmo de Grover

Ya se ha demostrado que el algoritmo de Grover es asintóticamente óptimo, como se demuestra en [9] y se explora en [10]. Esto entra en conflicto con el algoritmo diseñado en este Trabajo.

Asumiendo que los cálculos del capítulo 3 sean correctos, y teniendo en cuenta la optimalidad, se saca la conclusión de que en la implementación de las puertas M_n hay que realizar evaluaciones del Oráculo, pero teniendo en cuenta el funcionamiento de dichas puertas, parece que esta afirmación no tiene demasiado sentido.

A falta de suficientes conocimientos para realizar un juicio, o incluso para entender completamente las ramificaciones de este conflicto, no es posible dar una respuesta a este dilema.

5.3. Trabajo futuro

Un posible Trabajo futuro sería intentar resolver dicho conflicto, ya sea de forma teórica o buscando implementaciones de las puertas M_n. Además se podría intentar realizar una implementación de las puertas M_n de menos cúbits, como podría ser M_4. Incluso, se podría definir una base de datos que contuviese muchas de estas matrices, por si fuera de utilidad.

Bibliografía

[1] M.A. Nielsen and I.L. Chuang, *Quantum Computation and Quantum Information*. American Journal of Physics, 70(5), 558. 2002. https://doi.org/10.1119/1.1463744

[2] Nasser Darwish Miranda, *Computación cuántica.* Universidad de La Laguna. 2007. https://www.mundotec.com.ar/Computacion-Cuantica.pdf

[3] Ulises Pastor Díaz, *Algoritmos fundamentales en Computación Cuántica.* Trabajo de Fin de Máster, Facultad de Matemáticas, Universidad de Sevilla. 2019.

[4] María Prieto Gil, *Algoritmos Cuánticos para la Resolución del Problema de Satisfacibilidad Booleana*. Trabajo de Fin de Grado, Escuela Politécnica Superior, Universidad Autónoma de Madrid. 2019.

[5] KamilÁ Valiev, *Quantum computers and quantum computations*. Physics-Uspekhi, 48(1), 1. 2005.

[6] Himar Manuel Barquín Carrasco, *Iniciación a la Computación Cuántica. Algoritmo de Dijkstra Multiobjetivo Cuántico*. Trabajo de Fin de Grado, Escuela Superior de Ingeniería y Tecnología, Universidad de La Laguna. 2021.

[7] Grover L.K., *Quantum Mechanics Helps in Searching for a Needle in a Haystack*. Physical review letters, 79(2), 325. 1997.

[8] María López López, *Implementación del Algoritmo de Grover en los Ordenadores Cuánticos de IBM*. Trabajo de Fin de Grado, Facultad de Ciencias, Universitat dÁlacant. 2020

[9] Charles H. Bennett, Ethan Bernstein, Gilles Brassard and Umesh Vazirani, *Strengths and Weaknesses of Quantum Computing*. SIAM Journal on Computing, 26(5), 1510-1523. 1996.

[10] Arikan, E. *An Information-Theoretic Analysis of Grover's Algorithm.* In: Shumovsky, A.S., Rupasov, V.I. (eds) Quantum Communication and Information Technologies. NATO Science Series, vol 113. Springer, Dordrecht. https://doi.org/10.1007/978-94-010-0171-7_15. (2003)

Lista de símbolos y abreviaciones

$\mathbb{C}_\infty$	Esfera de Riemann
$\partial/\partial\bar{z}$	Operador de Cauchy Riemann $\frac{\partial}{\partial\bar{z}} = \frac{1}{2}\left(\frac{\partial}{\partial x} + i\frac{\partial}{\partial y}\right)$
$\mathbb{T}$	Circunferencia unidad
$C(X, \mathbb{R})$	Funciones reales en X
$C(K)$	Funciones complejas continuas en K
$C_c(X)$	Funciones continuas con soporte compacto en X
$C_0(X)$	Clausura uniforme de $C_c(X)$
Δ	Operador de Laplace $\Delta = 4\partial^2\partial z\partial\bar{z}$ / disco unidad
$\Delta(a, r)$	Disco abierto de centro $a \in \mathbb{C}$ y radio $r > 0$
$L^2_{loc}(\Omega)$	Funciones de cuadrado localmente integrable en Ω
$\mathcal{O}(\Omega)$	Funciones holomorfas en Ω
$\mathcal{O}(K)$	Funciones holomorfas en un entorno de K
$\omega_f(\delta)$	Módulo de continuidad de f
$U \Subset \Omega$	U relativamente compacto en Ω ($\overline{U}$ compacto y $\overline{U} \subset \Omega$)
PL	Problema lineal
PNL	Problema no lineal

Alternative to Grover's Algorithm

Adrián Daniel Pérez Galván
Facultad de Ciencias • Sección de Matemáticas
Universidad de La Laguna
alu0101159980@ull.edu.es

Abstract

This Bachelor Thesis serves as an introduccion of quantum computing and also proposes a modification of Grover's algorithm. The motivation is to make a partition of the qbits to reduce the number of operations and thus, reducing the complexity of the algorithm.

1. Introduction

QUANTUM COMPUTING is a theory that comes from computational science but takes an interest in the capability of some particles to be in separate states at the same time. It makes use of this in quantum bits, that can be 0, 1 or a combination of both.

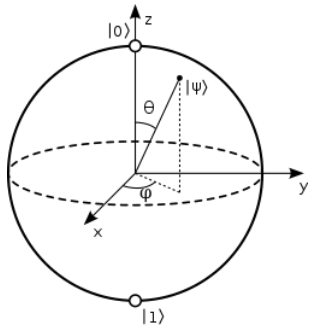


Figure 1: *Representation of a qbit in a Bloch Sphere.*

This property of qbits lets us design algorithms that make things that clasical algorithms are not capable of.

- Because qbits can be in all the posible states at the same time, we can evaluate functions on all the entries at the same time
- Qbits lets us build the first true random algorithm.
- We can make two qbits dependant one to another, which lets us teleport information.

2. Grover's algorithm

GROVER'S ALGORITHM solves the problem of finding the input that, when fed to a certain function, it gives the desired output, effectivly inverting the function.
The algorithm works only with states with real coeficientes, and that are a combination of a state that I will call $|\varphi\rangle$ and the state $|x_0\rangle$, which is our solution to the problem. This way we can represent all the states invloved, such as the uniform superposition $|\alpha\rangle$, in a circuference like the first image of the Figure 2. Then applies a rotation with respect to $|\varphi\rangle$ followed by a rotation with respect to $|\alpha\rangle$ to get closer to the objective state, $|x_0\rangle$.

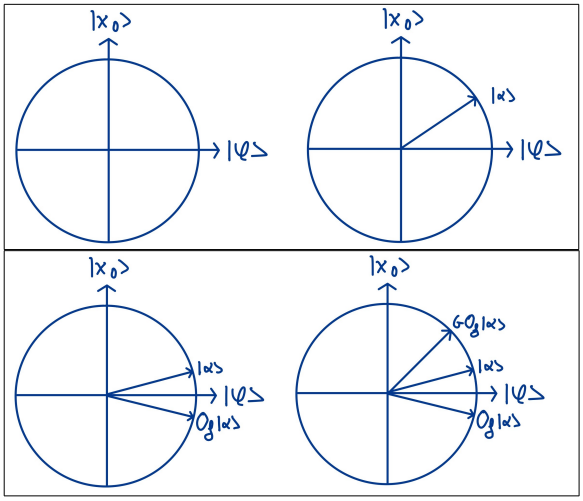


Figure 2: *Representation of the Grover's algorithm in a circumference*

3. Alteration of Grover's algorithm

THE ALTERATION of the algorithm consists of starting in the state $|\alpha\rangle$ and appling a rotation with respect to $|\varphi\rangle$, just like in the original. This starts in the uniform superposition and then changes the sign of the component $|x_0\rangle$, marking the desired state in the set of all qbits. Then, we apply a series of gates to pass down that mark to pairs of qbits, making the state that coincides with the two consecutive digits of the number x_0 in binary that correspond to that pair of qbits. Then, we only need to apply the simetry with respect to $|\alpha\rangle$ once to get the solution $|x_0\rangle$ with probability 1.

References

[1] LAMPORT, Leslie. *LaTeX: A Document Preparation System, User's Guide and Reference Manual.* 2nd edition. Reading: Addison-Wesley, 1994.

[2] *LaTeX* [en línea]. WikiBooks. [Consulta: 17-10-2019] Disponible en: https://en.wikibooks.org/wiki/LaTeX.

[3] WAGNER, C.H. Simpson's Paradox in Real Life. *The American Statistician*, 1982, vol. 36, pp. 46–48.