Algoritmo cuántico de P. Shor

Teoría de la información cuántica

D. Orellana B. Sandoval

Ingeniería civil matemática
Universidad de concepción

Julio 2022

Tabla de contenidos

Tabla de contenidos

Teorema fundamental de la aritmética

Teorema

Sea $N \in \mathbb{Z}$ mayor que 1, entonces admite factorización prima, esto es

$$N = \prod_{i=1}^{k} \xi_i^{N_i}$$

donde $\xi_1, ..., \xi_k$ son números primos y $N_1, ..., N_k$ son enteros positivos.

¿Existe algún método que permita obtener los $\xi_1, \ldots, \xi_k$?

La respuesta a esta pregunta es si, por ejemplo, se tiene la criba de Eratóstenes, este método tiene una complejidad computacional $\mathcal{O}(N \log\log N)$ lo cual para N suficientemente grande el algoritmo se hace imposible de utilizar. En general, no hay un algoritmo clásico que pueda resolver este problema en un tiempo razonable.

Algoritmo Shor

Si bien en el mundo clásico no hay una forma eficiente de resolver el problema, para nuestra suerte (o quizás mala suerte) en el mundo cuántico si hay una solución a este problema, en 1994 el matemático Peter Shor presenta un algoritmo cuántico el cual es capaz de factorizar un número entero N de n bits en tiempo polinomial $\mathcal{O}(n^3)$.

Congruencias

Definicion

Sea $N \in \mathbb{Z} - \{0\}$ diéremos que dos números enteros a y b son congruentes modulo m si el número $(a - b)$ es divisible por m. Lo anterior se puede escribir

$$a \equiv b \pmod{N}$$

Observación: se puede demostrar que la congruencia define una relación de equivalencia.

Tabla de contenidos

Definición

Sean x y N enteros positivos y coprimos tales que $x < N$. Se define el **orden** de x modulo N como el menor entero positivo r tal que

$$x^r \equiv 1 \pmod{N}$$

Resolver el problema de encontrar el orden de x modulo N (con x y N dados) es bastante difícil utilizando un computador clásico, de hecho, no se conoce un algoritmo que sea capaz de resolver el problema utilizando una cantidad de bits del orden polinomial $\mathcal{O}(L)$ con $L = \lceil \log N \rceil$ que es el número de bits necesarios.

Algoritmo Order-finding

Si bien la computación clásica no resuelve este problema de forma eficiente, en el mundo cuántico si hay una solución, la cual se basa en utilizar el algoritmo de estimación de fase, pero con una ligera modificación, esta es, usar el siguiente operador unitario

$$U|y\rangle = |xy \bmod N\rangle$$

donde $y \in \{0,1\}^L$.
Nota: en lo que sigue, cuando $N \leq y \leq 2^L - 1$, utilizamos la convención que $U|y\rangle = |y\rangle$ y cuando $0 \leq y \leq N-1$ se tiene que U actúa de forma no trivial sobre $|y\rangle$.

Se puede demostrar que el estado definido por

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2i\pi k \frac{s}{r}} |x^k \bmod N\rangle, \quad s \in \mathbb{Z} \cap [0, r-1]$$

es un estado propio de U y además

$$U|u_s\rangle = e^{2\pi i \frac{s}{r}} |u_s\rangle$$

Observación: el estimador de fase nos permite obtener, con gran precisión, los correspondientes valores propios $e^{2\pi i \frac{s}{r}}$, a partir de los cuales podemos obtener el orden r.

Requisitos

Para que al aplicar el algoritmo de estimación de fase solucione el problema de forma eficiente, debemos tener en cuenta lo que sigue

1. Al momento de aplicar las operaciones control-U^{2^j} en el circuito estas se deben obtener de forma eficiente para todo $j \in \mathbb{Z}_0^+$.
2. Debemos ser capaces de preparar eficientemente un estado propio $|u_s\rangle$ con un valor propio no trivial o al menos una superposición de tales estados propios.

Solucion a los requisitos

1. El primer requisito se pude cumplir usando un procedimiento conocido como **Modular exponentiation** con el cual se obtiene

$$\begin{aligned} |z\rangle |y\rangle &\rightarrow |z\rangle U^{z_t 2^{t-1}} \ldots U^{z_1 2^0} |y\rangle \\ &= |z\rangle |x^{z_t 2^{t-1}} \times \cdots \times x^{z_1 2^0} y (\text{mod } N)\rangle \\ &= |z\rangle |x^z y (\text{mod } N)\rangle \end{aligned}$$

   con este método utilizamos una cantidad de puertas del orden polinomial $\mathcal{O}(L^3)$.
2. El segundo requisito es mas complejo, pues para tener $|u_s\rangle$ debemos saber el valor de r. Afortunadamente hay una propiedad que nos permite solucionar esto, la cual veremos en la siguiente proposición.

Proposición

Sean $\{|u_s\rangle\}_{s\in\mathbb{Z}\cap[0,r-1]}$ los estados propios del operador U definidos anteriormente, entonces

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle = |1\rangle$$

Observación: con argumentos similares utilizados en la demostración se puede mostrar que

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}e^{2\pi ik\frac{s}{r}}\,|u_s\rangle = |x^k \bmod N\rangle$$

Demostración: sea $\{|u_s\rangle\}_{s\in\mathbb{Z}\cap[0,r-1]}$ una familia de estados propios de U y además sea r el orden de x modulo N, entonces

$$\begin{aligned}
\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle &= \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\sum_{k=0}^{r-1}\frac{1}{\sqrt{r}}e^{-2\pi ik\frac{s}{r}}\,|x^k \bmod N\rangle \\
&= \frac{1}{r}\sum_{k=0}^{r-1}\underbrace{\left[\sum_{s=0}^{r-1}\left(e^{\frac{-2\pi ik}{r}}\right)^s\right]}_{r\delta_{k,0}}|x^k \bmod N\rangle \\
&= \frac{1}{r}\sum_{k=0}^{r-1} r\delta_{k,0}\,|x^k \bmod N\rangle \\
&= \sum_{k=0}^{r-1}\delta_{k,0}\,|x^k \bmod N\rangle = |x^0 \bmod N\rangle = |1\rangle
\end{aligned}$$

Al realizar el procedimiento de estimación de fase para $t = 2L + 1 + \lceil \log\left(2 + \frac{1}{2\varepsilon}\right) \rceil$ qubits, preparamos el segundo registro en el estado $|1\rangle$ (que es sencillo de calcular) y así se tiene que $\forall s \in \mathbb{Z} \cap [0, r-1]$ obtenemos una estimación de fase $\varphi \approx \frac{s}{r}$ exacta a $2L + 1$ bits con probabilidad de al menos $\frac{1-\varepsilon}{r}$.

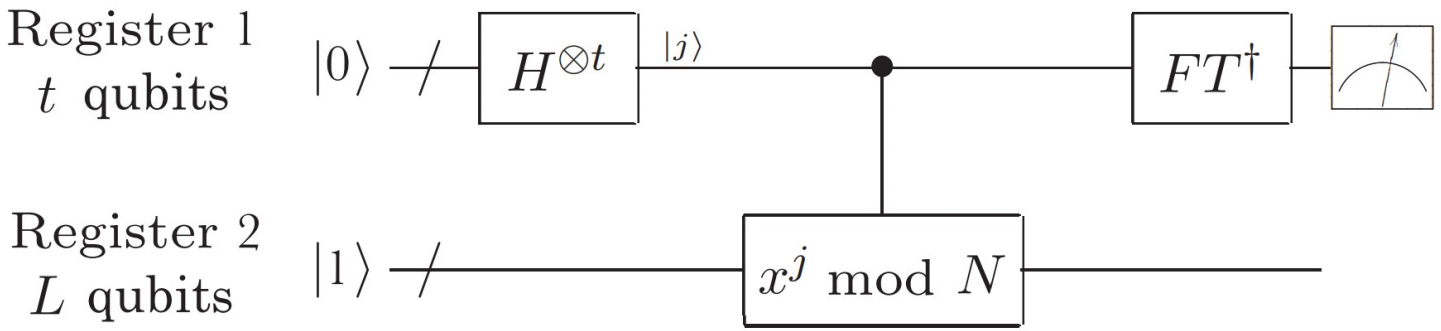


Figure: Circuito cuántico para el algoritmo order-finding.

La reducción de order-finding a la estimación de fase se completa luego de obtener el valor de r, a partir del algoritmo de estimación de fase encontramos $\varphi \approx \frac{s}{r}$ donde solo conocemos $2L + 1$ bits de φ, pero también sabemos a priori que es un numero racional, de esta forma si pudiéramos encontrar la fracción mas cercana a φ podríamos obtener a r.

Expansión de la fracción continua

La idea de este algoritmo es describir un numero real en términos de números enteros utilizando expresiones de la forma

$$[a_0, ..., a_M] = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{... + \cfrac{1}{a_M}}}}$$

donde $a_j \in \mathbb{Z}^+$ para todo $j \in \{1, ..., M\}$. Definimos la $m-$ésima convergencia $(0 \leq m \leq M)$ a esta fracción continua como $[a_0, ..., a_m]$.

Ejemplo

Supongamos que intentamos descomponer $\frac{31}{13}$ como una fracción continua, lo primero que haremos será encontrar el cociente q y el resto r de la división entera entre 31 y 13

$$31 = 13q + r$$

de lo anterior se deduce $q = 2$ y $r = 5$, entonces

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

Procedemos de forma análoga para descomponer $\frac{13}{5}$, así

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

Es claro que el algoritmo de las fracciones continuas termina después de un número finito de pasos de "dividir e invertir" para cualquier número racional, ya que los numeradores que aparecen ($31, 5, 3, 2, 1$ en el ejemplo) son estrictamente decrecientes. ¿con qué rapidez se produce esta terminación? Resulta que si $\varphi = \frac{s}{r}$ es un número racional y tanto s como r son números enteros de L bits, entonces la expansión de la fracción continua para φ se puede calcular utilizando $\mathcal{O}(L^3)$ operaciones $\mathcal{O}(L)$ pasos de "dividir e invertir", cada uno utilizando $\mathcal{O}(L^2)$ puertas para aritmética elemental.

Teorema

Sea s/r un número racional tal que

$$\left|\frac{s}{r} - \varphi\right| \leq \frac{1}{2r^2}$$

entonces s/r es un convergente de la fracción continua para φ y por lo tanto puede ser calculado en $\mathcal{O}(L^3)$ operaciones utilizando el algoritmo de fracciones continuas.

Algoritmo cuántico de order-finding

Entrada:

1. Una caja negra $U_{x,N}$ que realiza la transformación $|j\rangle |k\rangle \rightarrow |j\rangle |x^j k \bmod N\rangle$, para x coprimo al numero N de $L-$bits.
2. $t = 2L + 1 + \lceil \log\left(2 + \frac{1}{2\varepsilon}\right) \rceil$ qubits inicializados en el estado $|0\rangle$.
3. $L = \lceil \log(N) \rceil$ qubits inicializados en el estado $|1\rangle$.

Salida: El menor entero positivo r tal que $x^r \equiv 1 \pmod N$.
Tiempo ejecución: $\mathcal{O}(L^3)$ operaciones. Probabilidad de éxito $\mathcal{O}(1)$.

Algoritmo cuántico de order-finding

Procedimiento:

1. $|0\rangle |1\rangle$ Estado inicial.
2. $\rightarrow \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle |1\rangle$ superposición de estados.
3. $\rightarrow \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle |x^j \bmod N\rangle \approx \frac{1}{\sqrt{r2^t}} \sum_{s=0}^{r-1} \sum_{j=0}^{2^t-1} e^{2\pi isj/r} |j\rangle |u_s\rangle$
4. $\rightarrow \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \widetilde{|s/r\rangle} |u_s\rangle$ Aplicamos IFT al primer registro
5. $\rightarrow \widetilde{s/r}$ Medición del primer registro.
6. $\rightarrow r$ Aplicamos el algoritmo de fracciones continuas.

Tabla de contenidos

Factorizacion

En lo que sigue veremos que el problema de order finding es equivalente al problema de encontrar los factores primos de un entero mayor que 1, en el sentido de que un algoritmo rápido para la búsqueda de órdenes puede convertirse fácilmente en un algoritmo rápido para la factorización.

Reducción de factorizacion a order-finding

1. mostrar que podemos calcular un factor de N, si podemos encontrar una solución no trivial $x \not\equiv \pm 1 \pmod{N}$ a la ecuación $x^2 \equiv 1 \pmod{N}$.
2. mostrar que un y arbitrario coprimo a N tiene un alta probabilidad de que tenga orden par, tal que, $y^{\frac{r}{2}} \not\equiv \pm 1 \pmod{N}$ y, por tanto, $x \equiv y^{\frac{r}{2}} \pmod{N}$. Una solución no trivial de $x^2 \equiv 1 \pmod{N}$

Resultados importantes

Teorema

Supongamos que N es un número compuesto de L bits y x es una solución no trivial de la ecuación $x^2 \equiv 1 \pmod{N}$ con $x \in [1, N]$, es decir, ni $x \equiv 1 \pmod{N}$ ni $x = N - 1 \equiv -1 \pmod{N}$. Entonces al menos uno de $\mathrm{mcd}(x - 1, N)$ y $\mathrm{mcd}(x + 1, N)$ es un factor no trivial de N que puede ser calculado mediante las operaciones de $\mathcal{O}(L^3)$.

Teorema

Suponga $N = \prod_{i=1}^{m} \xi_i^{N_i}$ es la factorizacion prima de un entero compuesto, impar y positivo. Sea $x \in \mathbb{Z} \cap [1, N-1]$ elegido al azar y ademas con la condicion de que sea coprimo con N. Sea r el orden de x modulo N. Entonces

$$P\left(r \text{ es par y } x^{r/2} \not\equiv -1 \pmod{N}\right) \geq 1 - \frac{1}{2^m}$$

Ambos teoremas pueden combinarse para dar un algoritmo que, con alta probabilidad devuelve un factor no trivial de cualquier compuesto N. Todos los pasos del algoritmo se pueden realizar eficientemente en un ordenador clásico, excepto (por lo que se sabe hoy en día) la subrutina order-finding que utiliza el algoritmo. Repitiendo el procedimiento podemos encontrar una factorización primaria completa de N. El algoritmo se resume a continuación.

Algoritmo Shor

Entrada: Un número compuesto N.
Salida: Un factor no trivial de N.
Tiempo de ejecución: $\mathcal{O}((\log(L))^3)$ operaciones. Probabilidad de éxito $\mathcal{O}(1)$.

Algoritmo Shor

Procedimiento:

1. Si N es par, devuelve el factor 2.
2. Determinar si $N = a^b$ para los enteros $a \geq 1$ y $b \geq 2$ y si es así devuelve el factor a.
3. Elegir al azar un $x \in \mathbb{Z} \cap [1, N-1]$. Si $\text{mcd}(x, N) > 1$ devolver el factor $\text{mcd}(x, N)$.
4. Utilice la subrutina de order-finding para encontrar el orden r de x módulo N.
5. si r es par y $x^{\frac{r}{2}} \not\equiv -1 \ (\text{mod} N)$ entonces calcula, $\text{mcd}(x^{\frac{r}{2}} - 1, N)$ y $\text{mcd}(x^{\frac{r}{2}} + 1, N)$ compruebe si uno de ellos es un factor no trivial, devolviendo ese factor si es así. En caso contrario, el algoritmo falla.

Tabla de contenidos

Algoritmo Shor

Entrada: $N = 15$.
Tiempo de ejecución: $\mathcal{O}((\log(\lceil \log 15 \rceil))^3)$ operaciones.
Probabilidad de éxito $\mathcal{O}(1)$.

Algoritmo shor

Procedimiento:

1. N es impar, entonces seguimos con el paso 2.
2. No existen enteros $a \geq 1$ y $b \geq 2$ tales que $N = a^b$, entonces seguimos con el paso 3.
3. Elegimos al azar un $x \in \mathbb{Z} \cap [1, 14]$, para este ejemplo, sea $x = 7$, notemos que $\text{mcd}(7, 15) = 1$, entonces seguimos con el paso 4.

Paso 4: Encontrar el orden de 7 modulo 15

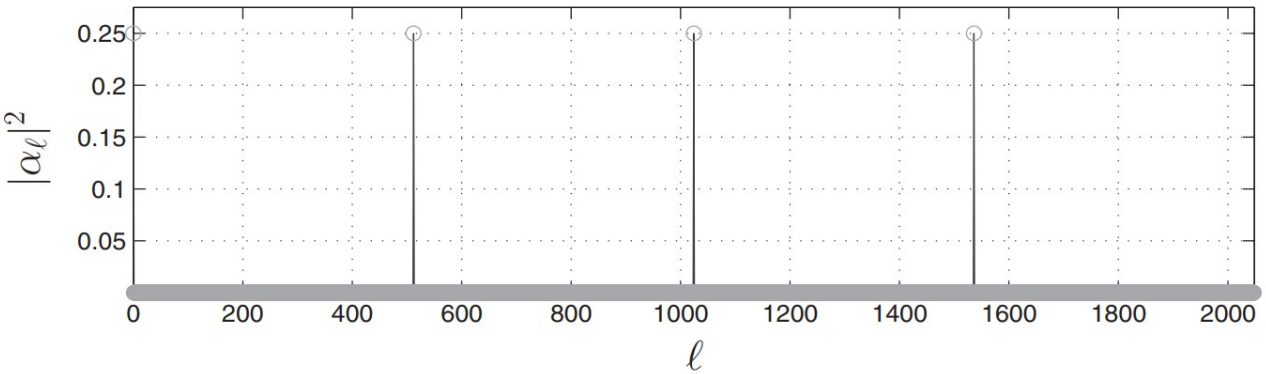
Comenzamos en el estado $|0\rangle |1\rangle$ y creamos el estado

$$|\Psi_1\rangle = \frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} |k\rangle |1\rangle = \frac{1}{\sqrt{2^t}} \left[|0\rangle + |1\rangle + \cdots + |2^t - 1\rangle\right] |1\rangle$$

Aplicando $t = 11$ transformaciones de Hadamard al primer registro. Esa elección de t garantiza una probabilidad de error como máximo de $\varepsilon = \frac{1}{4}$. Después de aplicar las puertas condicionales, se tiene el estado

$$|\Psi_2\rangle = \frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} |k\rangle |7^k \bmod 15\rangle$$

Aplicamos la transformada de Fourier inversa al primer registro y lo medimos. Aplicando el principio de medición implícita y midiendo el segundo registro, obtenemos un resultado aleatorio de 1, 7, 4 o 13. Supongamos que obtenemos 4, esto significa que la entrada del estado a U_{FT}^{*} habría sido $\frac{4}{2^t}\left[|2\rangle + |6\rangle + |10\rangle + |14\rangle + \cdots\right]$. después de aplicar U_{FT}^{*} se obtiene el estado $\sum_\ell \alpha_\ell |\ell\rangle$ con una distribución de probabilidad



mostrado para $2^t = 2048$. Por lo tanto, la medición final puede dar 0, 512, 1024 o 1536. Cada uno con una probabilidad casi exacta de $1/4$. Supongamos que obtenemos $\ell = 1536$ a partir de la medición, el cálculo de la expansión de la fracción continua da entonces $\frac{1536}{2048} = \frac{1}{1+\frac{1}{3}}$, por lo que $3/4$ aparece como convergente en la expansión, dando $r = 4$ como orden de $x = 7$. Por casualidad, r es par, y además $x^{r/2} \bmod N \equiv 72 \bmod 15 \equiv 4 \not\equiv -1 \bmod 15$, por lo que el algoritmo funciona: calcular el máximo común divisor $\mathrm{mcd}(x^2 - 1, 15) = 3$ y $\mathrm{mcd}(x^2 + 1, 15) = 5$ nos dice que $15 = 3 \cdot 5$

Tabla de contenidos

Referencias


Nielsen, M. A., & Chuang, I. (2002)
Quantum Computation and Quantum Information.