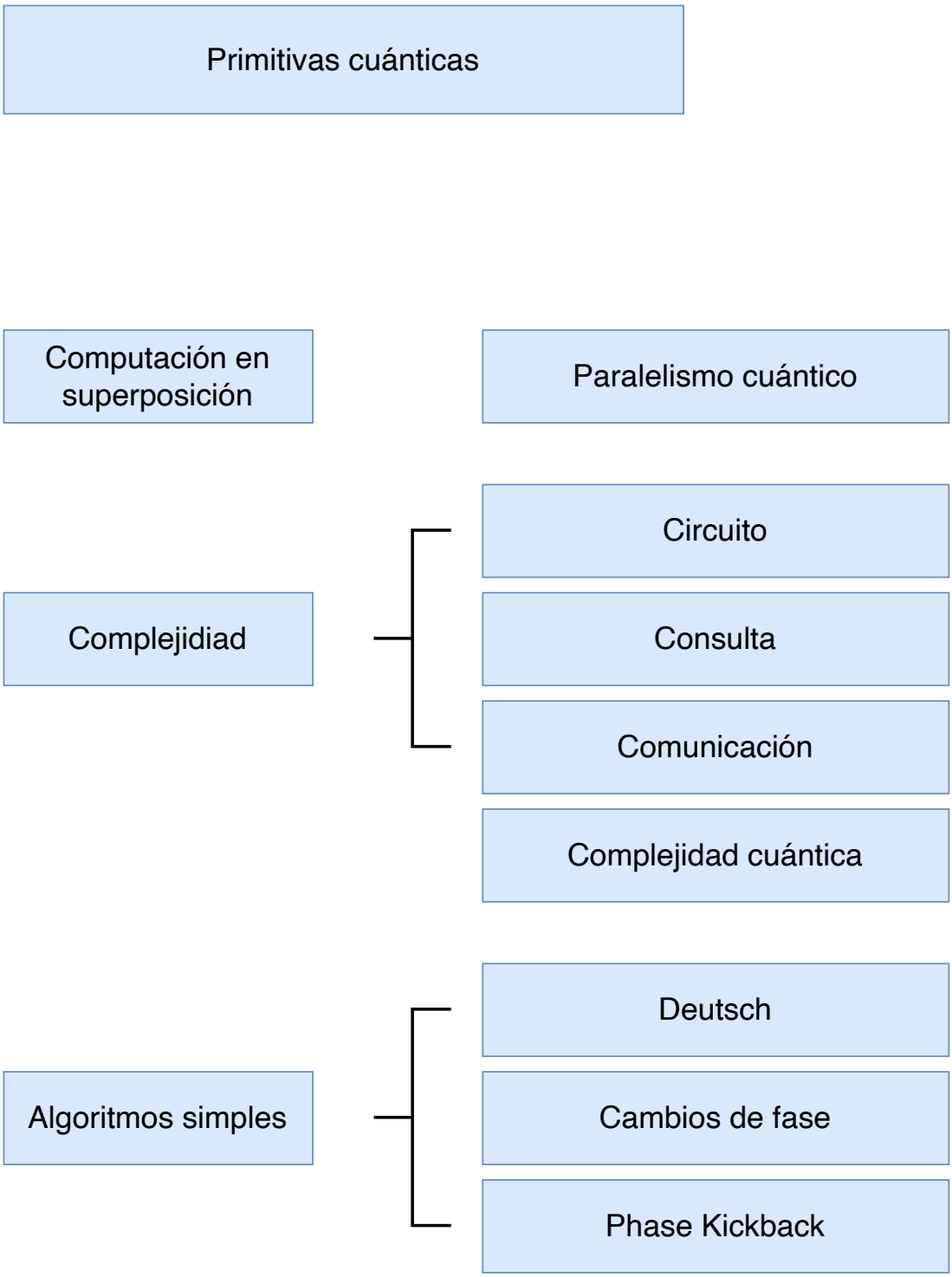
Computación Cuántica

Primitivas cuánticas

Índice

Esquema


Primitivas cuánticas
Computación en superposición
Paralelismo cuántico
Complejidiad
Circuito
Consulta
Comunicación
Complejidad cuántica
Algoritmos simples
Deutsch
Cambios de fase
Phase Kickback

Ideas clave

8.1 Introducción y objetivos

Este capítulo y los siguientes se dedican a los algoritmos cuánticos, cálculos cuánticos que superan a los clásicos. Estos algoritmos utilizan puertas simples y otras transformaciones unitarias más generales que no tienen equivalentes clásicas. Desde un punto de vista geométrico, todas las transformaciones del estado cuántico en n cúbits son rotaciones del espacio de estados complejo de 2^n dimensiones.

Se estudian las transformaciones cuánticas que pueden implementarse de manera eficiente y cómo tales transformaciones pueden usarse para acelerar ciertos tipos de computación. La clave para diseñar un algoritmo verdaderamente cuántico es descubrir cómo usar estas puertas unitarias básicas no clásicas para realizar un cálculo de manera más eficiente. En este y en los siguientes temas, los algoritmos se discuten en términos del modelo de circuito estándar de computación cuántica que se ha estudiado en temas anteriores.

La forma en que se calcula la eficiencia en el modelo de circuito cuántico se asemeja a la forma en que se calcula de forma clásica, lo que facilita la comparación de la eficiencia de los algoritmos cuánticos y clásicos.

Los primeros algoritmos cuánticos se diseñaron utilizando el modelo de circuito. Este modelo no es el único, ni necesariamente el mejor, para el diseño de algoritmos cuánticos.

En el modelo de circuito estándar de computación cuántica, la eficiencia de un algoritmo cuántico se calcula en función de la complejidad del circuito (número de puertas básicas junto con el número de cúbits utilizados), para el conjunto de circuitos utilizados en la implementación del algoritmo. A veces, la medida de la eficiencia considera otros recursos, por ejemplo, se podría considerar la cantidad de bits o cúbits trans-

mitidos entre dos partes para realizar una determinada tarea, o la cantidad de invocaciones a una función. A este último tipo de función se le suele denominar oráculo, ya que su funcionamiento interno se desconoce y no es accesible, solo el resultado de su aplicación. Estas diversas nociones de complejidad se estudian en los próximos apartados.

El tema comienza con el estudio general de la computación utilizando la superposición, incluyendo el concepto de paralelismo cuántico. A continuación se describen los tipos de complejidad relativos al circuito, la consulta y la comunicación. El algoritmo de Deutsch proporciona el primer ejemplo de un algoritmo verdaderamente cuántico, uno para el que no existe un análogo clásico.

Las subrutinas cuánticas permitirán la comprensión de varios algoritmos cuánticos simples, incluido el algoritmo de Simon, que inspiró el algoritmo de factorización de Shor. Si bien los problemas que resuelven estos algoritmos no son muy interesantes, un estudio de las técnicas que utilizan ayudará a comprender otros algoritmos más útiles como el algoritmo de Grover o el algoritmo de Shor.

- Computación en superposición
- Paralelismo cuántico
- Complejidad de circuito, consulta y comunicación
- Algoritmo de Deutsch
- Primitivas
- Cambios de fase
- Phase Kickback

8.2 Computando en superposición

Los algoritmos cuánticos suelen comenzar creando una superposición cuántica y luego haciéndola evolucionar a través de una versión cuántica de un circuito clásico que calcula una determinada función. Esta técnica, llamada paralelismo cuántico, no logra nada por sí misma (cualquier algoritmo que se detuviera en este punto no tendría ninguna ventaja sobre un algoritmo clásico), pero proporciona un punto de partida que resulta útil a la hora de diseñar algoritmos, por ejemplo, el algoritmo de Shor o el de Grover comienzan con la configuración de paralelismo cuántico.

El paralelismo cuántico, el primer paso de muchos algoritmos cuánticos, comienza utilizando la transformada de Hadamard para crear una superposición de todos los posibles valores de entrada.

La transformación de Hadamard, aplicada al estado $|0\rangle$ crea un estado en superposición, como se muestra gráficamente en la siguiente figura:

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

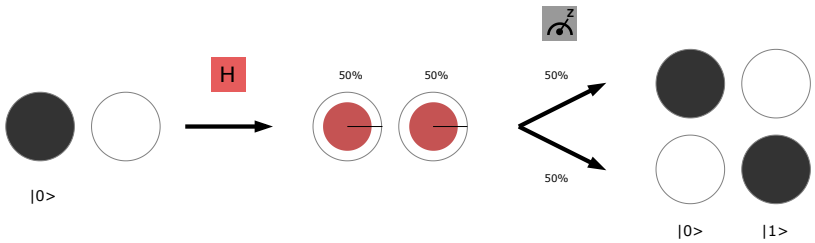


Figura 1: Aplicación de la puerta de Hadamard al estado $|0\rangle$,estado en superposición uniforme y medición. Elaboración propia.

Aplicando la transformada de Hadamard a cada uno de los cúbits de un registro de n cúbits, todos ellos en estado inicial $|0\rangle$, se genera una superposición de todos los 2^n vectores de la base computacional, que pueden verse como la representación en binario de los números de 0 a $2^n - 1$, es decir:

$$(H \otimes H \otimes \cdots \otimes H)|00...0\rangle = H^{\otimes n}|0\rangle^{\otimes n} =$$

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

$$\frac{1}{\sqrt{2^n}}(|0...00\rangle + |0...01\rangle + |0...10\rangle + \cdots + |1...11\rangle) = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$$

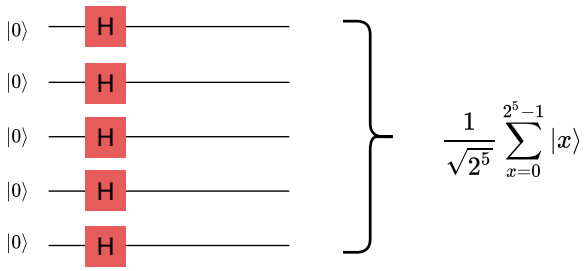


Figura 2: Aplicación de la puerta de Hadamard a un registro de 5 cúbits inicializados al estado. Elaboración propia.

Cualquier transformación de la forma $U_f = |x, y\rangle \rightarrow |x, y \oplus f(x)\rangle$ es lineal y por lo tanto, el resultado cuando actúa sobre una superposición de estados $\sum \alpha_x |x\rangle$ es el siguiente:

$$U_f : \sum_x \alpha_x |x, 0\rangle \rightarrow \sum_x \alpha_x |x, f(x)\rangle$$

Si se considera el efecto de aplicar U_f a la superposición de valores de 0 a $2^n - 1$, obtenidos de la transformación de Hadamard, se obtiene:

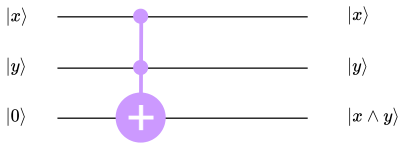
$$U_f : (H|0\rangle) \otimes |0\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle|0\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle|f(x)\rangle$$

Después de una sola aplicación de U_f, la superposición ahora contiene todos los 2^n valores de la función $f(x)$ entrelazados con su correspondiente valor de entrada x. A este efecto se le denomina paralelismo cuántico ya que n cúbits permiten trabajar simultáneamente con 2^n valores, el paralelismo cuántico, en cierto sentido, elude a la relación entre tiempo de cómputo y espacio necesario impuesto al paralelismo clásico, gracias a la capacidad de mantener, en una cantidad de espacio físico lineal, un número exponencial de valores.

Sin embargo, este efecto proporciona menos beneficios de los que se podrían esperar inicialmente. En primer lugar, solo se puede acceder a parte de la información en superposición, no es posible acceder de forma independiente a los 2^n valores de f. Se puede extraer información midiendo los estados, pero la medición, en la base estándar, proyectará el estado final en un solo par $|x, f(x)\rangle$ aleatoriamente.

El siguiente ejemplo muestra, utilizando la configuración básica, las limitaciones de

la superposición derivada del paralelismo cuántico por sí mismo, sin realizar ninguna transformación adicional.



$\|y\rangle$	$\|x\rangle$	$\|x \wedge y\rangle$
0	0	0
0	1	0
1	0	0
1	1	1

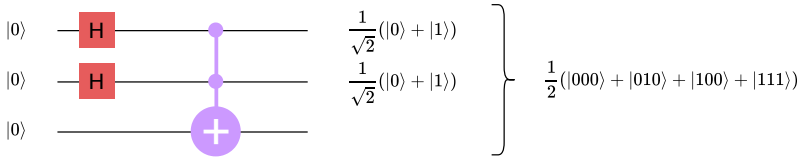


Figura 3: Puerta de Toffoli (CCNOT) que realiza la función AND de dos valores de entrada x e y, tabla de verdad de la función y resultado de su aplicación sobre un estado en superposición uniforme. Elaboración propia.

En lugar de elegir como valores de entrada para $|x\rangle$ e $|y\rangle$ los estados clásicos, $|0\rangle$ y $|1\rangle$, se proporciona una entrada en superposición de todas las posibles combinaciones de bits de x e y junto con un registro de un solo cúbit, inicializado a $|0\rangle$, para contener el resultado.

Aplicando puertas Hadamard a cada uno de los cúbits de entrada obtendremos el siguiente resultado:

$$H|00\rangle \otimes |0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle =$$

$$\frac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |110\rangle)$$

Al aplicar la puerta de Toffoli a esta superposición de entradas se obtiene:

$$CCNOT(H|00\rangle \otimes |0\rangle) = \frac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |111\rangle)$$

Esta superposición puede interpretarse como la tabla de verdad de la función AND. Los valores de x, y, y $x \bigwedge y$ están entrelazados de tal manera que la medición en la base estándar dará como resultado una de las líneas de la tabla de verdad.

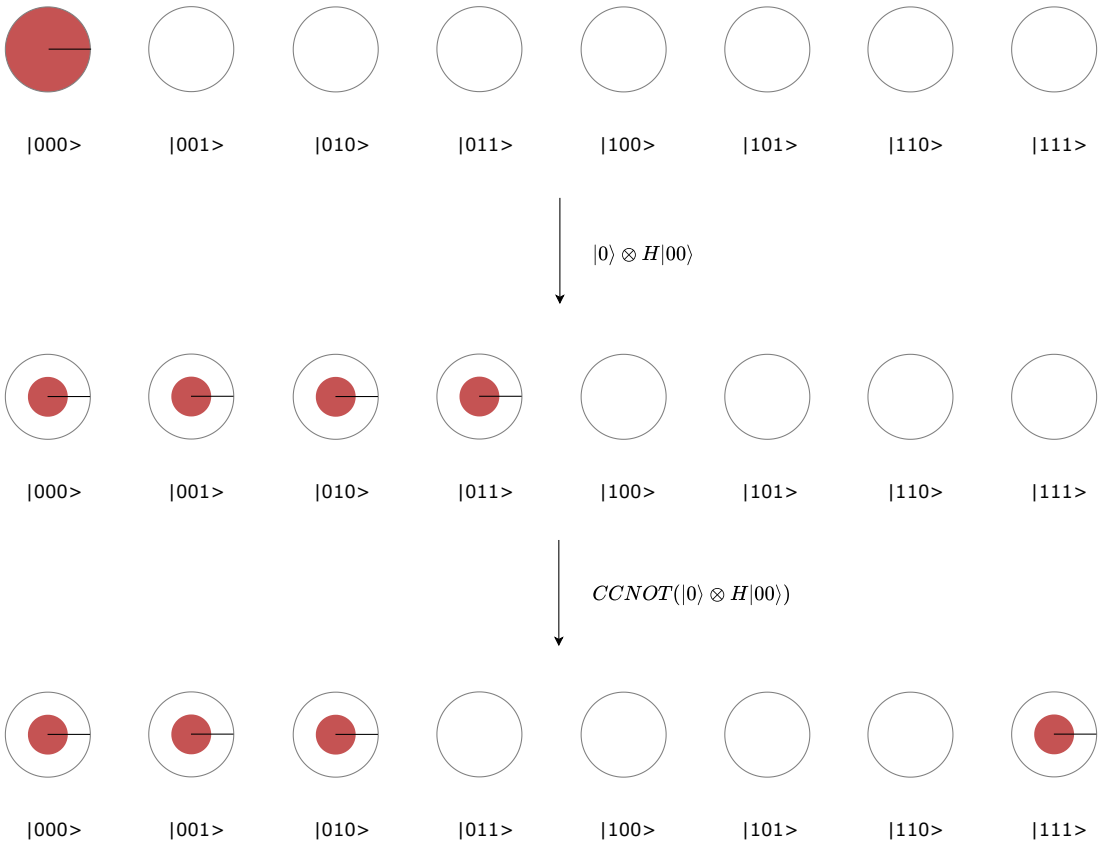


Figura 4: Notación con círculos del efecto de la puerta de Toffoli (CCNOT) sobre una superposición uniforme. Elaboración propia.

Realizar el cómputo de la función AND utilizando el paralelismo y luego realizar el proceso de medición utilizando la base estándar no proporciona ventaja alguna en comparación con el paralelismo clásico ya que solo se obtiene un resultado y es, además, aleatorio entre los posibles resultados.

La expresión $\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle|f(x)\rangle$ sugiere que la transformación cuántica U_f, actuando sobre la superposición $\sum_x |x\rangle|0\rangle$, está realizando una computación exponencialmente mayor a la que realiza una computación clásica que calcula $f(x)$ a partir de x. De manera similar, el tamaño exponencial del espacio de estados cuánticos de n cúbits puede sugerir que siempre se puede obtener una aceleración exponencial sobre el caso clásico utilizando el paralelismo cuántico. Esta afirmación es, generalmente, incorrecta, aunque en ciertos casos especiales la computación cuántica proporciona tales aceleraciones. Como se ha estudiado en varias ocasiones, el proceso de medida solo permite extraer un par entrada / salida del estado en superposición. No es posible extraer más pares de entrada / salida de ninguna otra manera ya que solo se pueden

extraer m bits de información de un estado de m cúbits. Por lo tanto, mientras que los 2^n valores de $f(x)$ aparecen en el estado de superposición, se necesitan 2^n cálculos de U_f para obtenerlos todos, como ocurre en el caso clásico. Si bien algunos algoritmos proporcionan una ventaja exponencial sobre los algoritmos clásicos, existen problemas para los que se sabe que ningún algoritmo cuántico puede proporcionar una aceleración exponencial. Además, es posible demostrar que para muchos problemas, la computación cuántica no puede proporcionar ninguna aceleración en absoluto. Por tanto, el paralelismo cuántico y la computación cuántica no proporcionan, en general, la aceleración exponencial sugerida por la notación. Además, una superposición como $\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle|f(x)\rangle$ no es sino un solo estado del espacio de estado. El espacio de estados cuánticos de un sistema de n cúbits es extremadamente grande, tan grande que la gran mayoría de los estados ni siquiera pueden aproximarse mediante un algoritmo cuántico eficiente. Por lo tanto, un algoritmo cuántico eficiente ni siquiera puede acercarse a la mayoría de los estados del espacio de estados. Por esta razón, el paralelismo cuántico no hace uso del espacio de estado completo como no lo hacen los algoritmos cuánticos eficientes. Incluso cuando el paralelismo cuántico puede usarse para describir un algoritmo, no es necesariamente correcto verlo como el factor determinante del algoritmo. Entender de dónde proviene el poder de la computación cuántica sigue siendo una cuestión de investigación abierta, como lo es, entender el potencial del entrelazamiento.

Cuando los algoritmos se describen en términos de paralelismo cuántico, la clave del algoritmo es la forma en que el algoritmo manipula el estado generado por el paralelismo cuántico. Este tipo de manipulación no tiene un análogo clásico y requiere técnicas de programación no tradicionales como la transformación del estado de tal forma que los valores que proporcionan los resultados del problema estén asociados a estados con una mayor amplitud y por tanto, tengan una mayor probabilidad de ser observados en el proceso de medida o la de encontrar propiedades en el conjunto de todos los valores de $f(x)$, como hacen varios de los algoritmos que se estudiarán en el siguiente tema.

Complejidad

La teoría de la complejidad analiza los recursos necesarios, normalmente tiempo o es-

pacio, para realizar un cálculo. Las máquinas de Turing proporcionan un modelo formal de computación que se utiliza a menudo para razonar sobre la complejidad computacional. Debido a que la mayoría de las investigaciones sobre algoritmos cuánticos discuten la complejidad en términos de complejidad de circuitos cuánticos, se seguirá ese enfoque. Otras medidas de complejidad es la complejidad de la consulta cuántica y una serie de medidas de complejidad que se utilizan para analizar los protocolos de comunicación cuántica.

Una familia de circuitos $C = C_n$ consta de circuitos C_n indexados por el tamaño máximo de entrada para ese circuito; el circuito C_n maneja entradas de tamaño n (bits o cúbits).

La complejidad de un circuito C se define como el número de puertas simples que contiene, donde se debe especificar el conjunto de puertas simples a considerar. Cualquiera de los conjuntos de puertas simples estudiados en temas anteriores proporcionaría la misma complejidad.

Los modelos de complejidad de circuitos no son uniformes, ya que para manejar tamaños de entrada más grandes se requieren circuitos diferentes y mayores. Tanto las máquinas de Turing cuánticas como las clásicas, por el contrario, pueden manejar una entrada arbitrariamente grande. La falta de uniformidad de los modelos de circuitos hace que la complejidad de los circuitos sea más complicada de definir que los modelos de máquina de Turing debido al siguiente problema: la complejidad puede ocultarse en la complejidad de construir los circuitos C_n, incluso si el tamaño de los circuitos C_n está acotado asintóticamente. Para poder obtener unos valores de complejidad sensatos, similares a los basados en la máquina de Turing, se deben imponer condiciones de uniformidad y de consistencia.

Una familia de circuitos C es consistente si sus circuitos C_n dan resultados consistentes para todo $m < n$: aplicar el circuito C_n a la entrada x de tamaño m debe dar el mismo resultado que aplicar C_m a esa entrada.

En cuanto a la uniformidad, la condición de uniformidad más común es la de uniformidad polinomial. Una familia de circuitos es polinomialmente uniforme si existe un algoritmo clásico de tiempo polinomial que genera los circuitos. La condición de uni-

formidad significa que la construcción del circuito no puede ser arbitrariamente compleja.

Complejidad en la consulta

Los primeros algoritmos cuánticos resuelven problemas de tipo oráculo, un oráculo genera el resultado $f(x)$ cuando se le proporciona la entrada x, como se muestra en la siguiente figura.

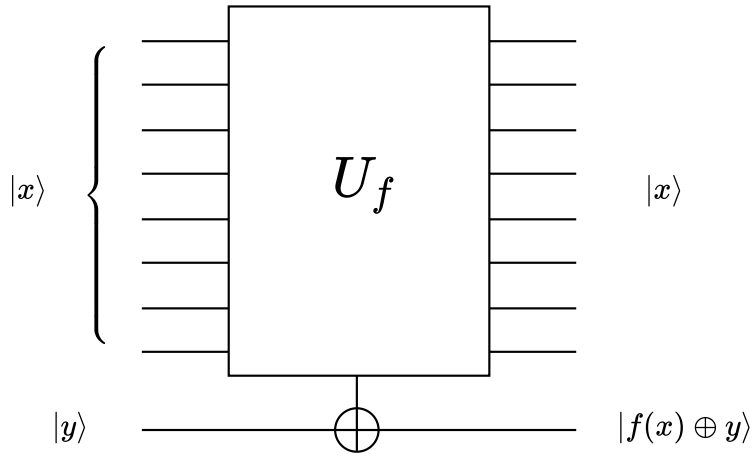


Figura 5: Oráculo. Elaboración propia.

El oráculo U_f transforma el estado de entrada de la siguiente forma:

Entrada: $\sum_x \alpha_x |y\rangle|x\rangle$

Resultado: $\sum_x \alpha_x |f(x) \oplus y, x\rangle$

El término oráculo resalta el hecho de que solo la salida puede usarse para resolver el problema, no se conoce, ni su implementación, ni cualquiera de los valores intermedios de su proceso de cómputo. El tipo de complejidad manejado normalmente para un oráculo es el de la complejidad de la consulta: cuántas consultas al oráculo son necesarias para resolver el problema.

Los algoritmos de tipo oráculo de baja complejidad de consulta, algoritmos que resuelven un problema con pocas consultas al oráculo, solo son de uso práctico si la implementación del oráculo es eficiente. El oráculo permite establecer límites inferiores a la complejidad del circuito de un problema. Si la complejidad de la consulta es de un orden de magnitud de N, es decir, se requieren al menos N consultas al oráculo,

entonces, la complejidad del circuito será, como mínimo, N.

Se han utilizado oráculos para determinar el límite inferior en la complejidad del circuito para ciertos algoritmos, pero su primer uso en la computación cuántica fue mostrar que la complejidad de la consulta del algoritmo cuántico, el número de consultas necesarias, para ciertos problemas de tipo oráculos, era significativamente menor que la complejidad de la consulta clásica de un oráculo clásico para resolver ese mismo mismo problema.

Complejidad de comunicación

Para los protocolos de comunicaciones, las medidas de complejidad incluyen el número mínimo de bits, o el número mínimo de cúbits, que deben transmitirse para realizar una tarea. Otros recursos, como el número de bits de aleatoriedad compartida o, en el caso cuántico, el número de pares EPR compartidos podrían o no considerarse. Existen varias nociones de complejidad relativas a la comunicación, dependiendo de si la tarea requiere la transmisión de información cuántica o clásica, si se pueden enviar cúbits o bits y qué recursos de entrelazamiento se pueden usar.

En el caso de la codificación superdensa, la complejidad que se considera es el número de cúbits que deben enviarse para comunicar n bits de información, que es $n/2$, al igual que el otro recurso, el número de pares entrelazados requeridos.

La teleportación, por el contrario, tiene como objetivo transmitir información cuántica utilizando un canal clásico que solo puede enviar bits, no cúbits. La noción de complejidad relevante es el número de bits necesarios para transmitir n cúbits de información cuántica de forma que se pueden usar $2n$ bits para transmitir el estado de n cúbits.

8.3 Algoritmo de Deutsch

Creado por David Deutsch en 1985, fue el primer algoritmo que mostró que la computación cuántica podría superar a la computación clásica, se trata de un problema de tipo oráculo con una complejidad de consulta inferior a cualquier algoritmo clásico, es

decir, resuelve el problema con menos consultas al oráculo de las que son necesarias en su versión clásica. Aunque se trata de un problema muy simple sin utilidad práctica, contiene una serie de elementos clave para la computación cuántica que se aplicarán en algoritmos cuánticos más complejos.

El problema de Deutsch se expresa de la siguiente forma: dada una función booleana $f : 0, 1 \rightarrow 0, 1$, se deberá determinar si esta función es o no constante.

Existen dos posibles configuraciones para el caso de una función constante:

Ambas entradas 0 y 1 son asignadas a 0. Es decir $f(0) = f(1) = 0$

Ambas entradas 0 y 1 son asignadas a 1. Es decir $f(0) = f(1) = 1$

Y otras dos para el caso en el que la función no es constante:

Ambas entradas 0 y 1 permanecen igual a la salida, sin cambios. Es decir $f(0) = 0$ y $f(1) = 1$

Ambas entradas 0 y 1 son intercambiadas en la salida. Es decir $f(0) = 1$ y $f(1) = 0$

El algoritmo cuántico de Deutsch, requiere una sola consulta al oráculo para resolver el problema mientras que cualquier algoritmo clásico requiere dos consultas a un oráculo clásico equivalente.

La siguiente implementación, mostrada en la figura, es irreversible y por tanto inválida en el contexto de la computación cuántica.

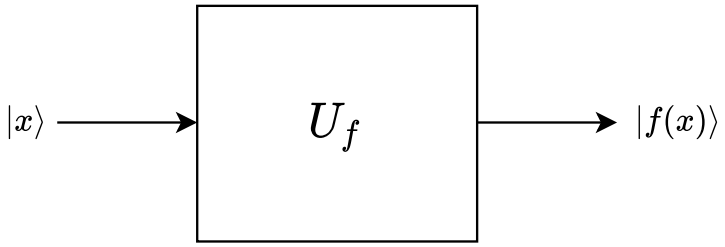


Figura 6: Una implementación de U_f no válida, al ser irreversible. Elaboración propia.

Como se ha estudiado en temas anteriores, para garantizar la reversibilidad, una función f que actúa sobre un único bit requiere una implementación cuántica mediante

una transformación unitaria U_f que tiene dos cúbits de entrada y dos de salida, como se puede observar en la figura siguiente.

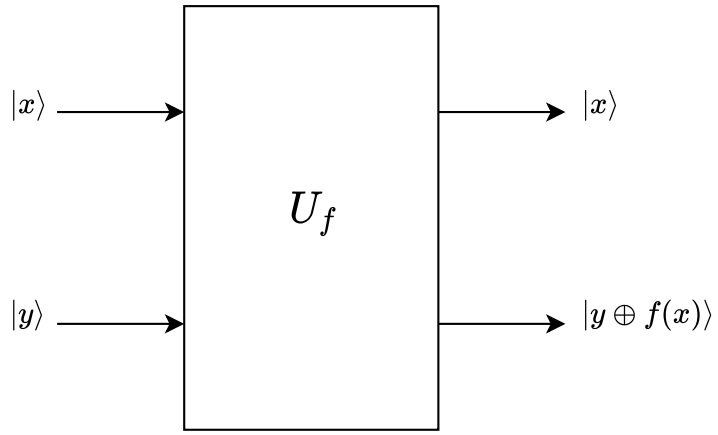


Figura 7: Efecto de la transformación unitaria U_f. Elaboración propia.

La clave del algoritmo está en colocar en estado de superposición, el segundo cúbit de entrada al oráculo.

Para la entrada $|x\rangle|y\rangle$, U_f produce $|x\rangle|f(x) \oplus y\rangle$, por lo que cuando $|y\rangle = |0\rangle$, el resultado de aplicar U_f es $|x\rangle|f(x)\rangle$.

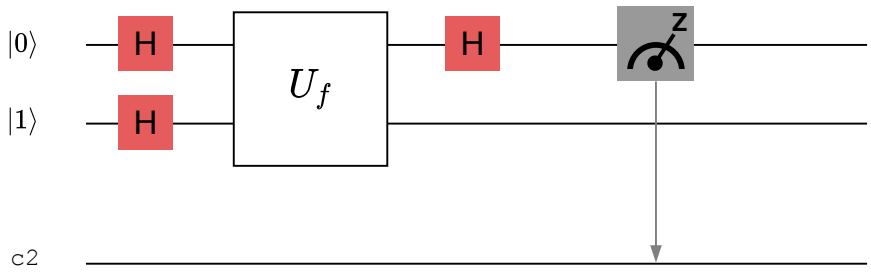


Figura 8: Algoritmo de Deutsch. Elaboración propia.

El algoritmo aplica U_f al estado de dos cúbit $|+\rangle|-\rangle$, donde el primer cúbit es una superposición de los dos valores en el dominio de f, y el segundo cúbit está en el estado en superposición: $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

Por lo tanto:

$$U_f(|+\rangle|-\rangle) = U_f(\tfrac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)) =$$

$$\tfrac{1}{2}(|0\rangle(|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle) + |1\rangle(|0 \oplus f(1)\rangle - |1 \oplus f(1)\rangle))$$

O lo que es equivalente: $U_f(|+\rangle|-\rangle) = \frac{1}{2}\sum_{x=0}^{1}|x\rangle(|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle)$

Si: $f(x) = 0$, entonces: $\frac{1}{\sqrt{2}}(|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle) = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$

Si: $f(x) = 1$, entonces: $\frac{1}{\sqrt{2}}(|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle) = \frac{1}{\sqrt{2}}(|1\rangle - |0\rangle) = -|-\rangle$

Por lo tanto:

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

En el caso en el que f es constante, $f(0) = f(1) = 0$ o $f(0) = f(1) = 1$ y por tanto:

$$\frac{1}{\sqrt{2}}\sum_{x=0}^{1}(-1)^{f(x)}|x\rangle|-\rangle = \frac{1}{\sqrt{2}}((-1)^0|0\rangle|-\rangle) + ((-1)^0|1\rangle|-\rangle)) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|-\rangle = |+\rangle|-\rangle$$

$$\frac{1}{\sqrt{2}}\sum_{x=0}^{1}(-1)^{f(x)}|x\rangle|-\rangle = \frac{1}{\sqrt{2}}((-1)^1|0\rangle|-\rangle) + ((-1)^1|1\rangle|-\rangle)) = \frac{1}{\sqrt{2}}(-|0\rangle - |1\rangle)|-\rangle = -|+\rangle|-\rangle$$

En el caso en el que f no es constante, $f(0) = 0, \quad f(1) = 1$ o $f(0) = 1, \quad f(1) = 0$ y por tanto:

$$\frac{1}{\sqrt{2}}\sum_{x=0}^{1}(-1)^{f(x)}|x\rangle|-\rangle = \frac{1}{\sqrt{2}}((-1)^0|0\rangle|-\rangle) + ((-1)^1|1\rangle|-\rangle)) = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)|-\rangle = |-\rangle|-\rangle$$

$$\frac{1}{\sqrt{2}}\sum_{x=0}^{1}(-1)^{f(x)}|x\rangle|-\rangle = \frac{1}{\sqrt{2}}((-1)^1|0\rangle|-\rangle) + ((-1)^0|1\rangle|-\rangle)) = \frac{1}{\sqrt{2}}(-|0\rangle + |1\rangle)|-\rangle = -|-\rangle|-\rangle$$

Si f es constante, $(-1)^{f(x)}$ es solo una fase global que carece de significado físico, por lo que el estado es simplemente $|+\rangle|-\rangle$.

En el caso en el que f no es constante, el término $(-1)^{f(x)}$ niega exactamente uno de los términos en la superposición, por lo que, ignorando la fase global, el estado es $|-\rangle|-\rangle$.

Si se aplica una puerta de Hadamard al primer cúbit y se realiza el proceso de medida, se obtendrá $|0\rangle$ en el caso de que f es constante y $|1\rangle$ cuando no lo es. Por tanto, con una sola consulta a U_f se puede resolver el problema demostrando además que los procesos cuánticos no tienen por qué ser probabilísticos.

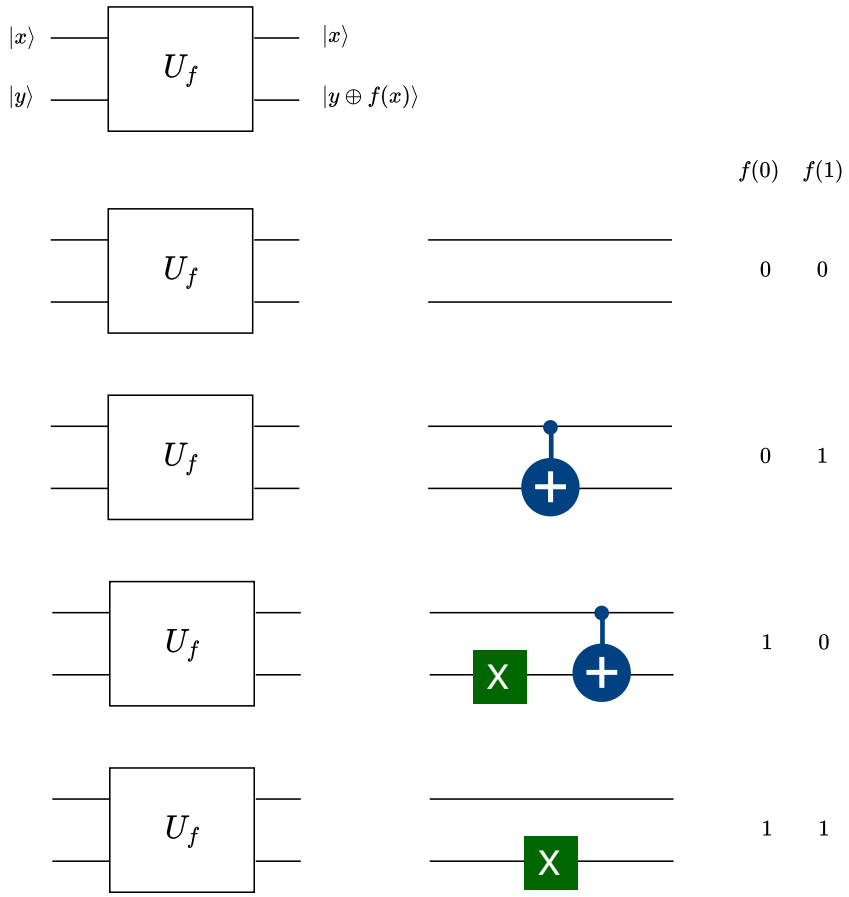


Figura 9: Implementación de cuatro posibles oráculos para el algoritmo de Deutsch. Elaboración propia.

La figura anterior muestra una forma de implementar cada uno de los cuatro oráculos U_f del algoritmo de Deutsch utilizando puertas cuánticas simples que realizan las cuatro posibles funciones f. En el caso 00, $f(0) = f(1) = 0$ y U_f actúa como la identidad. En el caso 11, $f(0) = f(1) = 1$ y U_f invierte el registro de salida con independencia del registro de entrada. En el caso 01, f actúa como la identidad, y por lo tanto U_f actúa como $CNOT$ con el registro de entrada como bit de control. En el caso 10, f intercambia 0 y 1, U_f invierte el bit de destino si y solo si el bit de control es 0. Esto es equivalente a combinar una puerta $CNOT$ con una inversión incondicional previa del bit de destino.

Rutinas cuánticas

Como se estudió en temas anteriores, es fundamental realizar la computación inversa de los cúbits auxiliares, con objeto, no solo de ahorrar espacio, como ocurre en compu-

tación clásica, sino también, para eliminar el entrelazamiento de los cúbits auxiliares con los cubis resultado de la computación, pues de otro modo, podría corromper los resultados del algoritmo. De otra forma, si un proceso de cómputo tiene como objetivo calcular $\sum_i \alpha|x_i\rangle$, no sería correcto calcular $\sum_i \alpha|x_i\rangle|y_i\rangle$ y desechar los cúbits correspondientes al registro $|y_i\rangle$ a no ser que no exista entrelazamiento entre ambos registros. Como se ha estudiado, no existe entrelazamiento si es posible expresar el estado como un producto tensor de los dos registros, lo cual puede ocurrir únicamente si $|y_i\rangle = |y_j\rangle \quad \forall i, j$.

$\sum_i \alpha|x_i\rangle|y_i\rangle = (\sum_i \alpha|x_i\rangle) \otimes |y_i\rangle$

En general, los estados $\sum_i \alpha|x_i\rangle$ y $\sum_i \alpha|x_i\rangle|y_i\rangle$ se comportan de forma muy distinta. Así, si sustituimos el oráculo utilizado en el algoritmo de Deutsch por otro V_f que actúa sobre tres cúbits, con el comportamiento que se muestra en la siguiente figura.

$V_f : |x, t, y\rangle \rightarrow |x, t \oplus x, y \oplus f(x)\rangle$

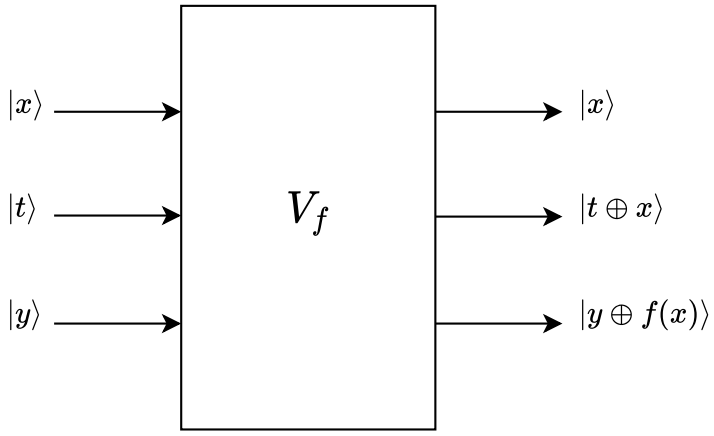


Figura 10: Efecto de la transformación unitaria V_f. Elaboración propia.

El algoritmo deja de funcionar correctamente. Si el estado inicial se establece como $|x, t, y\rangle = |+, 0, -\rangle$, al aplicar V_f:

$V_f(|+\rangle|0\rangle|-\rangle) = V_f(\frac{1}{\sqrt{2}} \sum_{x=0}^{1} |x\rangle|0\rangle|-\rangle) = \frac{1}{\sqrt{2}} \sum_{x=0}^{1} (-1)^{f(x)}|x\rangle|x\rangle|-\rangle)$

El primer cúbit se encuentra ahora entrelazado con el segundo y debido a ello, al aplicar la puerta de Hadamard al primer cúbit y realizar la medida, no se obtendrá el resultado esperado. Así, si f es constante, el estado será: $(|00\rangle + |11\rangle)|-\rangle$y aplicando

$H \otimes I \otimes I$ se obtendría: $\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle - |11\rangle)|-\rangle$ y por lo tanto, existe la misma probabilidad de observar $|0\rangle$ o $|1\rangle$, al igual que ocurre cuando la función no es constante y por lo tanto no es posible distinguir entre los dos estados, el entrelazamiento con el cúbit $|t\rangle$ ha alterado el algoritmo haciéndolo inservible. Si el oráculo V_f hubiera realizado la computación inversa al cúbit $|t\rangle$ devolviendolo al estado inicial $|0\rangle$, el algoritmo funcionaria correctamente, por ejemplo, para una función f constante el estado sería:

$$\frac{1}{2}(|00\rangle + |10\rangle + |00\rangle - |10\rangle)|-\rangle = \frac{1}{2}(2|00\rangle)|-\rangle = |00\rangle|-\rangle$$

Si una subrutina tiene como objetivo producir un determinado estado y hace uso de cúbits auxiliares, al finalizar la subrutina los cúbits auxiliares no deberán estar entrelazados con los cúbits objeto del computo pues de no ser así, no se obtendría el estado que se busca.

En las siguientes subrutinas siempre se realiza la computación inversa de los cúbits auxiliares garantizando que su estado final sea$|0\rangle$.

Cambio de fase para un subconjunto de vectores

El objetivo de esta subrutina es realizar un cambio en la fase de un determinado subconjunto $X : \{0, 1, ..., N-1\}$ de términos de una superposición $|\psi\rangle = \sum_i \alpha_i |i\rangle$. Es decir, se busca una implementación eficiente de la siguiente transformación:

$$S_X^\theta : \sum_{x=0}^{N-1} \alpha_x |x\rangle \rightarrow \sum_{x \in X} \alpha_x e^{i\theta} |x\rangle + \sum_{x \notin X} \alpha_x |x\rangle$$

Para ello será necesario, a su vez, un algoritmo eficiente para computar si un determinado elemento pertenece al conjunto X, es decir, una función $f(x)$, tal que:

$$f(x) = \begin{cases} 1, & x \in X \\ 0, & x \notin X \end{cases}$$

La mayoría de los subconjuntos X no tienen esta propiedad. Para los subconjuntos con esta propiedad, existe un circuito cuántico eficiente para U_f. Dada tal implementación para U_f, es posible calcular S_X^θ con algunos pasos adicionales. Se calcula f sobre un cúbit auxiliar utilizando U_f, se usa el valor del cúbit auxiliar para efectuar el cambio de fase, y a continuación se realiza la computación inversa sobre el cúbit auxiliar para

eliminar el entrelazamiento entre este y el resto del estado.

Primero se aplica la función f al cúbit auxiliar a, a continuación se desplaza la fase $e^{i\theta}$ solo en el caso en el que el cúbit a se encuentre en estado $|1\rangle$ y finalmente se realiza la computación inversa para eliminar el entrelazamiento entre el cúbit auxiliar a y el registro $|x\rangle$ de forma que $|x\rangle$ termine en el estado deseado y sin entrelazamiento con el cúbit auxiliar.

Cuando el ángulo $\theta = \pi$ existe una implementación muy simple: dada U_f, la transformación S_x^π puede implementarse inicializando un cúbit temporal en el estado $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ y entonces utilizando U_f para computar.

Sea: $|\psi\rangle = \sum_{x\in X} \alpha_x|x\rangle + \sum_{x\notin X} \alpha_x|x\rangle$ entonces:

$$U_f(|\psi\rangle \otimes |-\rangle) = U_f(\sum_{x\in X} \alpha_x|x\rangle \otimes |-\rangle) + U_f(\sum_{x\notin X} \alpha_x|x\rangle \otimes |-\rangle) =$$

$$-(\sum_{x\in X} \alpha_x|x\rangle \otimes |-\rangle) + (\sum_{x\notin X} \alpha_x|x\rangle \otimes |-\rangle) = (S_x^\pi|\psi\rangle) \otimes |-\rangle$$

En particular, el siguiente circuito, actuando sobre un estado $|0\rangle$ de n cúbits, junto con un cúbit auxiliar en estado $|1\rangle$ crea una superposición $|\psi_x\rangle = \sum(-1)^{f(x)}|x\rangle$ donde se aplica finalmente una puerta de Hadamard al cúbit auxiliar con objeto de que pueda ser reutilizado.

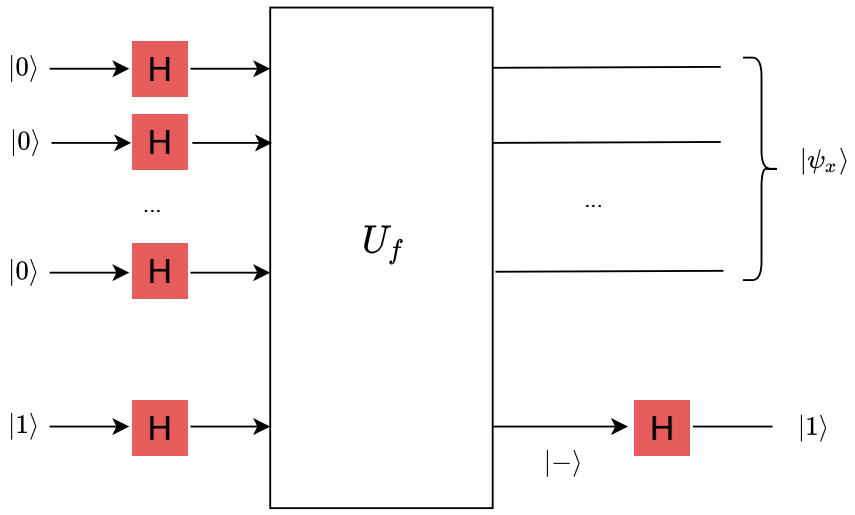


Figura 11: Circuito para el caso $\theta = \pi$. Elaboración propia.

Geométricamente, cuando se actúa sobre el espacio vectorial de dimensión N asociado con el sistema cuántico, la transformación S_X^π es una reflexión sobre el hiper-

plano de dimensión $N-k$ perpendicular al hiperplano de dimensión k generado por $|x\rangle|x \in X$.

Una reflexión sobre un hiperplano transforma cualquier vector $|v\rangle$ perpendicular al hiperplano en $-|v\rangle$.

Para cualquier transformación unitaria U, la transformación $US_X^{\pi}U^{-1}$ es una reflexión sobre el hiperplano perpendicular al hiperplano generado por los vectores $U|x\rangle|x \in X$.

Se puede escribir el resultado de aplicar la transformación S_X^{π} a la superposición $H|0\rangle^{\otimes n}$como:

$|\psi_x\rangle = \frac{1}{\sqrt{N}}\sum(-1)^{f(x)}|x\rangle$ donde f es la función booleana de pertenencia al conjunto X:

$$f(x) = \begin{cases} 1, & x \in X \\ 0, & x \notin X \end{cases}$$

O al revés, dada una función booleana f, se define S_f^{π} como S_X^{π} donde: $X = x|f(x) = 1$

Phase kickback

La ejecución de un algoritmo cuántico requiere preparar los estados, ejecutar los circuitos cuánticos sobre ellos, y finalmente leer el resultado. Phase Kickback utiliza la fase relativa, que es un número de la forma $e^{i\phi}$ y que juega un papel muy importante en la mecánica cuántica, la información cuántica y la computación cuántica.

Cada transformación unitaria puede escribirse en términos de los estados de su base propia, y su acción se reduce a un conjunto de estas fases. Si la aplicamos sobre uno de los autoestados, la salida es el mismo estado, multiplicado por esa fase.

Muchos algoritmos cuánticos implican la acumulación de fases por lo que es necesaria una manera de leer estas fases.

La más simple de todas las fases es la fase global que, como se ha estudiado en temas anteriores, carece de significado físico. Si un operador multiplica el estado por esta fase global, tal y como hace una transformación unitaria cuando actúa sobre sobre uno de sus autoestados, la probabilidad de observar un resultado, definida por la regla de Born, no cambia. Esto significa que la fase global no tiene ningún efecto medible

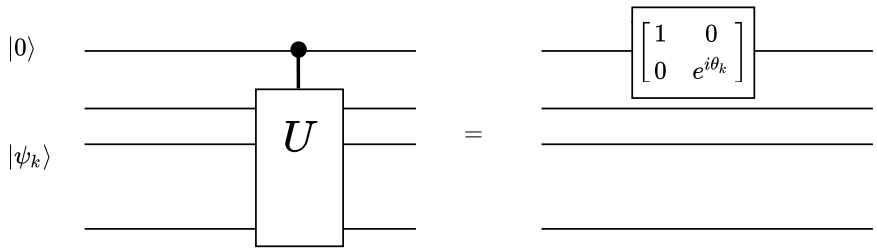
físicamente.

Sin embargo, se puede definir un operador O que multiplica cualquier estado por una fase global, de la forma:

$$O_\theta = \begin{bmatrix} e^{i\theta} & 0 \\ 0 & e^{i\theta} \end{bmatrix}$$

Sin embargo, cuando se define una operación controlada sobre ese operador, se transfiere una fase a un componente del estado del cúbit de control. Esto contrasta con la forma normal de pensar sobre las operaciones controladas, en el que el cúbit objetivo es modificado dependiendo del estado del cúbit de control.

Si se aplicamos una operación U controlada a un registro donde el subsistema objetivo está preparado en un autoestado de U, entonces U actúa como el operador O_θ, y la fase correspondiente se transfiere al cúbit de control, donde podría leerse.



$$U|\psi_k\rangle = e^{i\theta_k}|\psi_k\rangle$$

Figura 12: Circuito genérico que muestra la técnica de Phase kickback. Elaboración propia.

Si una sola lectura no permite leer la fase exactamente, este procedimiento se puede repetir, ya que no altera el estado de entrada y sigue siendo el caso si el registro contiene muchos cúbits. La versatilidad y simplicidad de usar un solo cúbit para leer la acción de una transformación unitaria, arbitrariamente grande, es lo que hace que phase kickback sea una técnica fundamental en computación cuántica.

En temas anteriores se ha estudiado como crear entrelazamiento entre dos cúbits utilizando una puerta $CNOT$ y colocando al cúbit de control en estado de superposición

uniforme $|+\rangle$, como se muestra en la siguiente figura..

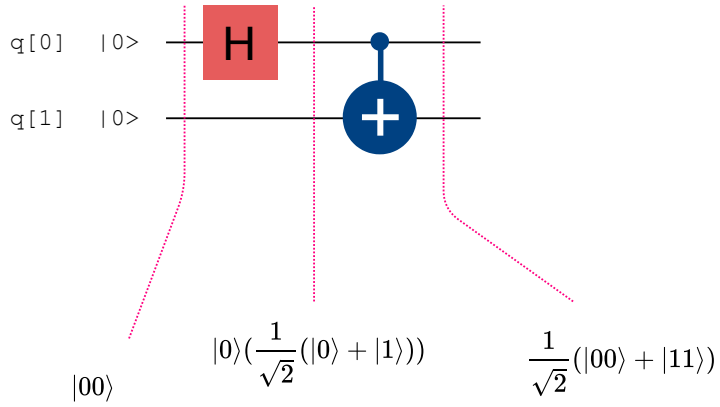


Figura 13: Entrelazamiento utilizando una puerta CNOT y una puerta de Hadamard que coloca al cúbit de control en el estado $|+\rangle$. Elaboración propia.

Si se colocara, no solo el cúbit de control sino también el cúbit objetivo en superposición, entonces la puerta $CNOT$ actuaría sobre el estado $|++\rangle$, que es una superposición uniforme de los cuatro estados de la base y puesto que $CNOT$ asigna el estado $|01\rangle \longrightarrow |11\rangle$ y al estado $|11\rangle \longrightarrow |01\rangle$ finalmente no se produciría ningún cambio y es sistema continuaría en el mismo estado $|++\rangle$.

Si en lugar de $|+\rangle$, se coloca al cúbit objetivo en el estado $|-\rangle$, que tiene una fase negativa, el estado, antes de aplicar la puerta $CNOT$ será: $|-+\rangle = \frac{1}{2}(|00\rangle + |01\rangle - |10\rangle - |11\rangle)$, al aplicar ahora $CNOT$ sobre ese estado, lo transformaría de la siguiente forma: $CNOT|-+\rangle = \frac{1}{2}(|00\rangle - |01\rangle - |10\rangle + |11\rangle) = |--\rangle$, lo cual es un resultado muy interesante ya que como se puede observar el cúbit objetivo permanece sin cambios pero cúbit de control tiene ahora una fase negativa, se ha visto afectado por la puerta $CNOT$ ya que el estado $|-\rangle$, es un autoestado del operador NOT.

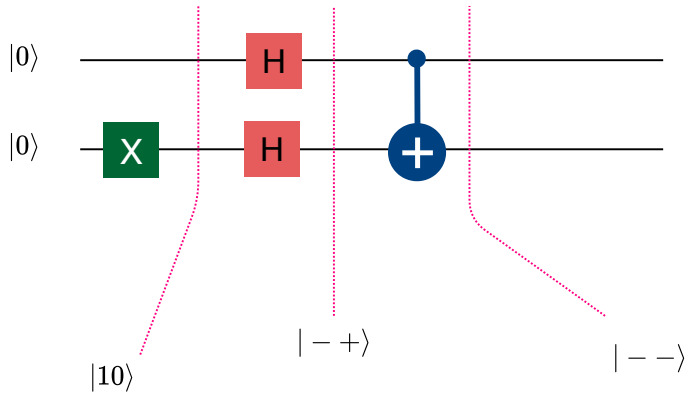
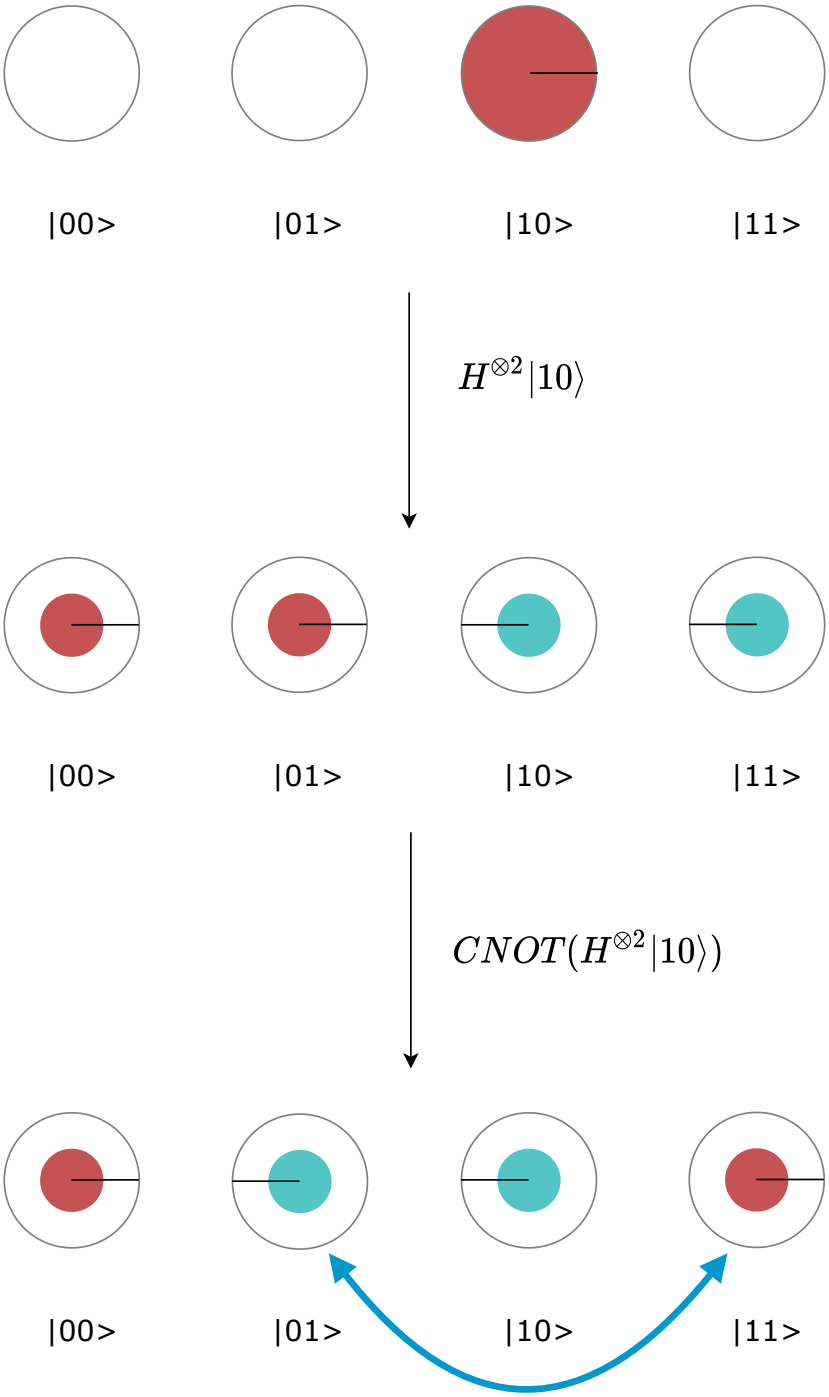


Figura 14: Phase kickback con $CNOT$ como puerta de control. El cúbit objetivo permanece sin cambios pero el cúbit de control adquiere una fase negativa. Elaboración propia.

Resumiendo, Phase Kickback es una técnica fundamental que se utiliza en multitud de circuitos cuánticos como Deutsch-Jozsa, Grover, Simon, Bernstein-Vazirani, Shor, etc. Es por tanto importante entenderla para facilitar a su vez la comprensión de esos otros algoritmos de una forma más intuitiva. Un aspecto clave para la aplicación de esta técnica es que el estado en el que se encuentra el cúbit objetivo de la operación de control deberá ser un autovector de la puerta correspondiente de la operación de control. Lo que se pretende con ello es que la aplicación de la operación al cúbit objetivo no cambie su estado, sino que solo añada una fase a su estado. Por lo tanto, aplicar el operador U al cúbit objetivo supondría: $U|\psi\rangle = e^{i\theta}|\psi\rangle$ Este valor $e^{i\theta}$ que multiplica al estado es el que será *propagado* al cúbit de control, modificándolo.



Evolución del circuito de la figura 14 en notación de círculos donde se observa la transferencia de la fase al cúbit de control. Elaboración propia.

8.4 Referencias bibliográficas

Nielsen and Chuang (2011) Quantum Computation and Quantum Information

Aaronson (2013), Quantum Computing Since Democritus

Eric R. Johnston, Nic Harrigan y Mercedes Gimeno-Segovia (2019), Programming Quantum Computers

Eleanor Rieffel and Wolfgang Polak (2011), Quantum Computing

Robert Sutor (2019), Dancing with cúbits