Computación Cuántica

Juan Pablo Salas
Código: 201821908

Profesor asesor: Gabriel Téllez, PhD.
Profesor revisor: Alejandro García, Dr.
Proyecto teórico computacional

29 de noviembre de 2021
Bogotá, Colombia

Índice

1. Contexto y estado del arte

1.1. Introducción

La mecánica cuántica tuvo avances teóricos y experimentales considerables a inicios del siglo XX. Estos avances permitieron explicar distintos fenómenos a pequeña y gran escala como la estructura atómica, la física del estado sólido, la superconductividad y la fusión nuclear de las estrellas, entre otros. Lo anterior abrió infinidad de posibilidades para aplicar los conceptos de la física cuántica a solventar los problemas de la humanidad atacándolos desde distintas ramas de la ciencia y la tecnología. Uno de estos posibles caminos es el área de la computación cuántica, el cual se originó por primera vez en 1980 [1].

Esta forma de computación consiste esencialmente en utilizar diferentes sistemas cuánticos para crear computadores y así aprovechar los diferentes fenómenos que suceden en este tipo de sistemas como la superposición o el entrelazamiento cuántico. Con esto, nace el concepto de información cuántica que es representado por los distintos estados cuánticos del sistema. Note que estos estados pueden hacer referencia tanto a los niveles energéticos de los átomos, como al espín o la polarización, entre otros [2].

Este proyecto busca entonces describir a profundidad las bases de la implementación física de los computadores cuánticos y cómo esto moldea la implementación de los distintos algoritmos de programación cuántica. Esto se ejemplificará en la investigación de los algoritmos de búsqueda, factorización y de transformación de Fourier. En este punto cabe aclarar que a pesar de los avances que se han dado durante cuatro décadas de computación cuántica, todavía no se tiene una implementación física a gran escala por lo que los algoritmos se pueden evaluar únicamente utilizando un simulador. No obstante, esto igual permite estudiarlos de forma teórica, particularmente en lo que concierne a su complejidad computacional.

Uno de los principales retos de la fabricación de computadores cuánticos reales es la habilidad de controlar diferentes sistemas cuánticos. Para esto, varias alternativas se han estudiado tales como trampas atómicas o iónicas [3], pero aún se observa que hay un largo camino por recorrer hasta tener computadores cuánticos a gran escala.

1.2. Justificación

La importancia de este proyecto recae en que, si bien la computación cuántica sigue en desarrollo, es de gran interés conocer su funcionamiento y alcance a nivel computacional. Es por esto que este proyecto permitirá desarrollar una estructura para la implementación de algoritmos cuánticos y evaluar su complejidad computacional para contrastarlos con su equivalente tradicional. Lo anterior es esencial para seguir motivando el desarrollo activo de la computación cuántica pero también pretende dar luz acerca de la ciencia computacional de este campo.

El interés en la computación cuántica aumentó en los años noventa cuando Peter Shor demostró que ciertos problemas se pueden resolver de forma más eficiente utilizando la computación cuántica [4]. Uno de estos es el algoritmo de factorización de números enteros, el cual es esencial para

la criptografía. Este tema es crucial en la actualidad puesto que para la mayoría de sistemas informáticos la seguridad es decisiva, como por ejemplo en los sistemas financieros.

Por otro lado, la computación cuántica tiene un impacto considerable como herramienta en otras ciencias. Tal es el caso de la biología, pues la programación cuántica permite simplificar los cálculos del descubrimiento de nuevos fármacos y la predicción de la estructura de proteínas [5]. En la química cuántica, también permitiría modelar reacciones a nivel cuántico de forma más precisa y eficiente [6].

1.3. Estado del arte

Para comprender el soporte físico de los computadores cuánticos es esencial comprender los aspectos esenciales de la mecánica cuántica. A un nivel fundamental, los computadores cuánticos están compuestos de *qubits*. Se han propuesto distintas implementaciones físicas de *qubits*, de las cuales se tratará más adelante, pero por ahora se puede pensar en estos como entidades matemáticas abstractas. De esta manera, utilizando la notación de Dirac, un bit cuántico es un sistema físico compuesto con dos estados base $|0\rangle$ y $|1\rangle$. La diferencia fundamental a un computador clásico es que este último está compuesto de *bits*, los cuales pueden admitir un estado único (0 o 1) mientras que debido a la superposición, un *qubit* puede ser una combinación lineal de sus estados base, esto es,

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

, donde α y β son números complejos tal que $|\alpha|^2 + |\beta|^2 = 1$ debido a que sus normas cuadradas representan probabilidades [7]. En los computadores cuánticos, los *qubits* también pueden estar agrupados, lo cual aumenta exponencialmente su capacidad de información. Esto se puede comprender con el hecho de que un sistema cuántico de d dimensiones, en el que los *qubits* pueden tener n estados, puede tener un total de d^n valores clásicos [2].

En un computador clásico, es evidente que la medición de la información de un bit corresponde al estado estático de ese bit. No obstante, mientras que el estado de un *qubit* es una superposición de estados, su medición resulta en un único valor de $|0\rangle$ o $|1\rangle$. Es este punto en el que aparece el concepto de *información cuántica*, el cual se desprende de la concepción clásica que se tiene de información y que la separa de la medición de la misma. Como se desarrollará más adelante, esta es una de las ventajas de los computadores cuánticos y lo que permite su rapidez computacional. Ahora bien, un computador no puede estar únicamente compuesto de *qubits*. Se necesita también circuitos que operen sobre estos valores, los cuales se conocen como compuertas cuánticas.

Las compuertas cuánticas tienen como principal característica que son reversibles, es decir que su entrada se puede deducir de su valor de salida. Los dos modelos más importantes se conocen como *compuertas cuánticas de Toffoli* y *compuertas cuánticas de Fredkin*. Estas son esenciales ya que cualquier circuito cuántico se puede descomponer en estas [2]. Lo anterior se debe a que estas compuertas son reversibles, lo que significa que su estado inicial se pueda recuperar de su estado final [8]. Estos modelos se podrían aplicar al mundo físico por medio de trampas iónicas,

superconductores, diamantes y puntos cuánticos, entre otros. Es así como, en 2009, se logró fabricar una compuerta de Toffoli utilizando trampas de iones [9].

Las trampas de iones consisten en enfriar átomos usando bombeo óptico hasta que llegan a su estado base. Después de esto, el espín nuclear del átomo se puede utilizar para definir un estado cuántico. Este estado se podrá manipular posteriormente utilizando pulsos de láser. No obstante, el inconveniente de esta implementación es la dificultad a la hora de preparar los iones en su estado base a bajas temperaturas.

Otra alternativa de fabricación de circuitos cuánticos es el uso de fotones polarizados. Esta polarización es precisamente la que permitiría leer algún tipo de información cuántica, la cual se podría transformar utilizando retardadores o divisores de haces. Para preparar estos sistemas, se deben crear estados de fotones únicos atenuando la luz de un láser y para medirlos se pueden utilizar detectores foto-multiplicadores.

Una tercera propuesta para los *qubits* consiste en utilizar la propiedad de espín de los núcleos atómicos pero manipularlos con pulsos de campo magnético. Para hacer esto, se debe primero polarizarlos utilizando un campo magnético fuerte. Después de esto, se procesan leyendo el voltaje inducido por el momento magnético de precesión. [3].

En este proyecto se pretende contrastar los principales algoritmos de programación cuántica con los algoritmos de programación clásica en términos de implementación física y complejidad computacional. Esto se hará describriendo las diferencias en la implementación física de los computadores cuánticos y los computadores clásicos y como estas diferencias permiten implementar los algoritmos de formas distintas. Así mismo, se analizarán los principales algoritmos cuánticos como los de búsqueda, transformada de Fourier y factorización. Por último, se evaluará la complejidad computacional de estos algoritmos contrastándola con sus contra partes tradicionales.

2. Metodología

Los objetivos propuestos son de carácter tanto teórico como computacional por lo que se usarán distintas estrategias para llevarlos a cabo.

En primer lugar, se pretende realizar una amplía revisión bibliográfica conducida por dos textos guía: *Programming Quantum Computers* [10] y *Quantum Computing and Quantum Information* [3]. Claramente, lo anterior estará acompañado de lecturas de artículos pertinentes relacionados con los objetivos.

Por otro lado, para la parte computacional se partirá de un simulador de programación cuántica desarrollado por [10]. Este simulador (*QCEngine*) [11] se escogió debido a que este simulador es el utilizado en uno de los principales libros de texto de la bibliografía. Así mismo, permite analizar el código como un circuito cuántico y visualmente como la suma de ciertos estados cuánticos. No obstante, para el trabajo final se usará el software de acceso libre conocido como *Qiskit*, el cual fue desarrollado por la compañía IBM y viene acompañado de una documentación robusta [12]. Este software es utilizado por la mayoría de investigadores de programación cuántica y permite una conjugación sencilla con un ambiente de *Python*.

Para la implementación de los códigos en programación clásica se utilizará el lenguaje *Python* simulado en la nube mediante la herramienta *Google Colab*.

3. Consideraciones éticas

En este proyecto, se hará un estudio sobre distintos algoritmos cuánticos, por lo cual no se hará uso de ningún tipo de dato sobre los cuales puedan haber modificaciones que alteren los resultados del estudio. Así mismo, el autor velará por la correcta citación de las fuentes utilizadas con el fin de mantener la transparencia y evitar cualquier apropiación indebida de conocimiento. Por último, se declara que no existen conflictos de interés de ningún tipo en la realización de este proyecto.

4. Resultados

4.1. Descripción matemática de circuitos cuánticos

Como se ha descrito anteriormente, el estado de un *qubit* puede entenderse como una combinación lineal de los estados $|0\rangle$ y $|1\rangle$ dada por $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$. Esta función de onda también se puede expresar como

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

con el fin de representar gráficamente los parámetros θ y ϕ en una esfera unitaria tridimensional denominada *esfera de Bloch*. Un ejemplo de representación de un qubit en la esfera de Bloch se puede ver en la figura 1. Note además que la base $|0\rangle$ está parametrizada por $\theta = 0$ mientras que $|1\rangle$ está parametrizada por $\theta = \pi$ y $\varphi = 0$. Otro aspecto importante a considerar bajo esta representación es que el coeficiente de $|0\rangle$ siempre será un número real, dado por $\cos\frac{\theta}{2}$. Sin embargo, esto no hace que la representación pierda generalidad puesto que el factor $e^{i\varphi}$ tendrá la información acerca de la diferencia de fase entre $|0\rangle$ y $|1\rangle$.

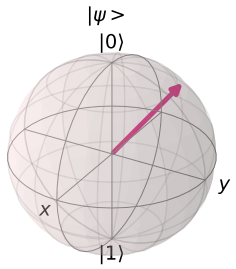


Figura 1: Representación de un qubit en la esfera de Bloch con $\theta = \pi/4$ y $\varphi = 3\pi/4$.

Ahora bien, es importante notar que los coeficientes α y β están asociados a la probabilidad de medir el qubit en estado $|0\rangle$ o en estado $|1\rangle$. Es decir, que al medir el estado de un qubit, este tendrá solamente dos posibilidades. Sin embargo, se pueden construir sistemas con más qubits y aumentar exponencialmente el número de estados. Un sistema con dos qubits estará descrito por cuatro coeficientes, así:

$$|\psi\rangle = \alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle + \alpha_{10}\,|10\rangle + \alpha_{11}\,|11\rangle \tag{3}$$

con lo que se duplica el número de estados posibles, es decir, su dimensión. En general, un sistema con q qubits tendrá dimensión 2^q.

4.1.1. Compuertas cuánticas para un qubit

Ahora bien, para poder ejecutar algoritmos no basta con poder representar los qubits sino que también se deben poder manipular, es decir, forzar a cambiar su estado. Para esto se utilizan

las compuertas cuánticas, las cuales transforman la información de un qubit a otro estado. Por definición, una *compuerta cuántica* es un operador que cambia el estado de un qubit. En este inciso se describirán las principales compuertas cuánticas con una breve descripción de su función, su representación matricial y su representación geométrica en la esfera de Bloch. Para la representación matricial de estas compuertas vale la pena aclarar que un qubit también se puede representar como vector dado por

$$|\psi\rangle = \begin{bmatrix}\alpha\\ \beta\end{bmatrix} = \alpha\,|0\rangle + \beta\,|1\rangle\,.$$

ya que los vectores de la base se pueden escribir como

$$|0\rangle = \begin{bmatrix}1\\ 0\end{bmatrix} \qquad\qquad |1\rangle = \begin{bmatrix}0\\ 1\end{bmatrix}$$

De esta manera, se pueden representar una compuerta cuántica sobre un *qubit* como una matriz M, de dimensión 2x2 tal que

$$M\begin{bmatrix}\alpha\\ \beta\end{bmatrix} = \begin{bmatrix}\alpha'\\ \beta'\end{bmatrix},$$

siempre y cuando M sea unitaria, es decir $M^\dagger M = I$. En el caso más general, para un sistema de q qubits, una compuerta cuántica es un operador cuya representación matricial $M \in \mathsf{M}_{2^q\times 2^q}(\mathbb{C})$. El ejemplo más sencillo de compuerta cuántica está asociado a la matriz identidad y se representa entonces por

$$I = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} = |0\rangle\langle 0| + |1\rangle\langle 1| \tag{4}$$

por lo que su efecto sobre un qubit será,

$$|\psi'\rangle = I\,|\psi\rangle = (|0\rangle\langle 0| + |1\rangle\langle 1|)(\alpha\,|0\rangle + \beta\,|1\rangle) = \alpha\,|0\rangle + \beta\,|1\rangle = |\psi\rangle \tag{5}$$

con lo cual se puede ver que no se modifica el qubit, como su nombre lo indica. Similarmente, se asocian tres compuertas cuánticas a las matrices de Pauli que están asociadas a los operadores de momento angular de espín cuando $s = \frac{1}{2}$ [13]. De esta manera se tiene la compuerta X, que se conoce también como NOT pues invierte la combinación lineal de los estados 0 y 1. Lo anterior se asemeja a la compuerta NOT en computación clásica que invierte el estado de un bit de verdadero a falso o viceversa. Esta se representa por

$$X = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} = |0\rangle\langle 1| + |1\rangle\langle 0| \tag{6}$$

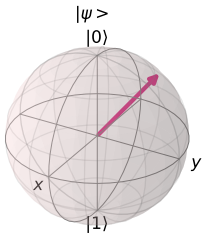
de tal manera que su efecto sobre un qubit será

$$|\psi'\rangle = X\,|\psi\rangle = \beta\,|0\rangle + \alpha\,|1\rangle \tag{7}$$

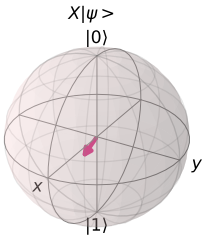
es decir, la inversión de los coeficientes α, β. Del mismo modo, las otras dos matrices de Pauli tendrán compuertas cuánticas asociadas dadas por

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = -i\,|0\rangle\langle 1| + i\,|1\rangle\langle 0| \tag{8}$$

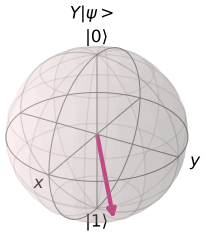
$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = |0\rangle\langle 0| - |1\rangle\langle 1| \tag{9}$$



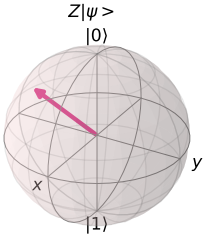
(a) Qubit original $|\psi\rangle$



(b) Efecto de compuerta X



(c) Efecto de compuerta Y



(d) Efecto de compuerta Z

Figura 2: Efectos de las compuertas de Pauli sobre $|\psi\rangle$ visualizados en la esfera de Bloch

En la figura 2, se puede ver el efecto de las compuertas X, Y y Z sobre un qubit $|\psi\rangle$. Note que geométricamente, estas compuertas representan una rotación de π radianes sobre los respectivos ejes x, y, z. De igual manera, vale la pena analizar los vectores propios de estos operadores o compuertas. Para la compuerta X tenemos que sus vectores propios son

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

$$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{11}$$

puesto que $X\,|+\rangle = |+\rangle$ y $X\,|-\rangle = -\,|-\rangle$. Por otro lado, los vectores propios de Y son

$$|\circlearrowleft\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i\,|1\rangle) \tag{12}$$

$$|\circlearrowright\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i\,|1\rangle) \tag{13}$$

puesto que $Y\,|\circlearrowleft\rangle = |\circlearrowleft\rangle$ y $Y\,|\circlearrowright\rangle = -\,|\circlearrowright\rangle$. La ventaja de la compuerta Z es que sus vectores propios son los mismos vectores de la base computacional cuántica, $Z\,|0\rangle = |0\rangle$ y $Z\,|1\rangle = -\,|1\rangle$.

Además de las compuertas asociadas a las matrices de Pauli, se tiene la compuerta de *Hadamard*, fundamental para varios algoritmos. Esta compuerta se puede representar por la matriz

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{\sqrt{2}} \left[|0\rangle \left(\langle 0| + \langle 1| \right) + |1\rangle \left(\langle 0| - \langle 1| \right) \right] \tag{14}$$

, la cual se puede considerar como una rotación de $\pi/2$ radianes alrededor del eje y seguida de una rotación de π radianes alrededor del eje x. Así mismo se puede ver como una combinación de las compuertas X y Z pues $H = \frac{1}{\sqrt{2}}(X + Z)$. Esta compuerta tiene un rol fundamental y es que puede llevar un qubit en estado $|0\rangle$ o $|1\rangle$ a un estado intermedio, tal como se aprecia en la figura 3.

(a) $|0\rangle$ (b) $H|0\rangle = |+\rangle$

(c) $|1\rangle$ (d) $H|1\rangle = |-\rangle$

Figura 3: Efectos de la compuerta Hadamard sobre la base de qubits visualizados en la esfera de Bloch

Se ha visto hasta el momento que las compuertas cuánticas se entienden como una combinación de rotaciones sobre la esfera de Bloch por lo que estas rotaciones se pueden generalizar. De esta manera se define una compuerta parametrizada P, en función de un ángulo de rotación ϕ alrededor del eje z,

$$P(\phi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix} \tag{15}$$

A partir de esta se definen casos específicos de rotación como la compuerta S con $\phi = \pi/2$,

$$S = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{bmatrix} \tag{16}$$

la cual también se conoce como $\sqrt{Z}$ puesto que cumple la propiedad de $SS = Z$. Por último se tiene la compuerta T con $\phi = \pi/4$,

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \tag{17}$$

la cual similarmente se conoce como $\sqrt[4]{Z}$ por que cumple una propiedad similar. La rotación se puede generalizar mediante la compuerta U,

$$U(\theta, \phi, \lambda) = \begin{bmatrix} \cos\frac{\theta}{2} & -e^{i\lambda}\sin\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2} & e^{i(\phi+\lambda)}\cos\frac{\theta}{2} \end{bmatrix} \tag{18}$$

4.1.2. Compuertas cuánticas para múltiples qubits

Como se observó en la ecuación 3, un sistema compuesto por dos qubits es representado por cuatro coeficientes. Más aún, si dos qubits q_1 y q_2 están en los estados

$$|q_1\rangle = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \quad ; \quad |q_2\rangle = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix},$$

el sistema compuesto por estos dos qubits se puede expresar como el producto tensorial de estos dos estados. Es relevante recordar que el producto tensorial de una matriz A de tamaño $m \times n$ con una matriz B de tamaño $p \times q$ está dado por la matriz $mp \times nq$ [14],

$$A \otimes B = \begin{bmatrix} A_{11}B_{11} & \dots & A_{11}B_{1q} & \dots & A_{1n}B_{11} & \dots & A_{1n}B_{1q} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{11}B_{p1} & \dots & A_{11}B_{pq} & \dots & A_{1n}B_{p1} & \dots & A_{1n}B_{pq} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{m1}B_{11} & \dots & A_{m1}B_{1q} & \dots & A_{mn}B_{11} & \dots & A_{mn}B_{1q} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{m1}B_{p1} & \dots & A_{m1}B_{pq} & \dots & A_{mn}B_{p1} & \dots & A_{mn}B_{pq} \end{bmatrix}$$

por lo cual, el producto tensorial de dos qubits q_1 y q_2 será

$$|q_1q_2\rangle = |q_1\rangle \otimes |q_2\rangle = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \otimes \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = \begin{bmatrix} a_0 \times \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \\ a_1 \times \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} a_0b_0 \\ a_0b_1 \\ a_1b_0 \\ a_1b_1 \end{bmatrix}$$

por lo que la base en la que está escrita el vector de estado 3 se puede representar como

$$|00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad ; \quad |01\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad ; \quad |10\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad ; \quad |11\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Como se vio anteriormente, las compuertas cuánticas para un sistema de dos qubits serán matrices de tamaños 4×4. De estas la más importante es la matriz de control no, denotada por

CNOT o U_{CN} y representada por

$$U_{CN} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{19}$$

Esta matriz toma un sistema de dos qubits y si el primer qubit es 1, efectúa una compuerta X sobre el segundo qubit. Es decir que su efecto sobre $|00\rangle$ será nulo pero su efecto sobre $|10\rangle$ será transformar el sistema en $|11\rangle$. Los qubits dobles generan una ventaja significativa sobre los sitemas de qubits únicos y son los estados enredados. Un ejemplo de estos estados son los estados Bell o estados EPR (Einstein, Podoslki y Rosen) que se pueden crear a partir de una compuerta de Hadamard que actúe sobre la base de $|0\rangle$ y $|1\rangle$. Como se vio anteriormente, esta compuerta creará los estados $|+\rangle$ y $|-\rangle$. Después de esto, se pasan los estados $|+0\rangle, |+1\rangle, |-0\rangle, |-1\rangle$ por una compuerta CNOT y se tendrán los estados

$$\begin{aligned} |\beta_{00}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ |\beta_{01}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \\ |\beta_{10}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\ |\beta_{11}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \end{aligned}$$

los cuales son de suma importancia porque son el ejemplo más claro de entrelazamiento cuántico. En particular note que para el estado $|\beta_0 0\rangle$ si al medir el primer qubit se obtiene un estado $|0\rangle$, ya se sabe que el segundo qubit estará en el estado $|0\rangle$ también puesto que los estados $|10\rangle$ y $|01\rangle$ no son posibles. Incluso si estos dos qubits se separan, con solo la medición de uno de ellos se tendrá certeza absoluta del estado del otro [15].

Los sistemas de qubits y sus respectivas compuertas, se pueden generalizar a una dimensión n-ésima. Para esto, se introduce la notación $|q\rangle^{\otimes n}$ para referirse al producto tensorial de $|q\rangle$ consigo mismo realizado n veces. Similarmente, un operador M se puede aplicar n veces a un qubit utilizando la notación $M^{\otimes n}$. Así mismo, se utilizará el qubit $|\mathbf{x}\rangle$ para hacer referencia al vector de la base en dimensión n dado por

$$|\mathbf{x}\rangle = |x_1 x_2 \dots x_n\rangle = |x_1\rangle |x_2\rangle \dots |x_n\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \dots \otimes |x_n\rangle \tag{20}$$

donde $x_i \in \{0,1\} \quad \forall i, 1 \leq i \leq n$. Ahora se procederá a generalizar el efecto de una compuerta Hadamard sobre un sistema de qubits en estado base. Para esto, observe que

$$H|x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^x |1\rangle) = \frac{1}{\sqrt{2}} \sum_{z \in \{0,1\}} (-1)^{xz}$$

Entonces, al operar esta compuerta n veces sobre un sistema de n qubits base, se tendrá [14]

$$\begin{aligned} H^{\otimes n}|\mathbf{x}\rangle &= H|x_1\rangle H|x_2\rangle \dots H|x_n\rangle \\ H^{\otimes n}|\mathbf{x}\rangle &= \frac{1}{\sqrt{2^n}}(|0\rangle + (-1)^{x_1}|1\rangle)(|0\rangle + (-1)^{x_2}|1\rangle)\dots(|0\rangle + (-1)^{x_n}|1\rangle) \\ H^{\otimes n}|\mathbf{x}\rangle &= \frac{1}{\sqrt{2^n}}\sum_{\mathbf{z}\in\{0,1\}^n}(-1)^{\mathbf{x}\cdot\mathbf{z}}|\mathbf{z}\rangle \end{aligned} \tag{21}$$

donde la sumatoria se realiza sobre la potencia cartesiana del conjunto $\{0,1\}$ dada por $\{0,1\}^n = \{y_1 \dots y_n | y_i \in \{0,1\} \forall i, 1 \leq i \leq n\}$. Cabe resaltar que este conjunto tiene cardinalidad 2^n por lo que este será el número de términos en la sumatoria. Observe por ejemplo que al operar la compuerta de Hadamard $H^{\otimes 3}$ sobre un vector $|110\rangle$ se tendrá

$$H^{\otimes 3}|110\rangle = \frac{1}{\sqrt{2^3}}\sum_{\mathbf{z}\in\{0,1\}^3}(-1)^{110\cdot\mathbf{z}}|\mathbf{z}\rangle$$

como la suma se realiza sobre $\{0,1\}^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$, se tiene

$$\begin{aligned} H^{\otimes 3}|110\rangle = \frac{1}{\sqrt{8}}[&(-1)^{110\cdot 000}|000\rangle + (-1)^{110\cdot 001}|001\rangle + (-1)^{110\cdot 010}|010\rangle + (-1)^{110\cdot 011}|011\rangle \\ &+ (-1)^{110\cdot 100}|100\rangle + (-1)^{110\cdot 101}|101\rangle + (-1)^{110\cdot 110}|110\rangle + (-1)^{110\cdot 111}|111\rangle] \end{aligned}$$

por lo que el resultado final será

$$H^{\otimes 3}|110\rangle = \frac{1}{\sqrt{8}}[|000\rangle + |001\rangle - |010\rangle - |011\rangle - |100\rangle - |101\rangle + |110\rangle + |111\rangle]$$

Otro aspecto fundamental de la compuerta generalizada de Hadamard es su inversa. En el caso de un qubit vemos que la representación matricial (dada por 14) de esta compuerta es su propia adjunta puesto que

$$H^\dagger = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = H$$

lo cual, junto al hecho de que los operadores son matrices unitarias $HH^\dagger = I$, demuestra que esta compuerta es su propia inversa. Esto se puede ver igualmente con la definición n-ésima que se da en la ecuación 21. Al volver a aplicar esta compuerta se tiene

$$\begin{aligned} H^{\otimes n}H^{\otimes n}|\mathbf{x}\rangle &= \frac{1}{2^n}\sum_{\mathbf{z}\in\{0,1\}^n}(-1)^{\mathbf{x}\cdot\mathbf{z}}\sum_{\mathbf{y}\in\{0,1\}^n}(-1)^{\mathbf{z}\cdot\mathbf{z}}|\mathbf{y}\rangle \\ &= \frac{1}{2^n}\sum_{\mathbf{y}\in\{0,1\}^n}|\mathbf{y}\rangle\sum_{\mathbf{z}\in\{0,1\}^n}(-1)^{\mathbf{z}\cdot(\mathbf{x}+\mathbf{y})} \end{aligned}$$

ecuación de la cual se desprenden dos casos. En primer lugar se tiene la posibilidad de que $\mathbf{x} = \mathbf{y}$ dentro de la primera sumatoria. Esto implicara

$$\mathbf{x} + \mathbf{y} = 2\mathbf{x} = \mathbf{0} \mod 2$$

Entonces se tendrá

$$\sum_{\mathbf{z}\in\{0,1\}^n}(-1)^{\mathbf{z}\cdot\mathbf{0}}=\sum_{\mathbf{z}\in\{0,1\}^n}1=2^n$$

por otro lado, si $\mathbf{x}\neq\mathbf{y}$ debe existir al menos un x_i y un y_i tal que $x_i\neq y_i$ por lo que se puede partir la suma en este bit para ver

$$\sum_{\mathbf{z}_{-i}\in\{0,1\}^{n-1}}(-1)^{\mathbf{z}_{-i}\cdot(\mathbf{x}_{-i}+\mathbf{y}_{-i})}\sum_{z_i\in\{0,1\}}(-1)^{z_i\cdot(x_i+y_i)}$$

Como x_i es diferente a y_i, se tiene $x_i+y_i=1 \mod 2$ por lo que el segundo término se puede evaluar trivialmente como $(-1)^0+(-1)^1=0$. Todo lo anterior permite ver que la sumatoria se puede escribir como

$$\sum_{\mathbf{z}\in\{0,1\}^n}(-1)^{\mathbf{z}\cdot(\mathbf{x}+\mathbf{y})}=2^n\delta_{\mathbf{xy}} \tag{22}$$

así pues se tiene al aplicar dos veces la compuerta de Hadamard,

$$H^{\otimes n}H^{\otimes n}\left|\mathbf{x}\right\rangle=\frac{1}{2^n}\sum_{\mathbf{y}\in\{0,1\}^n}\left|\mathbf{y}\right\rangle\sum_{\mathbf{z}\in\{0,1\}^n}(-1)^{\mathbf{z}\cdot(\mathbf{x}+\mathbf{y})}=\frac{1}{2^n}\sum_{\mathbf{y}\in\{0,1\}^n}\left|\mathbf{y}\right\rangle 2^n\delta_{\mathbf{xy}}=\left|\mathbf{x}\right\rangle$$

con lo cual se demuestra que la transformada de Hadamard es su propia inversa.

Varios algoritmos cuánticos requerirán el uso de funciones $f:\{0,1\}^n\to\{0,1\}$ por lo que se puede asociar una compuerta U_f que realice el siguiente procedimiento

$$U_f\left|\mathbf{x}\right\rangle\left|y\right\rangle=\left|\mathbf{x}\right\rangle\left|y\oplus f(\mathbf{x})\right\rangle \tag{23}$$

donde $\oplus$ es la adición modulo 2. La utilidad de esta compuerta parte de que se puede escoger el qubit $|y\rangle$ como $|-\rangle$ y así se tendrá

$$U_f\left|\mathbf{x}\right\rangle\left|-\right\rangle=U_f\left|\mathbf{x}\right\rangle\left(\frac{\left|0\right\rangle-\left|1\right\rangle}{\sqrt{2}}\right)=\left|\mathbf{x}\right\rangle\left(\frac{\left|0\oplus f(\mathbf{x})\right\rangle-\left|1\oplus f(\mathbf{x})\right\rangle}{\sqrt{2}}\right)$$

recordando que el valor de $f(\mathbf{x})$ solo puede ser 0 o -1, el resultado del segundo qubit será $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$ o bien $\frac{|1\rangle-|0\rangle}{\sqrt{2}}$, resultado que se puede generalizar como

$$U_f\left|\mathbf{x}\right\rangle\left|-\right\rangle=(-1)^{f(\mathbf{x})}\left|\mathbf{x}\right\rangle\left|-\right\rangle \tag{24}$$

con lo que se puede ver que escogiendo un $|y\rangle$ específico, el operador U_f sólo altera la fase del sistema, fenómeno que se conoce como ***phase kickback***[14].

4.2. Algoritmos cuánticos principales

Después de haber descrito la representación matemática de los qubits y varias compuertas cuánticas en una y n-dimensiones, se analizarán los algoritmos principales de la computación cuántica.

4.2.1. Algoritmo de Deutsch-Jozsa

Suponga que se tiene una función binaria $f : \{0,1\}^n \to \{0,1\}$. De entrada se asumirá que esta función es *balanceada* o *constante*. Una función *balanceada* es tal que la mitad de sus entradas producirá un valor de 0 y la otra mitad producirá un valor de 1. Por otro lado, una función *constante* es tal que todas sus entradas producen el mismo valor de salida, bien sea 0 o 1. Este algoritmo permite determinar si la función en cuestión es balanceada o constante.

Para implementar este algoritmo clásicamente, note que se requerirá hacer $2^{n-1}+1$ llamados a la función f. Esto se debe a que para poder afirmar con certeza que la función es constante (y no balanceada) se debe evaluar la mitad más uno de sus entradas. Esto implica que la complejidad del algoritmo clásico es de orden exponencial $O(2^n)$. A continuación se explicará el algoritmo cuántico para mostrar que su complejidad es de orden constante, $O(1)$, puesto que sólo se requerirá llamar a la función $f(\mathbf{x})$ una vez. Para iniciar, se debe preparar un subsistema, o registro, de n qubits inicializado en $|0\rangle$ así como un qubit adicional inicializado en $|-\rangle$,

$$|\psi_0\rangle = |0\rangle^{\otimes n} |-\rangle$$

Posteriormente, se aplica el operador de Hadamard al primer registro de n qubits. Como se mostró en la ecuación 21, se obtiene

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}\in\{0,1\}^n} |\mathbf{x}\rangle |-\rangle$$

expresión a la cual se le puede aplicar el operador U_f descrito anteriormente. En este caso la función f asociada al operador será la que se pretende investigar. Como se vió en la ecuación 24, se tiene

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} U_f \left(\sum_{\mathbf{x}\in\{0,1\}^n} |\mathbf{x}\rangle |-\rangle \right) = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}\in\{0,1\}^n} U_f |\mathbf{x}\rangle |-\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}\in\{0,1\}^n} (-1)^{f(\mathbf{x})} |\mathbf{x}\rangle |-\rangle$$

estado al cual se le aplica la compuerta de Hadamard de nuevo, únicamente al primer registro.

$$|\psi_3\rangle = \frac{1}{2^n} \sum_{\mathbf{x}\in\{0,1\}^n} (-1)^{f(\mathbf{x})} \sum_{\mathbf{y}\in\{0,1\}^n} (-1)^{\mathbf{x}\cdot\mathbf{y}} |\mathbf{y}\rangle |-\rangle = \frac{1}{2^n} \sum_{\mathbf{x}\in\{0,1\}^n} \sum_{\mathbf{y}\in\{0,1\}^n} (-1)^{f(\mathbf{x})+\mathbf{x}\cdot\mathbf{y}} |\mathbf{y}\rangle |-\rangle$$

Ahora bien, note que cuando se mide la amplitud del qubit $|\mathbf{y}\rangle = |0\rangle^{\otimes n}$ se tiene un valor de

$$\left| \frac{1}{2^n} \sum_{\mathbf{x}\in\{0,1\}^n} (-1)^{f(\mathbf{x})} \right|^2$$

si la función f es constante, esta probabilidad será igual a 1 puesto que el valor de $(-1)^{f(\mathbf{x})}$ será 1 o -1. Esto significa que en este registro, sólo se tendrá una posibilidad de medida, el estado en el que todos los qubits son están en estado base 0, o$|0\rangle^{\otimes n}$. Por otro lado, si esta función es balanceada, esta probabilidad se anulara puesto que para cada 1 habrá un -1 que lo cancele. Esto implicará que el estado $|0\rangle^{\otimes n}$ no es probable así que se medirá al menos un estado $|1\rangle$ en el registro de n qubits. En conclusión, si todos los qubits medidos son $|0\rangle$ la función es constante y de lo contrario es balanceada. Note que este algoritmo cuántico sólo requirió un llamado a la función $f(\mathbf{x})$ [16].

4.2.2. Algoritmo de Bernstein-Vazirani

Ahora bien, suponga que se tiene de nuevo una función binaria $f_b : \{0,1\}^n \to \{0,1\}$ que depende de un hilo de bits $|\mathbf{b}\rangle$. El resultado de esta función es $f_b(\mathbf{x}) = \mathbf{x} \cdot \mathbf{b} \mod 2$. El algoritmo que se implementará permitirá encontrar $|\mathbf{b}\rangle$.

El algoritmo clásico para resolver este problema consiste en evaluar la función en aquellos estados que sólo tengan un 1 en uno de sus bits para así determinar el bit que se encuentra en esta posición de la secuencia de $\mathbf{b}$. En este caso se tendrá que realizar n operaciones que consisten en observar el resultado de las entradas $100\ldots, 010\ldots, \ldots, \ldots 001$. Por esta razón, este algoritmo opera en complejidad $O(n)$. El algoritmo cuántico mostrado a continuación será de nuevo de orden constante, $O(1)$.

Para implementar este algoritmo, se debe preparar un registro de n qubits en estado 0 y un registro de un qubit $|1\rangle$,

$$|\psi_0\rangle = |0\rangle^{\otimes n} |1\rangle$$

a los cuales se les aplica la compuerta de Hadamard para tener

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}\in\{0,1\}^n} |\mathbf{x}\rangle |-\rangle$$

posteriormente se aplica una compuerta que depende de la función de interés f_b, dada por $U_{f_b} |\mathbf{x}\rangle |y\rangle = |\mathbf{x}\rangle |y \oplus f_b(\mathbf{x})\rangle$. Como se estudió previamente en la ecuación 24, esto tendrá como resultado

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}\in\{0,1\}^n} (-1)^{\mathbf{x}\cdot\mathbf{b}} |\mathbf{x}\rangle |-\rangle$$

Note que el primer registro se puede entender como la transformada de Hadamard de $|\mathbf{b}\rangle$ por lo que al aplicar de nuevo la transformada sobre este se tendrá

$$|\psi_2\rangle = |\mathbf{b}\rangle |-\rangle$$

esto debido a la propiedad de que la transformada de Hadamard es su propia inversa[17].

4.2.3. Algoritmo de Simon

El problema planteado por el algoritmo de Simon parte de nuevo de una función $f : \{0,1\}^n \to B$ donde B es un conjunto finito. De esta función se sabe que existe una secuencia de bits $\mathbf{s}$ tal que $f(\mathbf{x}) = f(\mathbf{y})$ si y solo si $\mathbf{x} = \mathbf{y}$ o $\mathbf{y} = \mathbf{x} \oplus \mathbf{s}$. Note que en el caso en el que la cadena $\mathbf{s}$ esté compuesta totalmente de ceros, se tiene una función inyectiva.

Clásicamente, este problema se puede resolver con una complejidad de $O(2^{n/2})$ [18]. Sin embargo, como se mostrará a continuación cuánticamente se puede resolver con una complejidad lineal $O(n)$. Para esto se inicia con un sistema de dos registros de n qubits cada uno,

$$|\psi_0\rangle = |0\rangle^{\otimes n} |0\rangle^{\otimes n}$$

al cual se le aplicará la compuerta de Hadamard únicamente al primero para obtener,

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}\in\{0,1\}^n} |\mathbf{x}\rangle \, |0\rangle^{\otimes n}$$

a este estado se le aplicará una compuerta con la función de interés $U_f \, |\mathbf{x}\rangle \, |\mathbf{y}\rangle = |\mathbf{x}\rangle \, |\mathbf{y} \oplus f(\mathbf{x})\rangle$,

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}\in\{0,1\}^n} |\mathbf{x}\rangle \, |f(\mathbf{x})\rangle$$

En este punto se procede a medir el segundo registro, en el cual se obtiene un valor fijo de $f(\mathbf{x})$. Note que esta medición altera el estado del primer registro puesto que la entrada puede corresponder únicamente a $\mathbf{x}$ o a $\mathbf{y} = \mathbf{x} \oplus \mathbf{s}$. De esta manera, el primer registro queda

$$|\psi_3\rangle = \frac{1}{\sqrt{2}}(|\mathbf{x}\rangle + |\mathbf{y}\rangle)$$

A este se le aplica de nuevo la compuerta de Hadamard para obtener

$$|\psi_4\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{\mathbf{z}\in\{0,1\}^n} ((-1)^{\mathbf{x}\cdot\mathbf{z}} + (-1)^{\mathbf{y}\cdot\mathbf{z}}) \, |\mathbf{z}\rangle$$

Ahora bien note que esta amplitud se anula si $\mathbf{x} \cdot \mathbf{z} \neq \mathbf{y} \cdot \mathbf{z}$ puesto que se tendrá la suma de 1 y -1. Es decir, que al hacer la medición, sólo será diferente de nula si

$$\begin{aligned} &\mathbf{x} \cdot \mathbf{z} = \mathbf{y} \cdot \mathbf{z} \\ &\mathbf{x} \cdot \mathbf{z} = (\mathbf{x} \oplus \mathbf{s}) \cdot \mathbf{z} = \mathbf{x} \cdot \mathbf{z} \oplus \mathbf{s} \cdot \mathbf{z} \\ &\mathbf{s} \cdot \mathbf{z} = 0 \end{aligned}$$

lo cual implica que las mediciones sobre los qubits que no se anulen van a corresponder aquellos cuyo producto punto con $\mathbf{s}$ es cero. Este procedimiento se puede realizar aproximadamente n veces para así obtener un sistema de ecuaciones

$$\begin{cases} \mathbf{s} \cdot \mathbf{z}_1 = 0 \\ \mathbf{s} \cdot \mathbf{z}_2 = 0 \\ \mathbf{s} \cdot \mathbf{z}_3 = 0 \\ \vdots \\ \mathbf{s} \cdot \mathbf{z}_n = 0 \end{cases}$$

con lo cual se podrá determinar el valor de $|\mathbf{s}\rangle$[19].

4.2.4. Transformada de Fourier cuántica (QFT)

La transformada de Fourier cuántica es esencial para el desarrollo de distintos algoritmos en programación cuántica. En esta sección, se desarrollará matemáticamente cómo se obtiene esta

transformada y un algoritmo cuántico para calcularla. En principio, la transformada de Fourier cuántica, o QFT está definida para un qubit $|\mathbf{x}\rangle$ como

$$QFT\,|\mathbf{x}\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} \exp\left\{\frac{2\pi i x y}{2^n}\right\} |y\rangle \tag{25}$$

note que en la ecuación anterior, tanto x como y están escrito en base decimal. Particularmente, yc orresponde al qubit $|\mathbf{y}\rangle = |y_1 y_2 \dots y_n\rangle$ por lo que se tiene

$$y = \sum_{k=1}^{n} y_k 2^{n-k}$$
$$|\mathbf{y}\rangle = \bigotimes_{k=1}^{n} |y_k\rangle$$

Así pues un qubit múltiple $|\mathbf{y}\rangle = |1101\rangle$ corresponde a $y = 13$. Por esta razón, se puede reescribir la transformada como

$$\begin{aligned}
QFT\,|\mathbf{x}\rangle &= \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} \exp\left\{2\pi i x \sum_{k=1}^{n} \frac{y_k}{2^k}\right\} |y\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{y_1=0}^{1} \dots \sum_{y_n=0}^{1} \exp\left\{2\pi i x \sum_{k=1}^{n} \frac{y_k}{2^k}\right\} |y_1 \dots y_n\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{y_1=0}^{1} \dots \sum_{y_n=0}^{1} \bigotimes_{k=1}^{n} \exp\left\{2\pi i x \frac{y_k}{2^k}\right\} |y_k\rangle \\
&= \frac{1}{\sqrt{2^n}} \bigotimes_{k=1}^{n} \left[\sum_{y_k=0}^{1} \exp\left\{2\pi i x \frac{y_k}{2^k}\right\} |y_k\rangle\right]
\end{aligned}$$

Realizando la sumatoria de dos términos se tiene

$$QFT\,|\mathbf{x}\rangle = \frac{1}{\sqrt{2^n}} \bigotimes_{k=1}^{n} \left[|0\rangle + \exp\left\{\frac{2\pi i x}{2^k}\right\} |1\rangle\right] \tag{26}$$

Para entender el circuito que produce la transformada de Fourier cuántica es importante reescribir la compuerta de Hadamard (ecuación 21) para uno de los qubits de $|\mathbf{x}\rangle$ como

$$H_k\,|x_k\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle + \exp\left\{\frac{2\pi i x_k}{2}\right\} |1\rangle\right)$$

donde H_k hace referencia a que opera sobre el k-ésimo qubit. Así mismo, se utilizará una compuerta cuántica parametrizada (ecuación 15) con $\phi = \frac{2\pi}{2_k}$ dada por

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & \exp\left\{\frac{2\pi i}{2^k}\right\} \end{bmatrix}$$

En particular, se utilizará una versión controlada de la compuerta anterior,

$$CR_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \exp\left\{\frac{2\pi i}{2^k}\right\} \end{bmatrix}$$

Esta compuerta producirá entonces el efecto

$$CR_k\left|0\right\rangle\left|x_j\right\rangle = \left|0\right\rangle\left|x_j\right\rangle \qquad\qquad CR_k\left|1\right\rangle\left|x_j\right\rangle = \exp\left\{\frac{2\pi i x_j}{2^k}\right\}\left|1\right\rangle\left|x_j\right\rangle$$

El algoritmo para encontrar la transformada de Fourier consiste en aplicar la compuerta de Hadamard al primer qubit de $|\mathbf{x}\rangle$,

$$H_1\left|\mathbf{x}\right\rangle = H_1\left|x_1x_2\ldots x_n\right\rangle = \frac{1}{\sqrt{2}}\left(\left|0\right\rangle + \exp\left\{\frac{2\pi i x_1}{2}\right\}\left|1\right\rangle\right)\otimes\left|x_2\ldots x_n\right\rangle$$

a lo cual se aplica la compuerta CR_2 para obtener

$$CR_2H_1\left|\mathbf{x}\right\rangle = \frac{1}{\sqrt{2}}\left(\left|0\right\rangle + \exp\left\{\frac{2\pi i x_2}{2^2} + \frac{2\pi i x_1}{2}\right\}\left|1\right\rangle\right)\otimes\left|x_2\ldots x_n\right\rangle$$

Despues de aplicar las compuertas CR_3 hasta CR_n se tendrá

$$\begin{aligned} CR_n\ldots CR_2H_1\left|\mathbf{x}\right\rangle &= \frac{1}{\sqrt{2}}\left(\left|0\right\rangle + \exp\left\{\frac{2\pi i x_n}{2^n} + \ldots + \frac{2\pi i x_1}{2}\right\}\left|1\right\rangle\right)\otimes\left|x_2\ldots x_n\right\rangle \\ &= \frac{1}{\sqrt{2}}\left(\left|0\right\rangle + \exp\left\{2\pi i\sum_{k=1}^{n}\frac{x_k}{2^k}\right\}\left|1\right\rangle\right)\otimes\left|x_2\ldots x_n\right\rangle \\ &= \frac{1}{\sqrt{2}}\left(\left|0\right\rangle + \exp\left\{\frac{2\pi i x}{2^n}\right\}\left|1\right\rangle\right)\otimes\left|x_2\ldots x_n\right\rangle \end{aligned}$$

donde el último paso se atribuye a que x es la representación decimal de $x_1\ldots x_n$. Después de aplicar esta secuencia de pasos a los qubits $x_2,\ldots,x_n$ se tendrá

$$\frac{1}{\sqrt{2^n}}\left(\left|0\right\rangle + \exp\left\{\frac{2\pi i x}{2^n}\right\}\left|1\right\rangle\right)\otimes\left(\left|0\right\rangle + \exp\left\{\frac{2\pi i x}{2^{n-1}}\right\}\left|1\right\rangle\right)\otimes\ldots\otimes\left(\left|0\right\rangle + \exp\left\{\frac{2\pi i x}{2^1}\right\}\left|1\right\rangle\right)$$

lo cual es equivalente al producto tensorial generalizado de la ecuación 26, la transformada de Fourier cuántica.

4.2.5. Estimación de fase

Uno de los problemas esenciales que permite resolver la transformada de Fourier cuántica es el de la estimación de fase. Como se ha visto anteriormente, los operadores cuánticos se pueden expresar como una serie de rotaciones alrededor de la esfera de Bloch. Es así como este algoritmo permite estimar la fase de un valor propio de un operador, es decir que si se tiene

$$U\left|\psi\right\rangle = e^{2\pi i\theta}\left|\psi\right\rangle \tag{27}$$

se estimará el ángulo θ usando la transformada de Fourier.

En este caso se necesitará almacenar los qubits del estado cuántico $|\psi\rangle$, los cuales permanecerán casi intactos. Por otro lado, se necesita un conjunto de n qubits para guardar el valor de la fase. Inicialmente se prepara un estado

$$\left|\psi_0\right\rangle = \left|0\right\rangle^{\otimes n}\left|\psi\right\rangle,$$

al cual se le aplicará la compuerta de Hadamard n veces al primer subsistema,

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle)^{\otimes n} |\psi\rangle$$

Posteriormente se puede aplicar una compuerta controlada al operador U de tal manera que se tenga

$$CU |0\rangle = |0\rangle; \qquad CU |1\rangle = e^{2\pi i\theta} |1\rangle ,$$

usando esta compuerta se puede proceder a aplicarla 2^j veces sobre cada qubit con j desde 0 hasta $n-1$ para tener

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}}(|0\rangle + e^{2\pi i\theta 2^{n-1}} |1\rangle) \otimes \cdots (|0\rangle + e^{2\pi i\theta 2^0} |1\rangle) \otimes |\psi\rangle$$

note que esta es la transformada de Fourier del estado $|2^n\theta\rangle$ por lo que se tiene

$$|\psi_2\rangle = (QFT |2^n\theta\rangle) \otimes |\psi\rangle$$

Al realizar la transformada de Fourier inversa sobre el primer subsistema se tendrá una medida aproximada de $2^n\theta$, de donde se puede estimar la fase θ [12].

4.2.6. Algoritmo de Grover

El algoritmo de Grover es esencial en la computación cuántica pues permite buscar objetos de una lista. Este algoritmo utiliza un operador conocido como oráculo para verificar si una entrada es solución, es decir, si es lo que se está buscando. Este oráculo se apoya de una función $f(\mathbf{x})$ tal que

$$f(\mathbf{x}) = \begin{cases} 1 & \text{si } \mathbf{x} = \mathbf{w} \\ 0 & \text{si } \mathbf{x} \neq \mathbf{w} \end{cases}$$

En este caso, $\mathbf{w}$ es lo que se está buscando. La función oráculo es idéntica a la que ya se ha utilizado previamente al desarrollar el phase kickback (ecuación 23 por lo que se tendrá el mismo desfase que en la ecuación 24. El procedimiento para el algoritmo de Grover inicia creando una superposición de estados usando la transformada de Hadamard

$$|\mathbf{s}\rangle = H^{\otimes n} |0\rangle^n = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}=0}^{2^n-1} |\mathbf{x}\rangle ,$$

a la cual se le aplica el operador oráculo mencionado previamente,

$$U_f |\mathbf{s}\rangle = (-1)^{f(\mathbf{s})} |\mathbf{s}\rangle$$

A este último estado se le aplica un operador conocido como *difusor*[12] equivalente a $U_s = 2 |\mathbf{s}\rangle \langle \mathbf{s}| - I$ y así tener el estado $U_s U_f |\mathbf{s}\rangle$. Después de repetir esta iteración t veces se tendrá el estado $(U_s U_f)^t |\mathbf{s}\rangle$ que al medir será equivalente a $|\mathbf{w}\rangle$. Para entender por qué esta iteración funciona se

pueden definir dos estados ortogonales de un sistema con $N = 2^n$ elementos y M soluciones (es decir, búsquedas correctas). Estos estados estarán dados por

$$|\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_{\mathbf{x}\neq\mathbf{w}} |\mathbf{x}\rangle$$
$$|\beta\rangle = \frac{1}{\sqrt{M}} \sum_{\mathbf{x}=\mathbf{w}} |\mathbf{x}\rangle$$

Esto implicará que el estado inicial se puede escribir como

$$|\mathbf{s}\rangle = \frac{1}{\sqrt{N}} \sum_{\mathbf{x}} |\mathbf{x}\rangle = \frac{1}{\sqrt{N}} \left(\sum_{\mathbf{x}\neq\mathbf{w}} |\mathbf{x}\rangle + \sum_{\mathbf{x}=\mathbf{w}} |\mathbf{x}\rangle \right)$$
$$= \sqrt{\frac{N-M}{N}} |\alpha\rangle + \sqrt{\frac{M}{N}} |\beta\rangle$$

Note también que al aplicar la función oráculo a estos estados se tendrá

$$U_f |\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_{\mathbf{x}\neq\mathbf{w}} U_f |\mathbf{x}\rangle = \frac{1}{\sqrt{N-M}} \sum_{\mathbf{x}\neq\mathbf{w}} |\mathbf{x}\rangle = |\alpha\rangle$$
$$U_f |\beta\rangle = \frac{1}{\sqrt{M}} \sum_{\mathbf{x}=\mathbf{w}} U_f |\mathbf{x}\rangle = -\frac{1}{\sqrt{M}} \sum_{\mathbf{x}=\mathbf{w}} |\mathbf{x}\rangle = -|\beta\rangle$$

De esta manera, al aplicar el operador oráculo al estado inicial se tendrá

$$U_f |\mathbf{s}\rangle = \sqrt{\frac{N-M}{N}} |\alpha\rangle - \sqrt{\frac{M}{N}} |\beta\rangle$$

Por simplicidad, en este punto se puede definir un ángulo θ dado por

$$\cos\frac{\theta}{2} = \sqrt{\frac{N-M}{N}} \qquad\qquad \sin\frac{\theta}{2} = \sqrt{1-\cos^2\frac{\theta}{2}} = \sqrt{\frac{M}{N}}$$

con lo cual el vector $|\mathbf{s}\rangle$ se puede expresar como $|\mathbf{s}\rangle = \cos\frac{\theta}{2}|\alpha\rangle + \sin\frac{\theta}{2}|\beta\rangle$. En la base de los estados $|\alpha\rangle$ y β, el operador oráculo U_f es equivalente a la matriz

$$U_f = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

así como el operador U_s se podra´expresar como

$$U_s = 2|\mathbf{s}\rangle\langle\mathbf{s}| - I = 2\begin{bmatrix} \cos\frac{\theta}{2} \\ \sin\frac{\theta}{2} \end{bmatrix}\begin{bmatrix} \cos\frac{\theta}{2} & \sin\frac{\theta}{2} \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} 2\cos^2\frac{\theta}{2} & 2\cos\frac{\theta}{2}\sin\frac{\theta}{2} \\ 2\sin\frac{\theta}{2}\cos\frac{\theta}{2} & 2\sin^2\frac{\theta}{2} \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix}$$

De esta manera, se define un operador Grover como aquel que consiste en la iteración de la función oráculo y la función difusora $G = U_s U_f$, el cual se escribirá en forma matricial como

$$G = U_s U_f = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Observe entonces que el resultado de realizar una iteración Grover sobre el estado inicial $|\mathbf{s}\rangle$ será

$$\begin{aligned} G|\mathbf{s}\rangle &= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\frac{\theta}{2} \\ \sin\frac{\theta}{2} \end{bmatrix} \\ &= \begin{bmatrix} \cos\theta\cos\frac{\theta}{2} - \sin\theta\sin\frac{\theta}{2} \\ \sin\theta\cos\frac{\theta}{2} + \cos\theta\sin\theta\frac{\theta}{2} \end{bmatrix} = \begin{bmatrix} \cos\left(\theta + \frac{\theta}{2}\right) \\ \sin\left(\theta + \frac{\theta}{2}\right) \end{bmatrix} = \begin{bmatrix} \cos\frac{3\theta}{2} \\ \sin\frac{3\theta}{2} \end{bmatrix} = \cos\frac{3\theta}{2}|\alpha\rangle + \sin\frac{3\theta}{2}|\beta\rangle \end{aligned}$$

Como se puede ver, una iteración de Grover genera una rotación de un ángulo θ. Por lo que se tiene

$$G^t|\mathbf{s}\rangle = \cos\frac{(2t+1)\theta}{2}|\alpha\rangle + \sin\frac{(2t+1)\theta}{2}|\beta\rangle$$

Entre más iteraciones t se hagan, aumentará el valor del coeficiente de $|\beta\rangle$ lo que hará que al medir el sistema, este se encuentre en un estado $|\beta\rangle$ que representa una solución al problema [3].

4.2.7. Algoritmo de Shor

El último algoritmo que se discutirá es quizá uno de los más importantes en la computación cuántica. El problema planteado por Shor es el de tener una función dada por

$$f(x) = a^x \mod N$$

donde a y N son enteros que no tienen factores comunes. El problema que este algoritmo resuelve es encontrar el número, r, más pequeño tal que $f(r) = 1$. Este r se conoce como *orden* o *período*. Para esto se puede definir un operador U dado por

$$U|y\rangle = |ay \mod N\rangle \tag{28}$$

De esta manera note que al evaluar iterativamente el operador U sobre $|1\rangle$, habrá solo r estados posibles. Como ejemplo sea $a = 5$ y $N = 19$ entonces

$$\begin{aligned} U|1\rangle &= |5\rangle \\ U^2|1\rangle &= |6\rangle \\ U^3|1\rangle &= |11\rangle \\ U^4|1\rangle &= |17\rangle \\ &\vdots \\ U^{r-1}|1\rangle &= \\ U^r|1\rangle &= |1\rangle \end{aligned}$$

por lo que un vector propio del operador U puede estará dado por

$$
\begin{aligned}
|u_s\rangle &= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-\frac{2\pi i s k}{r}} \, |a^r \bmod N\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-\frac{2\pi i s k}{r}} U^k \, |1\rangle \\
U \, |u_s\rangle &= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-\frac{2\pi i s k}{r}} U^{k+1} \, |1\rangle = \frac{1}{\sqrt{r}} \sum_{k=1}^{r} e^{-\frac{2\pi i s (k-1)}{r}} U^k \, |1\rangle \\
&= e^{\frac{2\pi i s}{r}} \frac{1}{\sqrt{r}} \sum_{k=1}^{r} e^{-\frac{2\pi i s k}{r}} U^k \, |1\rangle \\
U \, |u_s\rangle &= e^{\frac{2\pi i s}{r}} \, |u_s\rangle
\end{aligned}
$$

donde s es un entero tal que $0 \leq s \leq r-1$. Note que en este caso tenemos un problema de estimación de fase, al igual que el descrito en la ecuación 27 con $\theta = \frac{s}{r}$, por lo que se puede utilizar este algoritmo para encontrar la razón de s y r [12].

4.3. Complejidad de algoritmos cuánticos

4.3.1. Algoritmo de Deutsch-Jozsa

Este algoritmo busca ver si una función es balanceada o constante, partiendo de la premisa que estas dos categorías son las únicas posibles. Si en particular se tiene una función $f : \{0,1\}^n \to 0,1$ se tendrá que hacer $2^{n-1}+1$ llamados para asegurarse que es balanceada o es constante. Esto es porque si en los primeros 2^n llamados la función ha arrojado el mismo resultado, podría aún tratarse de una función balanceada hasta que se mida el primero de la segunda mitad. Note sin embargo, que con el algoritmo cuántico sólo se tiene que hacer un llamado a la función de prueba f. Esto reduce una complejidad exponencial $O(2^{n-1}+1)$ a una constante $O(1)$.

4.3.2. Algoritmo de Bernstein-Vazirani

Este algoritmo, en su versión clásica, necesitaría de n pruebas pues se debe revisar cada uno de los dígitos de la secuencia. Es decir, probar las entradas $100\ldots, 010\ldots, \ldots, \ldots 001$. Entonces la complejidad de este algoritmo es lineal $O(n)$ pero su contra parte cuántica lo reduce a una complejidad constante $O(1)$, pues sólo se debe hacer un llamado de la función.

4.3.3. Algoritmo de Simon

El algoritmo de Simon se demostró que tenía una complejidad clásica de $O(2^{n/2})$ [18] pero cuántica mente se reduce esta complejidad a una lineal $O(n)$ puesto que este es el número de veces que se tiene que llamar al algoritmo para obtener un elemento $\mathbf{s} \cdot \mathbf{z}_i$ del sistema de ecuaciones.

4.3.4. Algoritmo de Grover

Este algoritmo se utiliza para la búsqueda de un elemento en un arreglo, lo que en computación clásica tiene una complejidad de $O(N)$. Sin embargo, en este caso se tiene realizar la iteración de

Grover hasta que el resultado este más cerca al vector que se está buscando $|\beta\rangle$. Para esto se puede pensar que basta con que el ángulo inicial rote una cantidad de $\arccos\sqrt{M/N}$. Esto implicará que se debe evaluar la iteración

$$t = \frac{\arccos\sqrt{M/N}}{\theta}$$

lo cual implica que $t \leq \frac{\pi}{2\theta}$. Ahora bien, se puede escribir que este ángulo siempre será mayor a

$$\frac{\theta}{2} \geq \sin\frac{\theta}{2} = \sqrt{\frac{M}{N}}$$

y de esta manera ver que se debe realizar la iteración de Grover un máximo de $t = \frac{\pi}{4}\sqrt{\frac{N}{M}}$ por lo que la complejidad de búsqueda será $O(\sqrt{N/M})$. Note que cuando sólo hay una solución posible ($M = 1$), esta complejidad se reduce a $O(\sqrt{N})$.

4.3.5. Algoritmo de Shor

Hasta el momento, no se ha desarrollado un algoritmo clásico que pueda factorizar números enteros en un tiempo polinomial [20].

Sin embargo, al utilizar la transformada de Fourier cuántica y el algoritmo de estimación de fase, el algoritmo de Shor reduce esta complejidad a un orden $O((\log N)^2(\log\log N)(\log\log\log N)$ [21]. Es por el hecho de que un algoritmo de tiempo no polinomial se vuelve un algoritmo de tiempo sub-exponencial que este algoritmo es de los más importantes en la computación cuántica y uno de los que más justifica el desarrollo de maquinas de computación cuántica.

En este documento se desarrolló el tema de computación cuántica desde sus posibles maneras de implementación física hasta sus algoritmos cuánticos principales. Se hizo una breve descripción de los sistemas cuánticos actuales que se utilizan para crear circuitos cuánticos como las trampas de iones, los fotones polarizados y el espín de núcleos atómicos manipulados por pulsos de campo magnético.

Por otra parte, se hizo una descripción matemática de los circuitos cuánticos desde los estados de los qubits y las diferentes compuertas cuánticas que se han formulado para un qubit y para sistemas de varios qubits. Esto permitió describir las diferencias en la implementación física de los computadores cuánticos y los clásicos y cómo estas diferencias repercuten en la implementación de sus unidades base, los qubits y los bits.

Posteriormente, se analizaron los algoritmos principales cuánticos: el algoritmo de Deutsch-Josza, el algoritmo de Bernstein-Vazirani, el algoritmo de Simon, el algoritmo de Grover y el algoritmo de Shor. Estos últimos dos los cuales requieren ayuda de la transformada de Fourier cuántica y el algoritmo de estimación de fase para funcionar correctamente. Se comparó su complejidad con aquella de sus contra partes clásicas y se observó que efectivamente, en todos los casos se tiene una mejora en términos de complejidad, en algunos casos incluso disminuyendo desde una complejidad exponencial a una complejidad constante.

Referencias

[1] P. Benioff, «The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines», *Journal of Statistical Physics*, vol. 22, n.º 5, págs. 563-591, mayo de 1980, ISSN: 1572-9613. DOI: 10.1007/BF01011339. dirección: https://doi.org/10.1007/BF01011339.

[2] L. Gyongyosi y S. Imre, «A Survey on quantum computing technology», en, *Computer Science Review*, vol. 31, págs. 51-71, feb. de 2019, ISSN: 15740137. DOI: 10.1016/j.cosrev.2018.11.002. dirección: https://linkinghub.elsevier.com/retrieve/pii/S1574013718301709 (visitado 08-09-2021).

[3] M. A. Nielsen e I. L. Chuang, *Quantum computation and quantum information*, en, 10th anniversary ed. Cambridge ; New York: Cambridge University Press, 2010, ISBN: 978-1-107-00217-3.

[4] P. W. Shor, «Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer», *SIAM Journal on Computing*, vol. 26, n.º 5, págs. 1484-1509, oct. de 1997, ISSN: 1095-7111. DOI: 10.1137/s0097539795293172. dirección: http://dx.doi.org/10.1137/S0097539795293172.

[5] C. Outeiral, M. Strahm, J. Shi, G. M. Morris, S. C. Benjamin y C. M. Deane, «The prospects of quantum computing in computational molecular biology», *WIREs Computational Molecular Science*, vol. 11, n.º 1, 2021. DOI: https://doi.org/10.1002/wcms.1481. dirección: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1481.

[6] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. D. Sawaya, S. Sim, L. Veis y A. Aspuru-Guzik, «Quantum Chemistry in the Age of Quantum Computing», *Chemical Reviews*, vol. 119, n.º 19, págs. 10 856-10 915, oct. de 2019, Publisher: American Chemical Society, ISSN: 0009-2665. DOI: 10.1021/acs.chemrev.8b00803. dirección: https://doi.org/10.1021/acs.chemrev.8b00803.

[7] S. J. Gay, «Quantum programming languages: survey and bibliography», en, *Mathematical Structures in Computer Science*, vol. 16, n.º 4, págs. 581-600, ago. de 2006, Publisher: Cambridge University Press, ISSN: 1469-8072, 0960-1295. DOI: 10.1017/S0960129506005378. dirección: https://www.cambridge.org/core/journals/mathematical-structures-in-computer-science/article/abs/quantum-programming-languages-survey-and-bibliography/80E4ECC8AE770B625A48F2EE28358BA6 (visitado 27-08-2021).

[8] E. Fredkin y T. Toffoli, «Conservative logic», *International Journal of Theoretical Physics*, vol. 21, n.º 3, págs. 219-253, abr. de 1982, ISSN: 1572-9575. DOI: 10.1007/BF01857727. dirección: https://doi.org/10.1007/BF01857727.

[9] T. Monz, K. Kim, W. Hänsel, M. Riebe, A. S. Villar, P. Schindler, M. Chwalla, M. Hennrich y R. Blatt, «Realization of the Quantum Toffoli Gate with Trapped Ions», *Phys. Rev. Lett.*, vol. 102, pág. 040 501, 4 ene. de 2009. DOI: 10.1103/PhysRevLett.102.040501. dirección: https://link.aps.org/doi/10.1103/PhysRevLett.102.040501.

[10] E. R. Johnston, N. Harrigan y M. Gimeno-Segovia, «Programming Quantum Computers», en, pág. 333,

[11] ——, *Programming quantum computers*. dirección: https://oreilly-qc.github.io/.

[12] *Qiskit 0.29.1 documentation — Qiskit 0.29.1 documentation*. dirección: https://qiskit.org/documentation/ (visitado 10-09-2021).

[13] N. Zettili, *Quantum mechanics: concepts and applications*, en, 2nd ed. Chichester, U.K: Wiley, 2009, OCLC: ocn255894625.

[14] P. Kaye, R. Laflamme y M. Mosca, *An introduction to quantum computing*, en, 1. publ. Oxford: Oxford University Press, 2007, ISBN: 978-0-19-857000-4 978-0-19-857049-3.

[15] A. Peres y D. R. Terno, «Quantum Information and Relativity Theory», en, dic. de 2002. DOI: 10.1103/RevModPhys.76.93. dirección: https://arxiv.org/abs/quant-ph/0212023v2 (visitado 12-10-2021).

[16] D. Deutsch y R. Jozsa, «Rapid solution of problems by quantum computation», *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, vol. 439, n.º 1907, págs. 553-558, dic. de 1992, Publisher: Royal Society. DOI: 10.1098/rspa.1992.0167. dirección: https://royalsocietypublishing.org/doi/10.1098/rspa.1992.0167 (visitado 26-10-2021).

[17] E. Bernstein y U. Vazirani, «Quantum Complexity Theory», *SIAM Journal on Computing*, vol. 26, n.º 5, págs. 1411-1473, oct. de 1997, Publisher: Society for Industrial and Applied Mathematics, ISSN: 0097-5397. DOI: 10.1137/S0097539796300921. dirección: https://epubs.siam.org/doi/10.1137/S0097539796300921 (visitado 29-10-2021).

[18] G. Cai y D. Qiu, «Optimal separation in exact query complexities for Simon's problem», en, *Journal of Computer and System Sciences*, vol. 97, págs. 83-93, nov. de 2018, ISSN: 0022-0000. DOI: 10.1016/j.jcss.2018.05.001. dirección: https://www.sciencedirect.com/science/article/pii/S0022000018305178 (visitado 29-10-2021).

[19] D. R. Simon, «On the Power of Quantum Computation», *SIAM Journal on Computing*, vol. 26, n.º 5, págs. 1474-1483, oct. de 1997, Publisher: Society for Industrial and Applied Mathematics, ISSN: 0097-5397. DOI: 10.1137/S0097539796298637. dirección: https://epubs.siam.org/doi/10.1137/S0097539796298637 (visitado 29-10-2021).

[20] S. Arora y B. Barak, *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. DOI: 10.1017/CBO9780511804090.

[21] D. Beckman, A. N. Chari, S. Devabhaktuni y J. Preskill, «Efficient networks for quantum factoring», en, *Physical Review A*, vol. 54, n.º 2, págs. 1034-1063, ago. de 1996, ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.54.1034. dirección: https://link.aps.org/doi/10.1103/PhysRevA.54.1034 (visitado 29-11-2021).