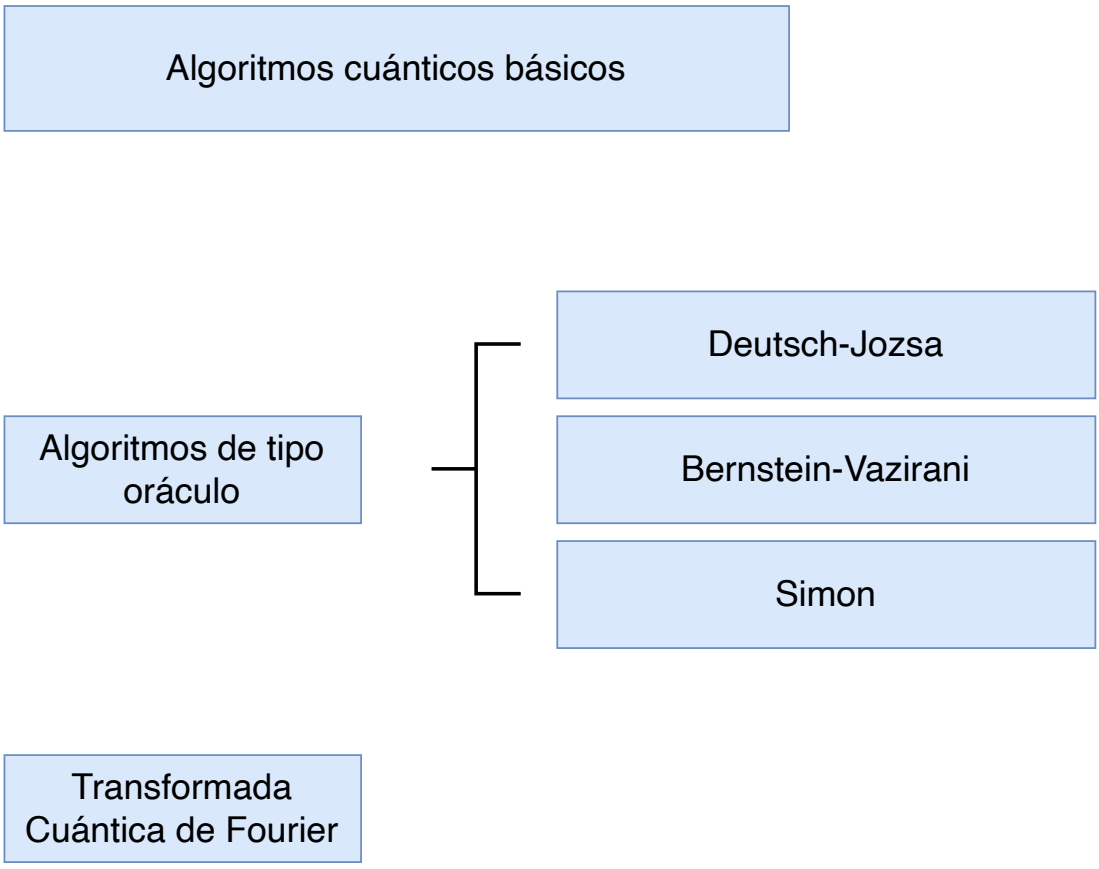
Computación Cuántica

Algoritmos cuánticos básicos

Índice

Esquema


Algoritmos cuánticos básicos
Algoritmos de tipo oráculo
Deutsch-Jozsa
Bernstein-Vazirani
Simon
Transformada Cuántica de Fourier

Ideas clave

9.1 Introducción y objetivos

En este tema se estudian varios algoritmos de tipo oráculo cuya complejidad de consulta es mejor que la de cualquier algoritmo clásico, son algoritmos con una aplicación practica cuestionable, relativamente simples, pero que demuestran una mayor eficiencia comparados con cualquier versión clásica, son los algoritmos de Deutsch-Jozsa, una generalización del algoritmo de Deutsch para múltiples cúbits, el algoritmo de Bernstein-Vazirani y el algoritmo de Simon, todos ellos proporcionan una solución al problema con certeza. También se estudia, finalmente la transformación cuántica de Fourier (QFT), una herramienta de extraordinaria importancia en computación cuántica, ella y sus generalizaciones tienen aplicación en muchos algoritmos cuánticos que muestran una ventaja exponencial frente a sus versiones clásicas.

- Algoritmo de Deutsch-Jozsa
- Algoritmo de Bernstein-Vazirani
- Algoritmo de Simon
- Transformada cuántica de Fourier

9.2 Algoritmo de Deutsch-Jozsa

El algoritmo Deutsch-Jozsa es un algoritmo cuántico determinista propuesto por David Deutsch y Richard Jozsa en 1992 con aportaciones posteriores de Richard Cleve, Artur Ekert, Chiara Macchiavello y Michele Mosca. Si bien, como se ha mencionado, es un algoritmo de escasa aplicación práctica, es uno de los primeros ejemplos de un algoritmo cuántico que es exponencialmente más rápido que cualquier posible algoritmo

clásico determinista.

En el problema propuesto por David Deutsch y Richard Jozsa, se proporciona un oráculo que implementa una función $f : \{0,1\}^n \to \{0,1\}$.

La función recibe como entrada una cadena de dígitos binarios de longitud n y produce un 0 o un 1 como salida para cada uno de esos valores. Se asegura que la función es constante (0 en todas las salidas o 1 en todas las salidas) o balanceada (devuelve 1 para la mitad del dominio de entrada y 0 para la otra mitad). El objetivo es determinar si f es constante o balanceada mediante consultas al oráculo $U_f : |x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle$

El algoritmo comienza con un estado de $n+1$ cúbits, $|0\rangle^{\otimes n}|1\rangle$, es decir los primeros n cúbits, correspondientes al primer registro, inicializados en el estado $|0\rangle$ y un cúbit auxiliar, correspondiente al segundo registro, en estado $|1\rangle$. A continuación se aplican puertas de Hadamard a todos los cúbits, dejando al sistema en el siguiente estado:

$$|\psi_0\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle(|0\rangle - |1\rangle)$$

La función f está implementada como un oráculo que asigna el estado $|x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle$, por tanto, aplicando el oráculo se obtiene:

$$|\psi_1\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle(|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle) = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle(|f(x)\rangle - |1 \oplus f(x)\rangle)$$

Para cada x, $f(x)$ es 0 o 1, con lo cual, el estado anterior resulta en:

$$|\psi_1\rangle = \frac{1}{\sqrt{2}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle(|0\rangle - |1\rangle)$$

El cúbit auxiliar, que se encuentra en el estado $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, se ignora:

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle$$

Finalmente se aplican puertas de Hadamard a cada cúbit del primer registro, para obtener:

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \left[\frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} (-1)^{x \cdot y} |y\rangle \right] = \frac{1}{2^n} \sum_{y=0}^{2^n-1} \left[\sum_{x=0}^{2^n-1} (-1)^{f(x)} (-1)^{x \cdot y} \right] |y\rangle$$

donde $x \cdot y = x_0 y_0 \oplus x_1 y_1 \oplus \cdots \oplus x_{n-1} y_{n-1}$ es la suma del producto bit a bit y $\oplus$ es la suma módulo 2.

Finalmente, la probabilidad de medir $|0\rangle^{\otimes n}$ en la base computacional, viene dada por la expresión:

$$P(|0\rangle^{\otimes n}) = \left| \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \right|^2,$$

Y se evalúa como 1 si $f(x)$ es constante (interferencia constructiva) o 0 si f es balanceada (interferencia destructiva). De otra forma, la medición final será $|0\rangle^{\otimes n}$, si $f(x)$ es constante o producirá algún otro estado si es balanceada.

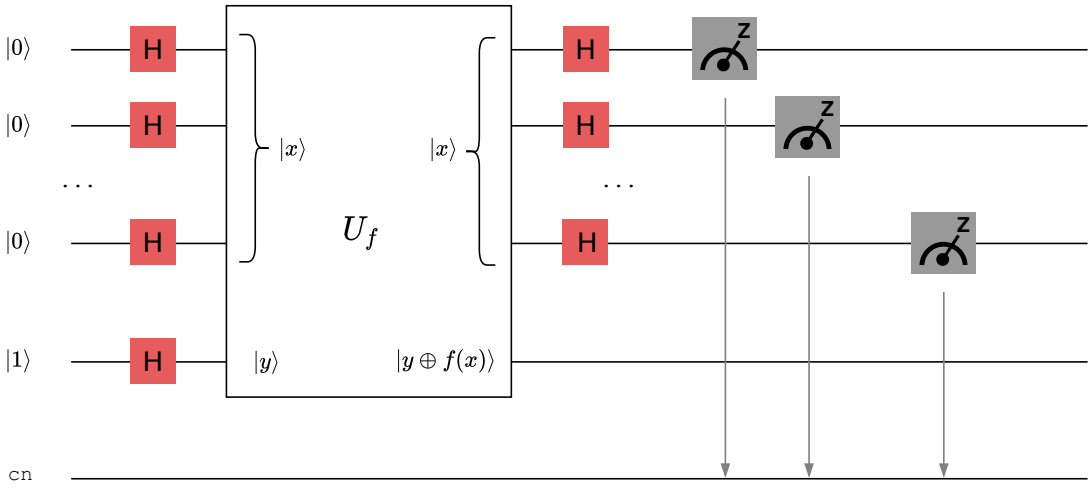


Figura 1: Algoritmo de Deutsch-Jozsa. Elaboración propia.

Cuando el oráculo es constante, no tiene ningún efecto en los cúbits de entrada, y el estado, antes y después de aplicar el oráculo es el mismo. Dado que la puerta H es su propia inversa, al aplicar de nuevo las puertas de Hadamard se obtiene el estado inicial $|0\rangle^{\otimes n}$ en el primer registro.

$$H^{\oplus n} \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2^n}} \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \xrightarrow{U_f} H^{\oplus n} \frac{1}{\sqrt{2^n}} \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Cuando el oráculo es balanceado, tras aplicar las puertas de Hadamard a la configuración inicial se obtiene un registro de entrada en superposición uniforme de todos los vectores de la base computacional. Al aplicar el oráculo balanceado como consecuencia del *Phase Kickback*, se añade una fase negativa a, exactamente, la mitad de

los estados.

$$H^{\oplus n}\begin{bmatrix}1\\0\\0\\\vdots\\0\end{bmatrix} = \frac{1}{\sqrt{2^n}}\begin{bmatrix}1\\1\\1\\\vdots\\1\end{bmatrix} \xrightarrow{U_f} \frac{1}{\sqrt{2^n}}\begin{bmatrix}-1\\1\\-1\\\vdots\\1\end{bmatrix}$$

El estado tras aplicar el oráculo es ortogonal al estado previo a la aplicación del oráculo, así, al aplicar las puertas de Hadamard, el estado final es ortogonal a $|0\rangle^{\otimes n}$, es decir nunca se observará el estado $|0\rangle^{\otimes n}$.

Este algoritmo cuántico resuelve el problema de Deutsch-Jozsa con una sola consulta al oráculo (U_f), mientras que cualquier algoritmo clásico debe evaluarlo, al menos, $2^{n-1}+1$ veces para resolver el problema con certeza. Por lo tanto, existe una ventaja exponencial entre la complejidad de la consulta de este algoritmo cuántico y la complejidad de la consulta de cualquier posible algoritmo clásico que resuelva ese problema con certeza. Existen algoritmos clásicos que resuelven este problema en menos evaluaciones, pero solo con alta probabilidad de éxito, no con certeza.

A continuación se muestra como crear oráculos que implementan funciones constantes y funciones balanceadas. El caso de las funciones constantes es muy simple ya que bastaría con aplicar la puerta identidad al segundo registro, el cúbit auxiliar, cuando $f(x) = 0$ o aplicar una puerta NOT para el caso en el que $f(x) = 1$.

Para crear un oráculo balanceada existen muchos posibles circuitos, una forma de asegurar una función balanceada es aplicar una puerta $CNOT$ para cada cúbit en el primer registro siendo el cúbit de control el cúbit del primer registro y el cúbit objetivo el auxiliar.

Se pueden crear variaciones simplemente colocando puertas NOT en distintos cúbits del registro de entrada, antes y después del grupo de puertas $CNOT$. La siguiente figura muestra algunos ejemplos.

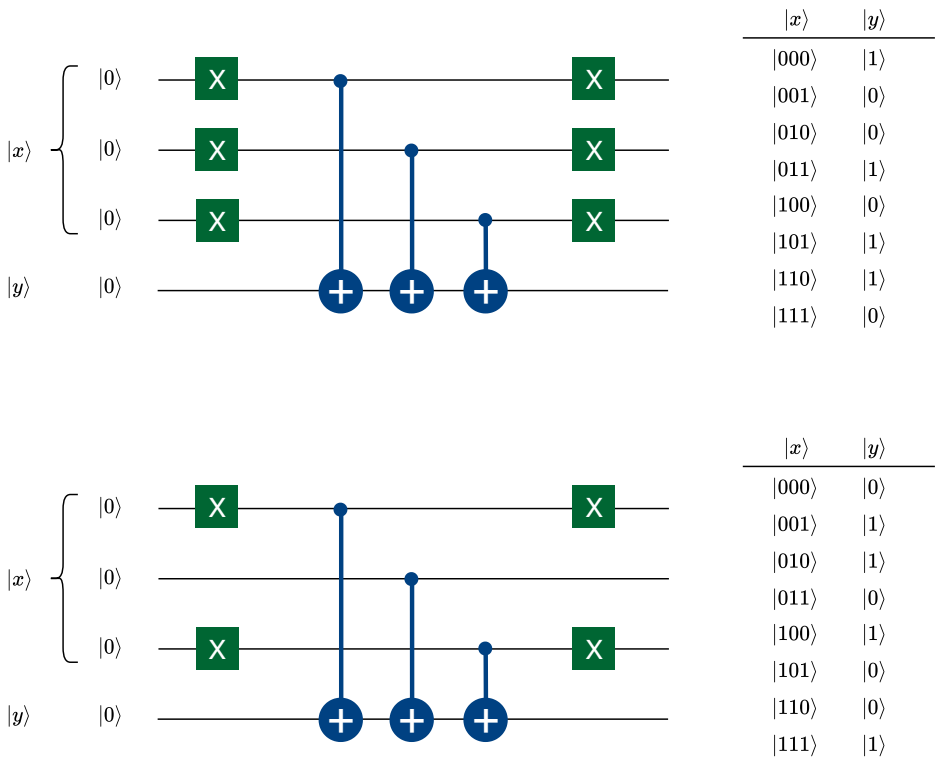


Figura 2: Varios oráculos balanceados para el algoritmo de Deutsch-Jozsa. Elaboración propia.

Finalmente una implementación de ejemplo del algoritmo de Deutsch-Jozsa para el caso de tres cúbit y una función balanceada.

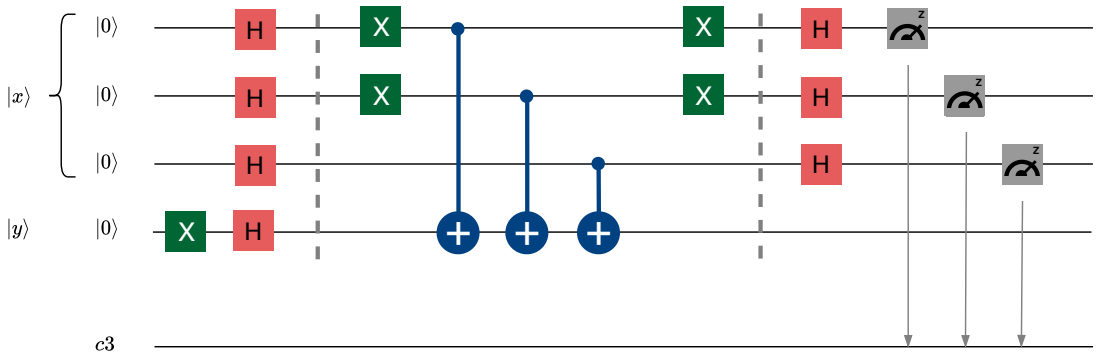


Figura 3: Implementación del Algoritmo de Deutsch-Jozsa para un oráculo balanceado. Elaboración propia.

Algoritmo de Bernstein-Vazirani

El algoritmo de Bernstein-Vazirani, es un algoritmo cuántico desarrollado por Ethan Bernstein y Umesh Vazirani en 1992.

El problema que plantea, es una versión restringida del algoritmo de Deutsch–Jozsa del apartado anterior donde, en lugar de distinguir entre dos clases diferentes de funciones, su objetivo es descubrir una cadena codificada en una función.

El problema que plantea es el siguiente, dado un oráculo que implementa una función $f : \{0,1\}^n \rightarrow \{0,1\}$ para la que se asegura que $f(x)$ es el producto escalar módulo 2 entre x y una cadena secreta $s \in \{0,1\}^n$, $f(x) = x \cdot s(mod\ 2) = x_1 s_1 \oplus x_2 s_2 \oplus \cdots \oplus x_n s_n$, encontrar s.

El circuito clásico reversible que implementa el oráculo es el representado en la siguiente figura:

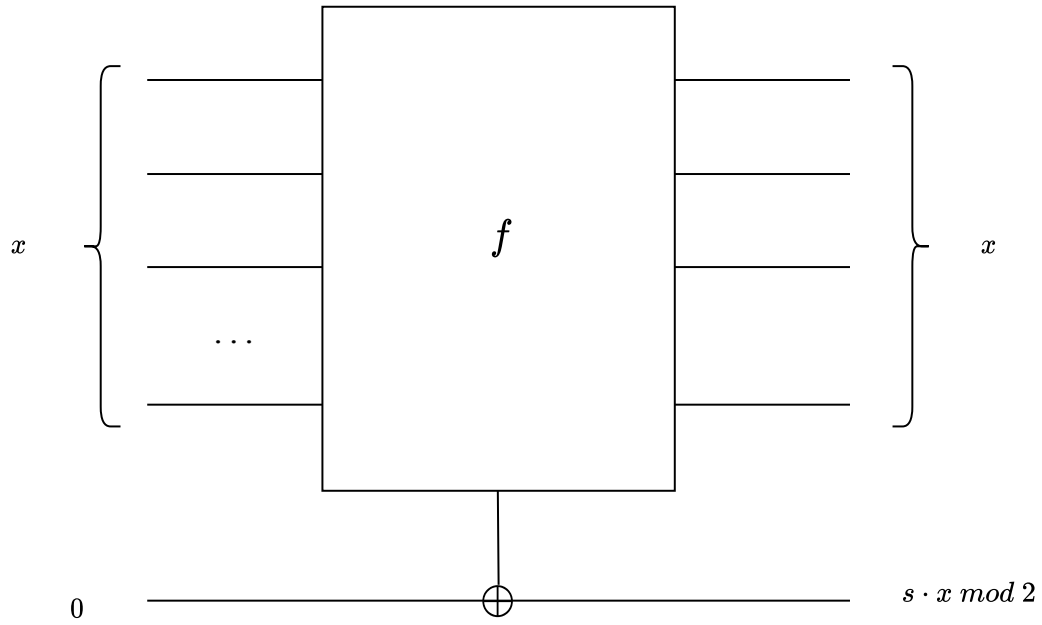


Figura 4: Oráculo clásico para el algoritmo de Bernstein-Vazirani. Elaboración propia.

De forma clásica, el método más eficiente para encontrar la cadena secreta es evaluar la función n veces con valores de entrada $x = 2^i$ para todo $i \in \{0, 1, 2, ..., n-1\}$:

$f(0...00001_n) = s_1$

$f(0...00010_n) = s_2$

$f(0...00100_n) = s_3$

...

$f(1...0000_n) = s_n$

Por ejemplo, si $s = 010$, entonces, con tres consultas obtendríamos la cadena:

$f(001) = (0 \cdot 0) \oplus (0 \cdot 1) \oplus (1 \cdot 0) = 0 + 0 + 0 = 0$

$f(010) = (0 \cdot 0) \oplus (1 \cdot 1) \oplus (0 \cdot 0) = 0 + 0 + 0 = 1$

$f(100) = (1 \cdot 0) \oplus (0 \cdot 1) \oplus (0 \cdot 0) = 0 + 0 + 0 = 0$

A diferencia de la solución clásica, que necesita n consultas al oráculo para encontrar la cadena secreta con certeza, el algoritmo cuántico tan solo necesita una consulta en superposición para determinar la cadena de forma exacta, es decir su complejidad de consulta es 1. El algoritmo es muy simple y se construye de la siguiente forma:

Se inicializan todos los cúbits de entrada al estado $|0\rangle^{\otimes n}$ y el cúbit auxiliar, de salida, al estado $|-\rangle$

Se aplican puertas Hadamard a todo el registro de entrada

Se aplica el oráculo

Se aplican puertas Hadamard a todo el registro de entrada

Se realiza el proceso de medida

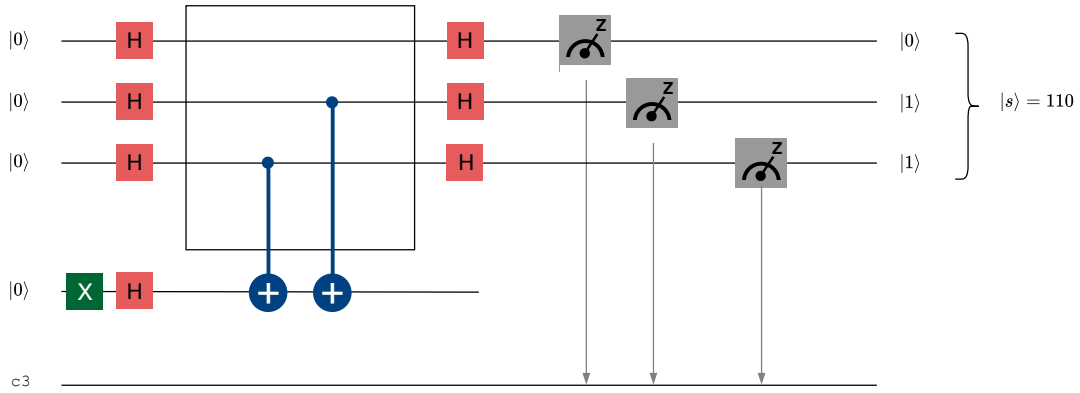


Figura 5: Algoritmo de Bernstein-Vazirani para 3 cúbits donde la cadena s=110. Elaboración propia.

La medida, actúa de la forma habitual y calcula el valor de la función en el cúbit auxiliar.

En primer lugar se inicializan los $n+1$ cúbits al estado: $|1\rangle|0\rangle^{\otimes n}$

Si se aplican puertas Hadamard a los cúbits de un estado $|a\rangle$ de n cúbits, se obtiene el estado siguiente:

A continuación se aplican puertas de Hadamard a todos los cúbits inicializados a $|0\rangle$, excepto el cúbit auxiliar, inicializado a $|1\rangle$, obteniendo el estado: $|a\rangle \xrightarrow{H^{\oplus n}} \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{a\cdot x}|x\rangle$

En el caso en el que $|a\rangle = |00..0\rangle$ la transformación anterior produce el estado siguiente, ya que el término de fase $(-1)^{a\cdot x}$ desaparece ya que $a = 0$ y por tanto $(-1)^{a\cdot x} = 1$:

$|a\rangle \xrightarrow{H^{\oplus n}} \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$

El oráculo clásico devuelve 1 para una entrada tal que $s \cdot x \; mod \; 2 = 1$ y devuelve 0 en cualquier otro caso. En el algoritmo cuántico, al aplicar el oráculo actuando sobre el cúbit auxiliar en estado $|-\rangle$ se obtiene el siguiente estado:

$|x\rangle \xrightarrow{f_s} (-1)^{s\cdot x}|x\rangle$

Por tanto, para la entrada $|00..0\rangle$ el resultado sería:

$|00...0\rangle \xrightarrow{H^{\oplus n}} \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \xrightarrow{f_s} \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{s\cdot x}|x\rangle$

Puesto que la puerta de Hadamard es su propia inversa, aplicando puertas de Hadamard al registro x se obtiene finalmente:

$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{s\cdot x}|x\rangle \xrightarrow{H^{\oplus n}} |s\rangle$

Realizando el proceso de medida en la base computacional se obtiene la cadena secreta s.

Algoritmo de Simon

El problema de Simon se enmarca dentro del grupo de algoritmos donde se estudia la complejidad de la consulta, fue concebido por Daniel Simon en 1994. Simon exhibió un algoritmo cuántico que resuelve el problema de Simon exponencialmente más rápido y con exponencialmente menos consultas que el mejor algoritmo clásico probabilístico (o determinista). En particular, el algoritmo de Simon utiliza un número lineal de consultas y cualquier algoritmo probabilístico clásico debe utilizar un número exponencial

de consultas.

Debido a que este problema asume la existencia de un oráculo, este algoritmo es de escaso valor práctico, pero, sin embargo, permite probar la existencia de una mejora exponencial.

El problema de Simon se describe de la siguiente forma: dada una función f, implementada como un oráculo, tal que: $f : \{0,1\}^n \rightarrow \{0,1\}^n$ y para la cual se asegura que existe una cadena $s \in \{0,1\}^n$ para todas las $x, y \in \{0,1\}^n$ donde:

$f(x) = f(y)$ si y solo si $x \oplus y \in \{0^n, s\}$

Se recuerda que, por ejemplo, la colección de cadenas $\{0,1\}^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$

El objetivo es identificar la cadena s minimizando las consultas al oráculo $f(x)$.

Una forma alternativa de enunciar el problema consiste en distinguir entre el caso en el que la función es 1:1 (es decir, cada entrada distinta tiene una salida distinta) y $s = 0^n$ del caso donde la función es 2:1 (es decir, una función que asigna dos entradas diferentes a una misma salida) y $s \neq 0^n$ satisfaciendo que $f(x) = f(x \oplus s)$.

El siguiente es un ejemplo para n=3 que satisface la propiedad mencionada.

$f(000) = f(110) = 101$

$f(001) = f(111) = 010$

$f(010) = f(100) = 000$

$f(011) = f(101) = 110$

En este caso, la cadena buscada es $s = 110$ y puede verificarse ya que cada resultado de f tiene dos ocurrencias y las dos cadenas de entrada correspondientes a una misma cadena de salida, al realizar XOR produce $s = 110$, la solución buscada. La función es de tipo, por tanto dos a una pues $s \neq 0$ frente al caso de una función una a una donde $f(x) = f(y)$ como en el caso $f(1) = 1, f(2) = 2, f(3) = 3$ ...

Se trata de un problema exponencial, muy difícil de resolver de forma clásica pues es necesario encontrar dos entradas diferentes x e y para las cuales $f(x) = f(y)$. Puesto

que no hay necesariamente una estructura en la función f que ayude a encontrar dos de estas entradas, es necesario probar $2^{n+1}+1$ entradas para encontrar s con certeza. El algoritmo cuántico de Simon permite encontrar s utilizando menos consultas que cualquier método clásico, en concreto con $O(n)$.

El objetivo, por tanto, del problema es encontrar $s \neq 0^n$. Se utilizan dos registros, como se puede observar en la siguiente figura.

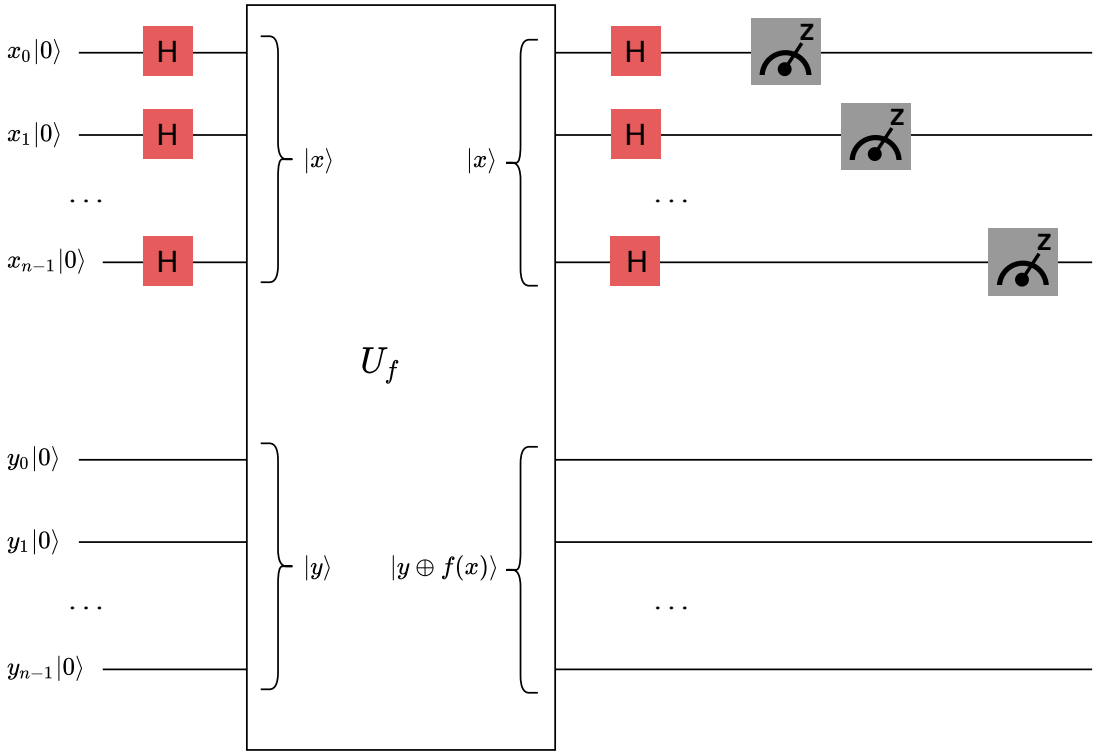


Figura 6: Algoritmo de Simon. Elaboración propia.

El registro x es necesario pues $f : \{0,1\}^n \rightarrow \{0,1\}^n$ y otro registro auxiliar y. Por tanto son necesarios 2^n cúbits y será necesario iterar el algoritmo varias veces.

Inicialmente el estado del sistema se encuentra en $|\psi\rangle = |0\rangle^{\otimes n} \otimes |0\rangle^{\otimes n}$, al aplicar las puertas de Hadamard al registro x el sistema evolucionara al estado:

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

Es decir, una superposición uniforme del registro x combinado con un registro y en estado $0^{\otimes n}$. A continuación se aplica el oráculo, la función $f : |x\rangle \oplus |y\rangle \longrightarrow |x\rangle \otimes |y \oplus f(x)\rangle$ aplicada al estado anterior resulta en un nuevo estado $|x\rangle \otimes |f(x)\rangle$ ya que $|y\rangle = |0\rangle^{\otimes n}$ y el circuito finaliza aplicando puertas de Hadamard de nuevo al registro

x, se obtiene el estado final:

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \otimes |f(x)\rangle \xrightarrow{H^{\oplus n}} \frac{1}{2^n} \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^n-1} (-1)^{x\cdot y} |y\rangle \otimes |f(x)\rangle =$$

$$\sum_{y=0}^{2^n-1} |y\rangle \oplus \left[\frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{x\cdot y} |f(x)\rangle \right]$$

Por lo tanto, de la expresión anterior, la probabilidad de observar una $|y\rangle$ es el cuadrado de la magnitud del coeficiente asociado, de otra forma, se observará $|y\rangle$ con probabilidad:

$$\left\| \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{x\cdot y} |f(x)\rangle \right\|^2$$

Y por la regla de Born, la suma de los cuadrados de los valores absolutos de los coeficientes deberán sumar 1, es decir:

$$\sum_{y=0}^{2^n-1} \sum_{x=0}^{2^n-1} \left| \frac{1}{2^n} (-1)^{x\cdot y} \right|^2 = 1$$

En el caso en el que $s = 0$, es decir f es 1:1, debido a que hay un número finito de valores, cuando se itera sobre todos los valores de x, también se hace sobre todos los valores de $f(x)$, solo que en un orden diferente. Tomando la suma de los cuadrados de los coeficientes para todos los $|x\rangle$ para obtener el cuadrado de la longitud, se obtiene el mismo valor que la suma de los cuadrados de los coeficientes cuando usamos $|f(x)\rangle$. En otras palabras, dos vectores que solo se diferencian en una permutación de sus coeficientes tienen la misma longitud. Para cada $|f(x)\rangle$se eleva al cuadrado cada coeficiente $\frac{1}{2^n}(-1)^{x\cdot y}$ para obtener $\frac{1}{2^{2n}}$ y sumando todos los 2^n de $|f(x)\rangle$.

Por tanto: $\left\| \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{x\cdot y} |f(x)\rangle \right\|^2 = 2^n \frac{1}{2^{2n}} = \frac{1}{2^n}$ y se puede concluir que para el caso en el que $s = 0$, la probabilidad de observar cualquiera de los 2^n de $|y\rangle$ en el proceso de medida es $\frac{1}{2^n}$. Si se ejecuta el circuito muchas veces y se observa esta distribución uniforme de $|y\rangle$ indicaría que $s = 0$.

Aunque es trivial en este caso, debido a que $s = 0$, esta distribución uniforme observada es igual a todo $|y\rangle$ tal que $y \cdot r = 0$.

Para el caso en el que $s \neq 0$ la función f es $2 : 1$. Mientras que x puede tomar cualquiera de los 2^n valores en $\{0,1\}^n$, $f(x)$ solamente podrá tomar la mitad de esos valores que se expresa como el conjunto $\{0,1\}_f^n$ o rango de f.

Para uno de esos valores z en el rango de f existen dos cadena binarias de longitud n que tiene asignadas: x_z y su compañera $x_z \oplus s$.

La probabilidad de observar un estado $|y\rangle$ particular al realizar el proceso de medida es, todavía:

$$\left\| \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{x \cdot y} |f(x)\rangle \right\|^2 = \left\| \frac{1}{2^n} \sum_{z \in \{0,1\}_f^n} \left[(-1)^{x_z \cdot y} + (-1)^{(x_z \oplus s) \cdot y} \right] |z\rangle \right\|^2 =$$

$$\left\| \frac{1}{2^n} \sum_{z \in \{0,1\}_f^n} (-1)^{x_z \cdot y} + (1 + (-1)^{s \cdot y}) |z\rangle \right\|^2 = \begin{cases} 0, & s \cdot y = 1 \\ \frac{1}{2^{n-1}}, & s \cdot y = 0 \end{cases}$$

La distribución uniforme de estados observados en este caso son aquellos para los que $y \cdot s = 0$.

$y_1 \cdot s = 0$

$y_2 \cdot s = 0$

...

$y_{n-1} \cdot s = 0$

La medición de este estado da como resultado una serie de estados aleatoria que satisfacen tal que $y \cdot s = 0 \ mod \ 2$. Este cálculo se repite hasta que se han encontrado n ecuaciones linealmente independientes. Cada vez que se repite el cálculo, la ecuación resultante tiene al menos un 50 por ciento de posibilidades de ser linealmente independiente de las ecuaciones anteriores obtenidas. Después de repetir el cálculo 2^n veces, hay un 50 por ciento de probabilidad de que se hayan encontrado n ecuaciones linealmente independientes. Estas ecuaciones se pueden resolver para encontrar s en $O(n^2)$ pasos. Por lo tanto, con alta probabilidad, la cadena buscada s se encontrará con $O(n)$ llamadas a U_f, seguidas de $O(n^2)$ pasos para resolver el conjunto de ecuaciones resultante.

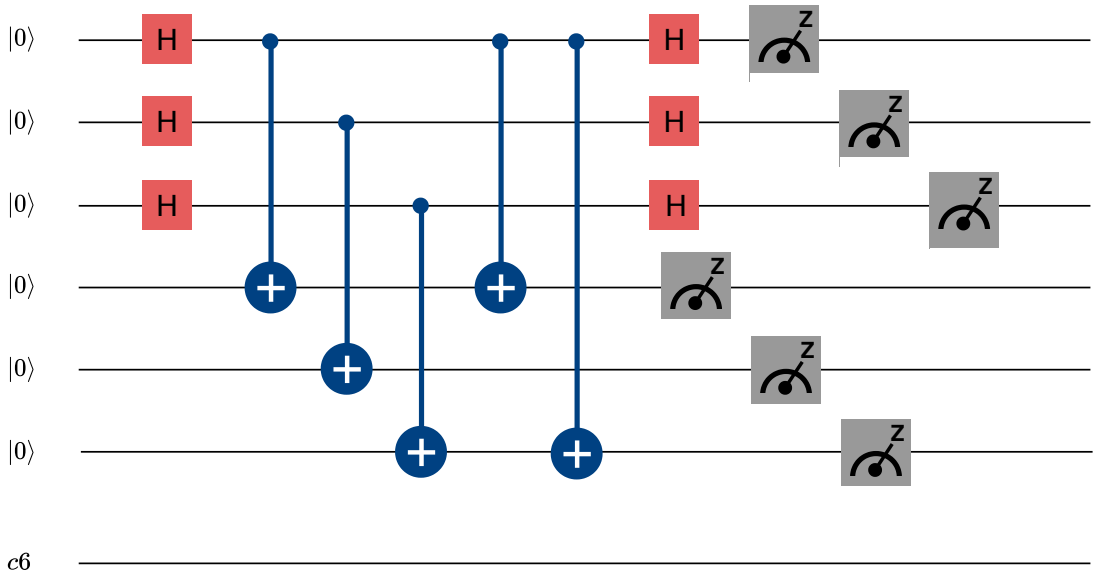


Figura 7: Algoritmo de Simon para 3 cúbits con s=101. Elaboración propia.

Transformada cuántica de Fourier

La transformada de Fourier tiene multitud de aplicaciones y versiones en la computación clásica, desde el procesamiento de señales hasta la compresión de datos y la teoría de la complejidad. La transformada cuántica de Fourier (QFT) es la implementación cuántica de la transformada discreta de Fourier sobre las amplitudes de una función de onda. Es parte de muchos algoritmos cuánticos, sobre todo el algoritmo de factorización de Shor y el de estimacion de fase (QPE, Quantum Phase Estimation).

La transformación cuántica de Fourier (QFT) está basada en la transformación discreta de Fourier (DFT) clásica y su implementación eficiente, la transformada rápida de Fourier (FFT). A continuación se describe brevemente la transformada de Fourier discreta clásica (DFT) y la transformada de Fourier rápida (FFT) antes de describir la transformada cuántica de Fourier (QFT) y su implementación cuántica sorprendentemente eficiente.

La transformada discreta de Fourier (DFT) transforma una función matemática representada en el dominio del tiempo en otra representada en el dominio de la frecuencia.

La DFT transforma un vector de N números complejos $(x_0, ..., x_{N-1})$ en otro vector de N números complejos $(y_0, ..., y_{N-1})$ mediante la fórmula:

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \omega_N^{jk}$$

donde:

$$\omega_N^{jl} = e^{2\pi i \frac{jk}{N}}$$

La transformada discreta de Fourier se puede ver como una transformación lineal:

$$\begin{bmatrix} x(0) \\ x(1) \\ \dots \\ x(N-1) \end{bmatrix} \xrightarrow{F} \begin{bmatrix} y(0) \\ y(1) \\ \dots \\ y(N-1) \end{bmatrix}$$

De forma similar la transformada cuántica de Fourier actúa sobre un estado $|X\rangle = \sum_{j=0}^{N-1} x_j |j\rangle$ y lo asigna al estado $|Y\rangle = \sum_{k=0}^{N-1} y_k |k\rangle$ según la formula:

$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \omega_N^{jk}$ donde $\omega_N^{jk} = e^{2\pi i \frac{jk}{N}}$

Que se puede expresar como: $|j\rangle \xrightarrow{QFT} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{jk}$ donde la transformación unitaria QFT tiene la representación matricial siguiente:

$$QFT = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} \omega_N^{jk} |k\rangle\langle j|$$

Observar que solo las amplitudes de esta transformación se ven afectadas por ella.

La transformada cuántica de Fourier (QFT) realiza un cambio de base, transformando el estado expresado en la base computacional al estado correspondiente en la base de Fourier. La puerta de Hadamard es la versión de un cúbit de la transformada cuántica de Fourier, transformando de la base computacional $|0\rangle, |1\rangle$ a la base de Hadamard $|+\rangle, |-\rangle$ y de la misma forma, cualquier estado de n cúbits en la base computacional tiene un estado correspondiente en la base de Fourier. Normalmente se utiliza la tilde para indicar que el estado está expresado en la base de Fourier: $QFT|x\rangle = |\tilde{x}\rangle$

Utilizando la base computacional, un número binario se almacena en un registro de cúbits utilizando la base computacional, así, utilizando un registro de cuatro cúbits, el número 5 se representa como $|0\rangle|1\rangle|0\rangle|1\rangle$, el número 7 se representa como $|0\rangle|1\rangle|1\rangle|1\rangle$, etc.

Se puede observar que la frecuencia con la que cambian los estados de los cuatro cúbits a medida que vamos recorriendo los valores del 0 al 15 es distinta para cada uno de ellos, el cúbit que representa al dígito menos significativo cambia cada vez que incrementamos en 1 el valor a representar, el siguiente cambia cada dos incrementos, el siguiente cada cuatro y el de mayor peso cambia cada 8 incrementos. Esta progresión continuaría de la misma forma para números representados con n cúbits, de forma que el dígito n cambiaria cada 2^n incrementos.

En la base de Fourier la información se almacena de forma diferente, se almacenan en la fase mediante rotaciones alrededor del eje Z. El número que se desea almacenar determina el ángulo de la fase en el que cada cúbit se deberá rotar alrededor del eje Z. Así, para representar el número 5 se deberá rotar el cúbit correspondiente al dígito de menos peso $\frac{5}{2^n}2\pi = \frac{5}{16}2\pi$ radianes. El siguiente se rotara el doble del anterior, es decir, $\frac{10}{16}2\pi = \frac{10}{8}\pi = \frac{5}{4}\pi$, el siguiente el doble, $\frac{20}{16}2\pi$ y el cuarto, el correspondiente al dígito más significativo se deberá rotar el doble del anterior, es decir $\frac{40}{16}2\pi = 5\pi = \pi$. Al igual que ocurre con la base computacional, en este caso también los cambios en los cúbits tienen diferente frecuencia a medida que se va incrementando el número a representar. El cúbit correspondiente al dígito menos significativo tiene la frecuencia más baja.

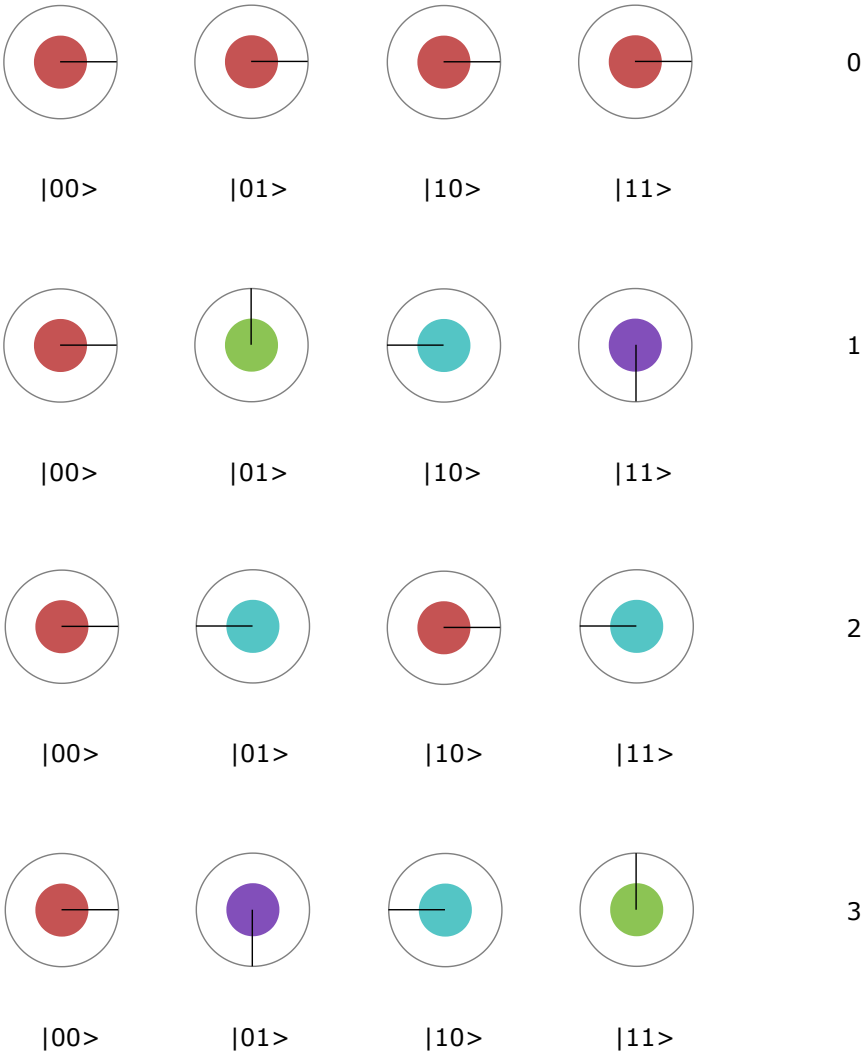


Figura 8: Representación de los números del 0 al 4 con dos cúbits, en la base de Fourier. Para el número 0, la fase asociada a cada elemento de la base es 0; para el número 1, la fase asociada a cada elemento de la base es la del elemento anterior más $\pi/2$; para el número 2, la fase asociada a cada elemento de la base es la del elemento anterior más π, finalmente para el número 3, la fase asociada a cada elemento de la base es la del elemento anterior más $3\pi/2$. Elaboración propia.

El ejemplo más simple posible es la transformada de Fourier cuántica actuando sobre un cúbit en el estado $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. En este caso, $x_0 = \alpha,\ x_1 = \beta$, and $N = 2$. Por lo tanto:

$$y_0 = \frac{1}{\sqrt{2}}\left[\alpha e^{(2\pi i \frac{0\cdot 0}{2})} + \beta e^{(2\pi i \frac{1\cdot 0}{2})}\right] = \frac{1}{\sqrt{2}}(\alpha + \beta)$$

$$y_1 = \frac{1}{\sqrt{2}}\left[\alpha e^{(2\pi i \frac{0\cdot 1}{2})} + \beta e^{(2\pi i \frac{1\cdot 1}{2})}\right] = \frac{1}{\sqrt{2}}(\alpha - \beta)$$

Por lo tanto, el estado final será:

$$QFT|\psi\rangle = \frac{1}{\sqrt{2}}(\alpha+\beta)|0\rangle + \frac{1}{\sqrt{2}}(\alpha-\beta)|1\rangle$$

Lo cual corresponde exactamente con la transformada de Hadamard actuando sobre el cúbit donde se puede ver como la puerta H realiza la transformada de Fourier sobre las amplitudes del estado para $N = 2$.

$$H|\psi\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}\alpha\\ \beta\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}\alpha+\beta\\ \alpha-\beta\end{bmatrix} = \tilde{\alpha}|0\rangle + \tilde{\beta}|1\rangle$$

A continuación se estudia la transformada cuántica de Fourier para el caso general de $N = 2^{n,}$ donde n es el numero de cúbits.

Como se ha descrito anteriormente, QTF es un operador unitario que transforma el estado $|x\rangle = \sum_{j=0}^{N-1} x_j|j\rangle = |x_{n-1}, x_{n-2}, \ldots x_0\rangle$, en el estado $|y\rangle = \sum_{j=0}^{N-1} y_j|j\rangle = |y_{n-1}, y_{n-2}, \ldots y_0\rangle$ de la siguiente forma:

$$QFT_N|x\rangle = \frac{1}{\sqrt{N}}\sum_{y=0}^{N-1}\omega_N^{xy}|y\rangle = \frac{1}{\sqrt{N}}\sum_{y=0}^{N-1} e^{2\pi i\frac{xy}{2^n}}|y\rangle$$

$$= \frac{1}{\sqrt{N}}\sum_{y=0}^{N-1} e^{2\pi i x\sum_{k=0}^{n-1}\frac{y_k 2^k}{2^n}}|y\rangle = \frac{1}{\sqrt{N}}\sum_{y=0}^{N-1} e^{2\pi i x\sum_{k=0}^{n-1}\frac{y_k}{2^{n-k}}}|y\rangle$$

$$= \frac{1}{\sqrt{N}}\sum_{y=0}^{N-1} e^{2\pi i x(\frac{y_o}{2^{n-0}}+\frac{y_1}{2^{n-1}}+\frac{y_2}{2^{n-2}}+\cdots\frac{y_{n-1}}{2^{n-n+1}})}|y\rangle$$

$$= \frac{1}{\sqrt{N}}\sum_{y=0}^{N-1} e^{2\pi i x\frac{y_o}{2^n}} e^{2\pi i x\frac{y_1}{2^{n-1}}} e^{2\pi i x\frac{y_2}{2^{n-2}}}\ldots e^{2\pi i x\frac{y_{n-1}}{2}}|y\rangle = \frac{1}{\sqrt{N}}\sum_{y=0}^{N-1}\prod_{k=0}^{n-1} e^{2\pi i x\frac{y_k}{2^{n-k}}}|y\rangle$$

Puesto que:

$$\sum_{y=0}^{N-1}|y\rangle = \sum_{y_{n-1}=0}^{1}\sum_{y_{n-1}=0}^{1}\sum_{y_{n-2}=0}^{1}\cdots\sum_{y_0=0}^{1}|y_{n-1}y_{n-2}y_{n-3\ldots}y_0\rangle$$

Se tiene que:

$$\frac{1}{\sqrt{N}}\sum_{y=0}^{N-1}\prod_{k=0}^{n-1} e^{2\pi i x\frac{y_k}{2^{n-k}}}|y\rangle = \sum_{y_{n-1}=0}^{1}\cdots\sum_{y_0=0}^{1}\frac{1}{\sqrt{N}}\prod_{k=0}^{n-1} e^{2\pi i x\frac{y_k}{2^{n-k}}}|y_{n-1}y_{n-2}y_{n-3\ldots}y_0\rangle$$

$$= \sum_{y_{n-2}=0}^{1}\cdots\sum_{y_0=0}^{1}\left[\frac{1}{\sqrt{N}}\prod_{k=0}^{n-2} e^{2\pi i x\frac{y_k}{2^{n-k}}} e^{2\pi i x\frac{y_{n-1}}{2^{n-(n-1)}}}|0y_{n-2}y_{n-3\ldots}y_0\rangle+\right.$$

$$\left.+\frac{1}{\sqrt{N}}\prod_{k=0}^{n-2} e^{2\pi i x\frac{y_k}{2^{n-k}}} e^{2\pi i x\frac{y_{n-1}}{2^{n-(n-1)}}}|1y_{n-2}y_{n-3\ldots}y_0\rangle\right]$$

$$= \sum_{y_{n-2}=0}^{1}\cdots\sum_{y_0=0}^{1}\left[\frac{1}{\sqrt{N}}\prod_{k=0}^{n-2} e^{2\pi i x\frac{y_k}{2^{n-k}}} e^{2\pi i x\frac{0}{2^{n-(n-1)}}}|0y_{n-2}y_{n-3\ldots}y_0\rangle+\right.$$

$$+\frac{1}{\sqrt{N}}\prod_{k=0}^{n-2} e^{2\pi i x \frac{y_k}{2^{n-k}}} e^{2\pi i x \frac{1}{2^{n-(n-1)}}} |1 y_{n-2} y_{n-3} \ldots y_0\rangle\Big]$$

$$= \sum_{y_{n-2}=0}^{1} \cdots \sum_{y_0=0}^{1} \Big[\frac{1}{\sqrt{N}}\prod_{k=0}^{n-2} e^{2\pi i x \frac{y_k}{2^{n-k}}} |0 y_{n-2} y_{n-3} \ldots y_0\rangle +$$

$$+\frac{1}{\sqrt{N}}\prod_{k=0}^{n-2} e^{2\pi i x \frac{y_k}{2^{n-k}}} e^{\frac{2\pi i x}{2}} |1 y_{n-2} y_{n-3} \ldots y_0\rangle\Big]$$

$$= \frac{1}{\sqrt{N}}(|0\rangle + e^{\frac{2\pi i x}{2}}|1\rangle) \otimes \sum_{y_{n-2}=0}^{1} \cdots \sum_{y_0=0}^{1} \prod_{k=0}^{n-2} e^{2\pi i x \frac{y_k}{2^{n-k}}} |y_{n-2} y_{n-3} \ldots y_0\rangle$$

$$= \frac{1}{\sqrt{N}}(|0\rangle + e^{\frac{2\pi i x}{2}}|1\rangle) \otimes (|0\rangle + e^{\frac{2\pi i x}{2^2}}|1\rangle) \otimes (|0\rangle + e^{\frac{2\pi i x}{2^3}}|1\rangle) \ldots \otimes (|0\rangle + e^{\frac{2\pi i x}{2^n}}|1\rangle)$$

Expandiendo x :

$$= \frac{1}{\sqrt{N}}(|0\rangle + e^{2\pi i(\frac{x_0}{2} + x_1 + 2x_2 + 2^2 x_3 + \ldots)}|1\rangle) \otimes (|0\rangle + e^{2\pi i(\frac{x_0}{4} + \frac{x_1}{2} + x_2 + 2x_3 + \ldots)}|1\rangle) \otimes (|0\rangle + e^{2\pi i(\frac{x_0}{8} + \frac{x_1}{4} + \frac{x_2}{2} + x_3 + \ldots)}|1\rangle) \otimes \ldots$$

Como las rotaciones $2\pi \cdot n$ no tienen efecto, ya que $e^{2\pi n} = 1$, se sigue que:

$$= \frac{1}{\sqrt{N}}(|0\rangle + e^{2\pi i \frac{x}{2}}|1\rangle) \otimes (|0\rangle + e^{2\pi i \frac{x}{2^2}}|1\rangle) \otimes (|0\rangle + e^{2\pi i \frac{x}{2^3}}|1\rangle) \otimes \ldots$$

La expresión anterior, representada gráficamente, corresponde a la siguiente secuencia:

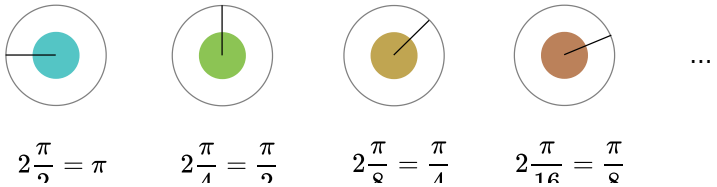


Figura 9: Representación gráfica de las rotaciones. Elaboración propia.

Expandiendo X se llega finalmente a la expresión:

$$QFT_N|x\rangle = \frac{1}{\sqrt{N}}(|0\rangle + e^{2\pi i \frac{x_0}{2}}|1\rangle) \otimes (|0\rangle + e^{2\pi i(\frac{x_0}{2^2} + \frac{x_1}{2})}|1\rangle) \otimes (|0\rangle + e^{2\pi i(\frac{x_0}{2^3} + \frac{x_1}{2^2} + \frac{x_2}{2})}|1\rangle) \otimes \ldots \otimes (|0\rangle + e^{2\pi i(\frac{x_0}{2^n} + \frac{x_1}{2^{n-1}} + \ldots + \frac{x_{n-1}}{2})}|1\rangle)$$

A continuación se describe como trasladar la anterior expresión a un circuito.

Puesto que:

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{\pi i x_k}|1\rangle) = \begin{cases} \frac{1}{\sqrt{2}}(|0\rangle + e^{0}|1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), & x_k = 0 \\ \frac{1}{\sqrt{2}}(|0\rangle + e^{\pi i}|1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) & x_k = 1 \end{cases}$$

Es justamente el comportamiento de la puerta de Hadamard actuando sobre x_k:

$$\boldsymbol{H}|z_k\rangle = \begin{cases} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle, & x_k = 0 \\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle, & x_k = 1 \end{cases}$$

Por otro lado, la puerta de fase $\boldsymbol{P}(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}$ actúa, asignando:

$|0\rangle \longrightarrow |0\rangle$

$|1\rangle \longrightarrow e^{i\theta}|1\rangle$

Así, para tres cúbits:

$|x\rangle = |x_2, x_1, x_0\rangle$

$$\boldsymbol{QFT}_8|x\rangle = \frac{1}{\sqrt{8}}(|0\rangle + e^{2\pi i \frac{x_0}{2}}|1\rangle) \otimes (|0\rangle + e^{2\pi i(\frac{x_0}{2^2} + \frac{x_1}{2})}|1\rangle) \otimes (|0\rangle + e^{2\pi i(\frac{x_0}{2^3} + \frac{x_1}{2^2} + \frac{x_1}{2})}|1\rangle)$$

El último término:

$$|0\rangle + e^{2\pi i(\frac{x_0}{2^3} + \frac{x_1}{2^2} + \frac{x_1}{2})}|1\rangle = |0\rangle + e^{2\pi i \frac{x_0}{2^3}} e^{2\pi i \frac{x_1}{2^2}} e^{2\pi i \frac{x_1}{2}}|1\rangle = |0\rangle + e^{\pi i \frac{x_0}{2^2}} e^{\pi i \frac{x_1}{2}} e^{\pi i x_2}|1\rangle$$

Y por tanto se puede representar como el circuito mostrado en la siguiente figura ya que el estado del cúbit x_2 debe rotar π radianes solo si es $|1\rangle$ lo cual es justamente el comportamiento que proporciona la puerta $\boldsymbol{H}$; deberá adicionalmente rotar $\pi/2$ radianes solo si es $|1\rangle$ el cúbit x_1 lo cual es el comportamiento que proporciona la puerta $\boldsymbol{P}(\pi/2)$ controlada por el cúbit x_1 y finalmente deberá rotar adicionalmente $\pi/4$ radianes solo si es $|1\rangle$ el cúbit x_0 lo cual, de nuevo, es el comportamiento que proporciona la puerta $\boldsymbol{P}(\pi/4)$ controlada por el cúbit x_0.

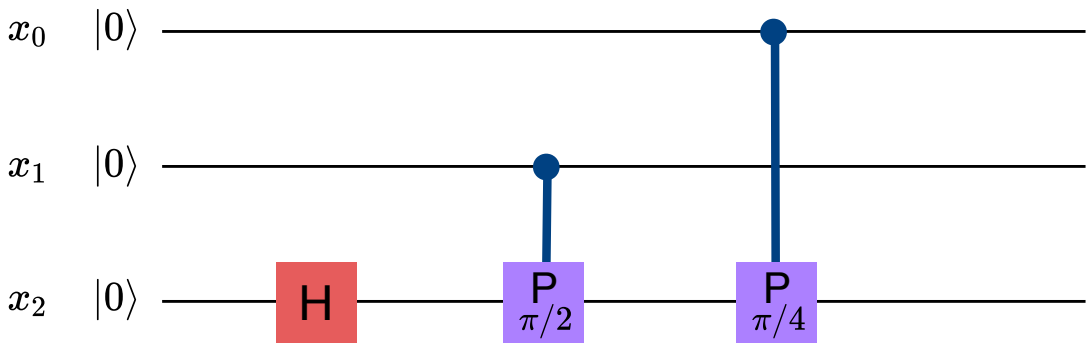


Figura 10: Representación como circuito del ultimo termino de la expresión correspondiente a la transformada cuántica de Fourier para tres cúbits. Elaboración propia.

Completando el circuito para el resto de términos siguiendo el mismo esquema se obtiene el circuito que proporciona la transformada cuántica de Fourier para tres cúbits que se muestra en la siguiente figura:

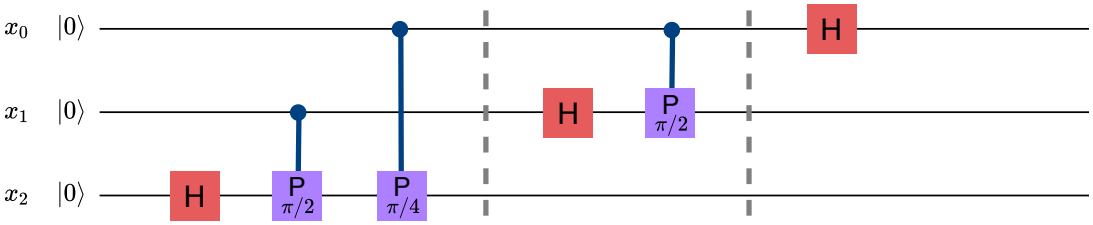


Figura 11: Circuito que implementa la transformada de Fourier para 3 cúbits. Elaboración propia.

9.3 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