Factorización de Enteros

El algoritmo de factorización de enteros es uno de los desarrollos más importantes de la computación cuántica. El protocolo RSA es el más usado para encriptar información importante como por ejemplo cuando se escribe un número de tarjeta de crédito en un sitio web. RSA se basa en la dificultad computacional de factorizar números grandes. Es decir que no se conoce un algoritmo clásico para encontrar los factores de un número de n bits que sea polinomial en n. Si se descubriese, se debilitaría la seguridad de RSA y habría que encontrar un nuevo protocolo en su reemplazo. Cuando Peter Shor descubrió que mediante una *computadora cuántica* es posible factorizar de manera eficiente grandes números se generó mucho interés en el potencial de la computación cuántica.
Como primer paso en el desarrollo del algoritmo de Shor vamos a plantear la equivalencia entre la *factorización* y la búsqueda del *orden* en enteros.

A. La Búsqueda del Orden

Los enteros *mod* N forman el conjunto $\{0, 1, \ldots, N-1\}$ que se conoce como $\mathbb{Z}_N$. Dos enteros se dice que son *equivalentes mod* N si N divide $s - t$ exactamente. En este caso se pone

$$s \equiv t (mod\ N).$$

Todo entero k puede ser *reducido mod* N tomando el resto r después de la división de k por N:

$$r = k\, mod\ N.$$

Una propiedad importante es que si $MCD(a, N) = 1$, el número 1 aparecerá en la secuencia $a\, mod\, N,\ a^2\, mod\, N,\ a^3\, mod\, N, \ldots$ y a partir de ahí la secuencia se repetirá en forma periódica.
Ejemplo
Tomemos la serie de potencias de 2

$$2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, \ldots.$$

Ahora veamos las potencias de 2 mod 15

$$2, 4, 8, 1, 2, 4, 8, 1, 2, 4, 8, \ldots$$

vemos que se repite con un período 4.

Definición

Dados los enteros a y N, que cumplen que $MCD(a, N) = 1$, el *orden* de $a \, mod \, N$ es el mínimo entero positivo r tal que $a^r \equiv 1 \, mod \, N$.

Aparentemente no es difícil encontrar el orden, simplemente se construye la serie modulo N y cuando se encuentra el 1, ya está. Sin embargo, el número de pasos para que suceda ésto puede ser tan grande como N que puede tener cientos o miles de dígitos. Ésta es la razón por la que clásicamente no se usa para factorizar números clásicamente.

Problema de Factorización de Enteros

Entrada: Entero N.

Problema: Encontrar los enteros positivos $p_1, p_2, \ldots, p_l, r_1, r_2, \ldots, r_l$ donde los p_i son primos distintos entre sí y $N = p_1^{r_1} p_2^{r_2} \ldots p_l^{r_l}$.

Supongamos que queremos factorizar el entero N. Suponemos primero que es impar porque es fácil eliminar los factores de 2. Como también es fácil factorizar potencias de un sólo primo, suponemos que N contiene por lo menos potencias de dos números primos distintos. Si podemos separar N en dos factores no triviales, entonces se puede encontrar los factores primos mediante algoritmos clásicos probabilísticos o deterministas de orden polinomial (aunque puede ser de grado alto en el caso determinista). De forma que el problema de factorizar N se puede reducir al problema de separación de factores en tiempos $O(log \, N)$.

Separación de Enteros

Entrada: Entero impar N con por lo menos dos factores primos distintos.

Problema: Encontrar dos enteros N_1 y N_2 tal que $N = N_1 \times N_2$.

Se demostró en 1975 que el problema de separación de enteros se puede *reducir de forma probabilística* al problema de *búsqueda del orden*. Ésto significa que si encontramos un algoritmo eficiente para búsqueda del orden, es posible dar un algoritmo probabilístico

eficiente para separar enteros.

Para separar N se comienza encontrando el orden de un entero cualquiera a que sea *coprimo* de N. Si a no es coprimo de N el $MCD(a, N)$ es un factor no trivial de N. Para encontrar el MCD se usa el algoritmo extendido de Euclides. De forma que se puede muestrear $\{2, 3, \ldots, N-2\}$ y testear con el algoritmo de Euclides para encontrar coprimos de N. Si encontramos un coprimo a ($MCD(a, N) = 1$), el orden r de a será par con probabilidad $\frac{1}{2}$. En ese caso $b = a^{r/2} mod N$ cumple que $b^2 - 1 = 0 mod N$, y por lo tanto N divide a $(b-1)(b+1)$. Se espera que $MCD(b-1, N)$ sea un factor no trivial de N. Si N tiene dos factores primos distintos, para un a con orden r par, la probabilidad de que $MCD(a^{r/2} - 1 mod N, N)$ sea un factor no trivial de N es de $\frac{1}{2}$. Queda claro entonces que sólo un número constante de valores de a deben ser probados para separar N con alta probabilidad.

Teorema

Sea r un entero positivo. Supongamos que los enteros k_1 y k_2 son seleccionados aleatoriamente de $\{0, 1, \ldots, r-1\}$. Sean c_1, r_1, c_2, r_2 enteros tales que $MCD(r_1, c_1) = MCD(r_2, c_2) = 1$ y $\frac{k_1}{r} = \frac{c_1}{r_1}$, $\frac{k_2}{r} = \frac{c_2}{r_2}$.

Entonces con probabilidad $\frac{6}{\pi^2}$ se tiene $r = MCM(r_1, r_2)$. El orden de tiempo del algoritmo para calcular r es $O(log^2 r)$.

De acuerdo a este teorema se puede reducir el problema de búsqueda del orden a buscar fracciones $\frac{k}{r}$ para enteros k seleccionados aleatoriamente de $\{0, 1, \ldots, r-1\}$.

Teorema de Fracciones Continuas

Todo número racional $\frac{x}{2^n}$ tiene una secuencia de $O(n)$ aproximaciones sucesivas, llamadas convergentes, $\frac{a_1}{b_1}, \frac{a_2}{b_2}, \ldots, \frac{a_m}{b_m}$, con $\frac{a_m}{b_m} = \frac{x}{2^n}$, con las propiedades:

- $a_1 < a_2 < \cdots < a_m, b_1, < b_2 < \cdots < b_m$.
- La lista de convergentes de $\frac{x}{2^n}$ se calcula en tiempo polinomial en n.
- Si alguna fracción $\frac{k}{r}$ satisface

$$|\frac{x}{2^n} - \frac{k}{r}| \leq \frac{1}{2r^2},$$

entonces $\frac{k}{r}$ estará en la lista de convergentes de $\frac{x}{2^n}$.

Este teorema muestra que se puede reducir la tarea de determinar exactamente la fracción $\frac{k}{r}$ a encontrar $\frac{x}{2^n}$ con $|\frac{x}{2^n} - \frac{k}{r}| \leq \frac{1}{2r^2}$.

Es importante notar que todas las reducciones de arriba son clásicas.

Algoritmo de Fracciones Continuas

El algoritmo de fracciones continuas es un método para encontrar las aproximaciones sucesivas a un número real. Vamos a dar un ejemplo. Supongamos que queremos descomponer 31/13 en fracciones continuas. El primer paso es separar 31/13 en partes entera y fraccionaria,

$$\frac{31}{13} = 2 + \frac{5}{13}.$$

Ahora invertimos la parte fraccionaria:

$$\frac{31}{13} = 2 + \frac{1}{\frac{13}{5}}.$$

Los pasos de separar e invertir son aplicados a 13/5:

$$\frac{31}{13} = 2 + \frac{1}{2 + \frac{3}{5}} = 2 + \frac{1}{2 + \frac{1}{\frac{5}{3}}}.$$

Sigue separar e invertir 5/3:

$$\frac{31}{13} = 2 + \frac{1}{2 + \frac{1}{1 + \frac{2}{3}}} = 2 + \frac{1}{2 + \frac{1}{1 + \frac{1}{\frac{3}{2}}}}.$$

Escribiendo 3/2 como 1 + 1/2 termina el desarrollo porque queda un 1 en el numerador sin necesidad de invertir:

$$\frac{31}{13} = 2 + \frac{1}{2 + \frac{1}{1 + \frac{1}{1 + \frac{1}{2}}}}.$$

El algoritmo termina despues de un número finito de pasos para un número racional. Si ese número se escribe como el cociente de enteros de L bits el algoritmo puede ser realizado en $O(L^3)$ operaciones, $O(L)$ de *separar e invertir* y $O(L^2)$ compuertas para aritmética elemental

Resumiendo, la búsqueda de orden r de a modulo N se reduce al problema de estimaciones de muestreo.

Estimaciones de Muestreo de un entero aleatorio múltiplo de $\frac{1}{r}$

Entrada: Enteros a y N tales que $MCD(a, N) = 1$. Sea r el orden (desconocido) de a.
Problema: Obtener $x \in \{0, 1, 2, \ldots, 2^{n-1}$ tal que para $k \in \{0, 1, \ldots, r-1\}$ se tiene

$$Pr\Big(|\frac{x}{2^n} - \frac{k}{r}| \leq \frac{1}{2r^2}\Big)$$

En el problema de muestreo es donde se usa un algoritmo cuántico.

B. Estimación de Autovalores para la Búsqueda de Orden

Sea U_a el operador que transforma

$$U_a : |s\rangle \rightarrow |sa \, mod \, N\rangle \, , \quad 0 \leq s < N.$$

Dado que $a^r \equiv 1(modN)$, tenemos

$$U_a^r : |s\rangle \rightarrow |sa^r \, mod \, N\rangle = |s\rangle$$

O sea que U_a es una raíz r-ésima de la identidad.
Como $amodN$ tiene orden r, U_a es una raíz r-ésima de la identidad.
Ejercicio
Pruebe que si un operador U cumple que $U^r = I$, entonces sus autovalores deben ser raíces r-ésima de 1, o sea de la forma $e^{2\pi i \frac{k}{r}}$ para algún entero k.

Consideremos el estado

$$|u_k\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r} s} \, |a^s modN\rangle \, .$$

Si le aplicamos U_a

$$\begin{aligned} U_a \, |u_k\rangle &= \tfrac{1}{\sqrt{r}} \textstyle\sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r} s} U_a \, |a^s modN\rangle \\ &= \tfrac{1}{\sqrt{r}} \textstyle\sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r} s} \, |a^{s+1} modN\rangle \\ &= e^{2\pi i \frac{k}{r}} \tfrac{1}{\sqrt{r}} \textstyle\sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r}(s+1)} \, |a^{s+1} modN\rangle \\ &= e^{2\pi i \frac{k}{r}} \, |u_k\rangle \end{aligned}$$

por lo tanto $|u_k\rangle$ es una autoestado de U_a con autovalor $e^{2\pi i\frac{k}{r}}$. La última igualdad se justifica porque $e^{2\pi i\frac{k}{r}r}\,|a^r mod N\rangle = e^{2\pi i\frac{k}{r}0}\,|a^0 mod N\rangle$.

Para cualquier valor de k entre 0 y $r-1$, podemos aplicar el algoritmo de estimación de autovalores para obtener k/r y así resolver el problema de búsqueda del orden.

Claro que no conocemos r, así que no sabemos cómo preparar los estados $|u_k\rangle$. Pero no se necesita porque una superposición de todos los autoestados con peso uniforme también sirve. El algoritmo de estimación de autovalores produce una superposición de estos autoestados *entangled* con las estimaciones de sus autovalores y una medición producirá como resultado una estimación de alguno de los autovalores. Veremos que es posible preparar una superposición de autoestados sin conocer r.

Tomemos

$$\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|u_k\rangle = \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}e^{-2\pi i\frac{k}{r}s}U_a\,|a^s mod\, N\rangle\,.$$

Si $s = 0(mod\, r)$ entonces $|a^s mod\, N\rangle = |1\rangle$. La amplitud de $|1\rangle$ en el estado de arriba es entonces la suma de las amplitudes sobre los términos con $s = 0$, o sea

$$\begin{aligned}\tfrac{1}{\sqrt{r}}\tfrac{1}{\sqrt{r}}\textstyle\sum_{k=0}^{r-1}e^{-2\pi i\frac{k}{r}0} &= \tfrac{1}{r}\textstyle\sum_{k=0}^{r-1}1\\ &= 1.\end{aligned}$$

Por lo tanto la amplitud del estado $|1\rangle$ es 1 y la amplitud del resto de los vectores de la base debe ser 0. Se tiene entonces

$$\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|u_k\rangle = |1\rangle\,.$$

Ésto significa que el algoritmo de estimación de autovalores transforma el estado de entrada

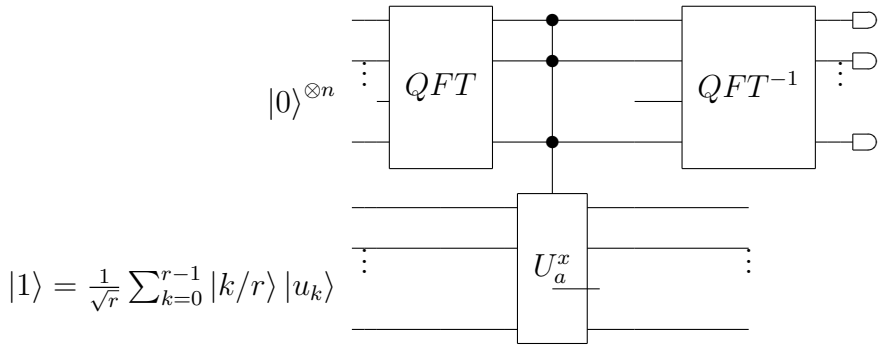
$$\begin{aligned}|0\rangle\,|1\rangle &= |0\rangle\left(\tfrac{1}{\sqrt{r}}\textstyle\sum_{k=0}^{r-1}|u_k\rangle\right)\\ &= \tfrac{1}{\sqrt{r}}\textstyle\sum_{k=0}^{r-1}|0\rangle\,|u_k\rangle\end{aligned}$$

al estado

$$\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|k/r\rangle\,|u_k\rangle\,.$$

Como el primer registro está en una mezcla uniforme de estados $|k/r\rangle$, una medición del mismo producirá un entero x tal que $\frac{x}{2^n}$ es una estimación de $\frac{k}{r}$ para algún $k \in \{0, 1, \ldots, r-1\}$. Como dijimos antes, esta estimación nos permite determinar con probabilidad alta $\frac{k}{r}$ de acuerdo al teorema de fracciones continuas.

El circuito cuántico correspondiente a la estimación de un múltiplo de $\frac{1}{r}$ para ser usado en la búsqueda de orden se muestra en la figura:



Los bits que se miden despues de QFT^{-1} son la representación binaria del entero x tal que $\frac{x}{2^n}$ es una estimación de $\frac{k}{r}$ para algún $k \in \{0, 1, \ldots, r-1\}$.

Resumen del Algoritmo de Búsqueda de Orden

1. Elegir un entero n tal que $2^n \geq 2r^2$. Típicamente se toma $n = 2logN$.
2. Inicializar un registro de n-qubits como $|0\rangle^{\otimes n}$. Éste es el *registro de control*.
3. Inicializar un registro de n-qubits como $|1\rangle = |00\ldots01\rangle$. Éste es el *registro target*.
4. Aplicar QFT al registro de control.
5. Aplicar $c - U_a^x$ a los registros target y control.
6. Aplicar QFT^{-1} al registro de control.
7. Medir el registro de control para obtener una estimación $\frac{x_1}{2^n}$ de un múltiplo entero de $\frac{1}{r}$.
8. Usar el algoritmo de las funciones continuas para obtener los enteros c_1 y r_1 tal que $|\frac{x_1}{2^n} - \frac{c_1}{r_1}| \leq \frac{1}{2^{\frac{n-1}{2}}}$. Si no se encuentran ese par de enteros, salida **'FALLO'**.
9. Repetir pasos 1-7 para encontrar otro entero x_2 y enteros c_2 y r_2 tal que $|\frac{x_2}{2^n} - \frac{c_2}{r_2}| \leq \frac{1}{2^{\frac{n-1}{2}}}$. Si no se encuentran ese par de enteros, salida **'FALLO'**.
10. Calcule $r = MCM(r_1, r_2)$. Calcule $a^r mod N$.
11. Si $a^r mod N = 1$, salida r. Si no, salida **'FALLO'**.

El algoritmo de búsqueda de orden obtiene el orden r de a correcto con probabilidad $\frac{384}{\pi^6} > 0.399$ y en los otros casos da **'FALLO'** como salida.

El cuello de botella computacional del algoritmo son las operaciones controladas del

operador U_a exponenciado, es decir las operaciones $c - U_a^{2^j}$ con $j = 0, 1, \ldots, 2^{n-1}$ necesarias para la estimación de autovalores. En principio calcular $c - U_a^{2^j}$ requiere 2^j aplicaciones de $c - U_a$. Sin embargo, se cumple que $c - U_a^{2^j} = c - U_{a^{2^j}}$ porque multiplicar 2^j veces por $amodN$ es equivalente que multiplicar sólo una por $a^{2^j} modN$. Podemos precalcular (clásicamente) $a^{2^j} modN$ con sólo j multiplicaciones módulo N (elevando al cuadrado y haciendo módulo N, empezando con a), lo que da una mejora exponencial con respecto a multiplicar por $amodN$ 2^j veces. El circuito cuántico es simplemente un circuito para multiplicar por el número $a^{2^j} modN$ que está entre 1 y $N - 1$. Las técnicas aritméticas standard requieren $O(log(N)loglog(N)logloglog(N))$ compuertas elementales para hacer esta multiplicación. Como la QFT requiere $O((logN)^2$ compuertas, este circuito cuántico requiere $O((log(N))^2 loglog(N)logloglog(N))$ compuertas cuánticas elementales. Nótese que solamente necesita un número constante de repeticiones para factorizar N en dos factores no-triviales con alta probabilidad de éxito.
Esta complejidad debe ser comparada con los mejores algoritmos heurísticos (tiempo de corrida usa suposiciones razonables pero no probadas) clásicos que usan $e^{O((logN)^{\frac{1}{3}}(loglogN)^{\frac{2}{3}})}$ y con los rigurosos que usan $e^{O((logN)^{\frac{1}{2}}(loglogN)^{\frac{1}{2}})}$.

Factorización de $N = 15$

Primero elegimos un número que no tenga factores comunes con 15, por ejemplo $x = 7$. Luego usamos el algoritmo cuántico para encontrar el orden r de x respecto de N. Se comienza con el estado $|0\rangle |0\rangle$ y aplicando QFT al primer registro se obtiene:

$$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle |0\rangle = \frac{1}{\sqrt{2^n}} \Big[|0\rangle + |1\rangle + \cdots + |2^n - 1\rangle \Big] |0\rangle$$

El valor de $n = 11$ es elegido para tener una probabilidad de error de 1/4 como máximo. Luego se calcula $x^k modN$ ubicando el resultado en el segundo registro. Ésto corresponde a la operación U_a^x controlada sobre el bit target del circuito cuántico mostrado arriba. Así se obtiene

$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle \left| x^k modN \right\rangle$

$= \frac{1}{\sqrt{2^n}} \Big[|0\rangle |1\rangle + |1\rangle |7\rangle + |2\rangle |4\rangle + |3\rangle |13\rangle + |4\rangle |1\rangle + |5\rangle |7\rangle + |6\rangle |4\rangle + \ldots \Big].$

Ahora hay que aplicar la QFT^{-1} al primer registro y calcular k/r para sacar r de ahí. Pero para ver qué se puede obtener nos podemos fijar en el segundo registro que tiene una periodicidad y repite 1, 7, 4 y 13. Supongamos que en la medida de este registro se obtiene 4, ésto significa que el estado del primer registro al que se aplica QFT^{-1} es $\sqrt{\frac{4}{2^n}}\Big[|2\rangle + |6\rangle + |10\rangle + |14\rangle + \ldots\Big]$. Despues de aplicar QFT^{-1} se obtiene un estado $\sum_l \alpha_l |l\rangle$. La distribución de probabilidad de este estado en el que $l = 2^n = 2048$ para $n = 11$ tiene cuatro picos de 1/4 cada uno en 0, 512, 1024 y 1536. Como todos ellos tienen la misma probabilidad, suponemos que se mide 1536. $1536/2048 = 3/4 = 1/(1 + 1/3)$, y 3/4 aparece como convergente en la expansión dando $r = 4$ como orden de $x = 7$ con $N = 15$. Como vimos antes, cuando el orden es par, como en este caso, $b = x^{r/2} mod N$ y $b^2 - 1 = 0 mod N$ y se cumple que N divide a $(b-1)(b+1)$. Se prueba con $MCD(b-1, N)$ o $MCD(b+1, N)$ para encontrar un factor no trivial de N, en este caso $b - 1 = 3$ y $b + 1 = 5$.

C. Algoritmo Original de Shor para Búsqueda de Orden

El que se presentó anteriormente no es el algoritmo original de Shor. El original era:

1. Crear el estado
$$|\psi_0\rangle = \sum_{x=0}^{2^n-1} \frac{1}{\sqrt{2^n}} |x\rangle \, |a^x mod N\rangle \, .$$
Se puede reescribir este estado como
$$\sum_{b=0}^{r-1} \Big(\frac{1}{\sqrt{2^n}} \sum_{z=0}^{m_b-1} |zr + b\rangle \, \big|a^b mod N\big\rangle \, .$$
donde m_b es el entero mas grande tal que $(m_b - 1)r + b \leq 2^n - 1$. La forma de escribir este estado refleja la periodicidad asociada al orden r buscado. Una expresión de este tipo se vió en la clase anterior en la búsqueda del período de un estado periódico.

2. Medir el segundo registro. Se obtendrá un valor de $a^b mod N$ con $b \in \{0, 1, \ldots, r-1\}$. El pimer registro quedará en
$$\frac{1}{\sqrt{mb}} \sum_{z=0}^{m_b-1} |zr + b\rangle \, .$$

Si se pudiese aplicar $QFT^{-1}_{m_b r}$ a este estado, se produciría la superposición

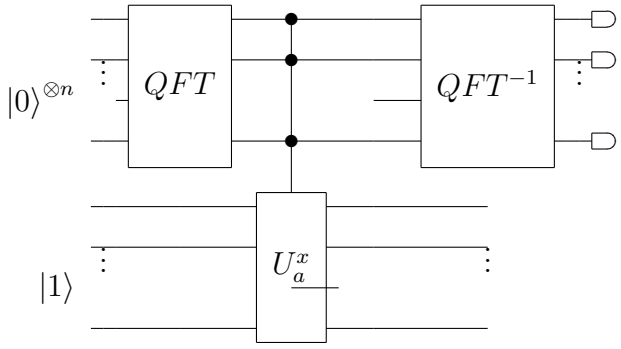
$$\sum_{j=0}^{r-1} e^{-2\pi i \frac{b}{r} j} \left| m_b j \right\rangle .$$

y se medirían valores x tales que $\frac{x}{r m_b} = \frac{j}{r}$ para algún entero j. Sin embargo como no conocemos r y m_b hay que usar $QFT^{-}_{2^n} 1$.

3. Aplicar $QFT^{-1}_{2^n}$ al primer registro y despues medir. Sea x el valor medido.
4. Salida $\frac{x}{2^n}$.

El resto del algoritmo sigue los pasos del que vimos antes.

El circuito cuántico correspondiente al algoritmo de Shor se muestra en la figura:



Nótese que es exactamente el mismo circuito que el descripto en la sección previa. La diferencia entre los dos algoritmos es la base en la que el estado del segundo registro es expresado: en la versión previa en la base de autovectores y en la de Shor en la computacional. En la tabla siguiente se ven las diferencias.

	Shor	Estimación Autovalores
Estado Inicial	$\left\|0\right\rangle \left\|1\right\rangle$	$\sum_k \left\|0\right\rangle \left\|u_k\right\rangle$
QFT	$\sum_x \left\|x\right\rangle \left\|1\right\rangle$	$\sum_k \sum_x \left\|x\right\rangle \left\|u_k\right\rangle$
$c - U_a^x$	$\sum_b (\sum_z \left\|zr + b\right\rangle) \left\|a^b\right\rangle$	$\sum_k \left(\sum_x e^{2\pi i \frac{kx}{r} \left\|x\right\rangle} \right) \left\|u_k\right\rangle$
QFT^{-1}	$\frac{x}{2^n}$	$\frac{x}{2^n}$

D. Subgrupos Ocultos

Los problemas vistos hasta ahora pueden ser reformulados como un problema llamado del *subgrupo oculto*.

Problema del Subgrupo Oculto

Sea $f : G \to X$ que mapea un grupo G a algún conjunto finito X con la propiedad de que existe algún subgrupo $S \leq G$ tal que para todo $x, y \in G, f(x) = f(y)$ si y sólo si $x + s = y + s$. Se puede decir también que f es constante en cosets o coconjuntos de S y distinta en otros coconjuntos. Veamos qué es un coconjunto o coset.

Si S es un subgrupo de G, se llama coset o coconjunto de S a las copias desplazadas de S que cubren todo G sin solaparse. Por ejemplo, sea G el plano xy y sea S el eje x. Por cada número real c, la recta que corre paralela al eje x, pasando por $y = c$, es un coset del eje x. Estos cosets cubren el plano entero sin solaparse.

Problema de Deutsch (versión original):

Se tiene una "caja negra" para calcular una función desconocida $f : \{0, 1\} \to \{0, 1\}$.

Problema: Determinar $f(0) \oplus f(1)$ haciendo consultas por f.

Problema de Deutsch (versión subgrupo oculto):

$G = \mathbb{Z}_2,\ X = \{0, 1\}$ y $S = 0$ si f es balanceado y $S = \mathbb{Z}_2$ si f es constante.

Búsqueda de Orden (versión original):

Dados los enteros a y N, que cumplen que $MCD(a, N) = 1$, el *orden* de $a \, mod \, N$ es el mínimo entero positivo r tal que $a^r \equiv 1 \, mod \, N$.

Búsqueda de Orden (versión subgrupo oculto):

$G = \mathbb{Z},\ X =$ un grupo finito H, r es el orden de $a \in H$. El subgrupo $S = r\mathbb{Z}$ es el subgrupo oculto de G y un generador para S revela r.

Búsqueda de Período (versión subgrupo oculto):

$G = \mathbb{Z},\ X =$ cualquier conjunto. r es el período de f. El subgrupo $S = r\mathbb{Z}$ es el subgrupo oculto de G y un generador para S revela r.