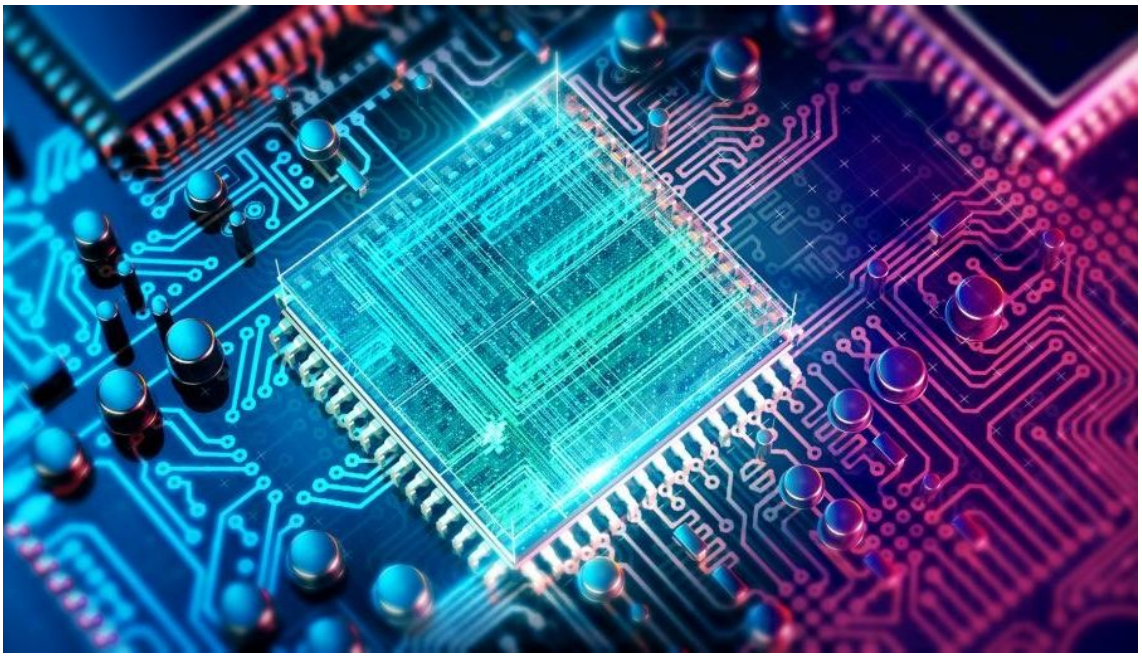
Prime Factorization Algorithms: A quantum approach.

Ramon Ruiz Dolz



Índice

1. Introducción **3**
1.1. Complejidad computacional 3

2. Conceptos básicos **4**
2.1. Bits cuánticos 4
2.2. Computación cuántica 4
2.3. Transformada cuántica de Fourier 5

3. Algoritmo de Shor **7**
3.1. Parte clásica 7
3.2. Parte cuántica 7

4. Conclusiones **9**

Referencias **10**

1. Introducción

1.1. Complejidad computacional

En la actualidad, los problemas se clasifican en función del número de pasos computacionales que cuesta encontrar una solución a una instancia grande del problema. En la Figura 1 podemos apreciar las clases existentes para los problemas según la teoría de la complejidad.

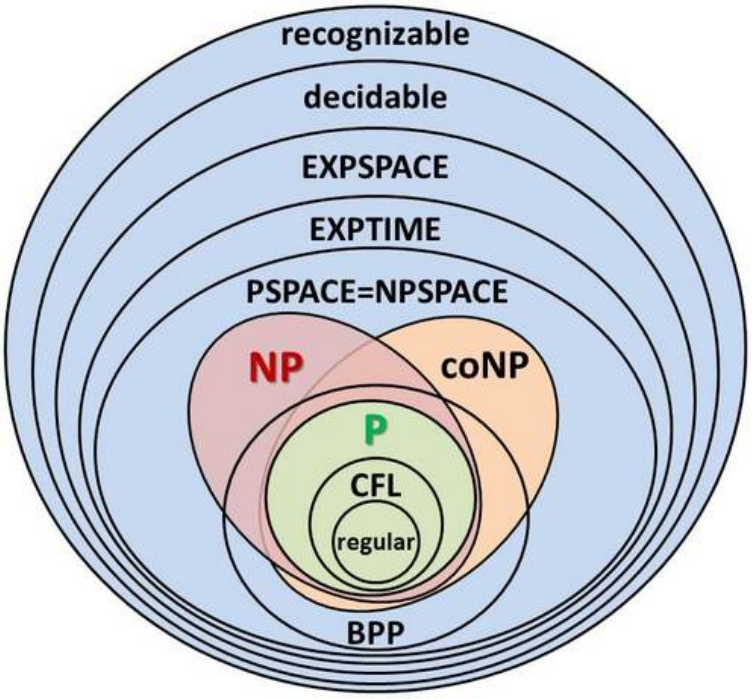


Figura 1: Clases de complejidad

Se entiende como un algoritmo eficiente todo algoritmo que, sea n la talla del problema, utiliza un numero de pasos de computación que crece como n elevado a una potencia fija.

Para este trabajo, son de especial interés las clases de problemas P y NP. La clase P de *polynomial* incluye a aquellos problemas cuya solución se puede obtener de manera eficiente por parte de un computador clásico, es decir en tiempo polinómico. La clase NP *nondeterministic polynomial* contiene aquellos problemas cuya solución, una vez obtenida es fácil de verificar, sin embargo tiene coste polinómico en maquinas no deterministas. Esto implica que en los ordenadores clásicos, hoy en día, estos problemas son irresolubles de forma eficiente.

La computación cuántica, un nuevo paradigma de la computación, plantea la posibilidad de reducir radicalmente los tiempos de resolución de los problemas NP. Esto es de gran interés para el problema de factorización de números, que tiene ya soluciones obtenibles de manera eficiente según el paradigma de la computación cuántica, concretamente mediante el algoritmo de Shor.

2. Conceptos básicos

2.1. Bits cuánticos

Un bit cuántico o qubit tiene múltiples estados, como un bit clásico. El qubit puede tomar los estados 0 o 1 como un bit clásico, estos estados se representan mediante la notación cuántica $|0\rangle$ o $|1\rangle$. La principal diferencia entre los dos tipos de bit es que los qubits pueden tomar otros valores distintos de estos. También es posible formar combinaciones lineales de estados llamadas superposiciones:

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

Los valores α y β son números complejos, aunque a veces son tratados como números reales. Esto se puede interpretar como que el estado del qubit es un vector en un espacio vectorial bidimensional complejo.

En la computación clásica es totalmente común consultar el estado de un bit, por ejemplo cuando realizamos consultas a memoria. En cambio consultar el estado de un qubit, es decir obtener los valores de α y β es imposible. Cuando se mide el estado de un qubit, se obtiene 0 con probabilidad de $|\alpha|^2$ y 1 con la probabilidad de $|\beta|^2$. Por lo tanto $|\alpha|^2+|\beta|^2 = 1$ ya que las probabilidades deben sumar uno en total. Geométricamente esto se puede ver como la normalización del estado del qubit, por ello el estado es realmente un vector unitario en un espacio vectorial bidimensional complejo.

Estos qubits son completamente reales, existen en la naturaleza como por ejemplo los electrones. Un electrón puede estar en en su estado fundamental $|0\rangle$ o en estado excitado $|1\rangle$. Encendiendo luz en un átomo con la energia adecuada es posible variar el estado de los electrones de $|0\rangle$ a $|1\rangle$ y viceversa. Además, moderando el tiempo de iluminación del átomo con la luz el electrón puede quedar en un estado intermedio $|+\rangle$.

2.2. Computación cuántica

Las leyes de la mecánica cuántica únicamente permiten transformaciones unitarias en los vectores de estado, por ello, para poder tener una máquina física que realice este tipo de computación será necesario tener la capacidad de realizar cambios en el estado del sistema. Una matriz unitaria es aquella cuya matriz conjugada transpuesta es igual a su inversa, que se requieran transformaciones de estado representadas por matrices unitarias asegura que la suma de posibilidades de obtener cualquier salida es igual a uno.

La definición de circuitos cuánticos y maquinas de Turing cuánticas únicamente permiten transformaciones unitarias en forma de transformaciones sobre un número fijado de bits. Esto esta justificado, ya que no esta claro cómo implementar físicamente una transformación unitaria general en n bits, sin embargo transformaciones de dos bits se pueden implementar con relativa facilidad. Es por esto que los sets de transformaciones de dos bits componen el bloque básico de la computación cuántica, de la misma forma que las puertas lógicas clásicas lo son para la computación clásica. De hecho en un set universal de puertas cuánticas basta con tomar las puertas de un único bit y tan solo una de dos bits, el NOT controlado que niega el segundo bit si y solo si el primer bit es un uno.

Una puerta cuántica es únicamente factible si su matriz correspondiente es unitaria por las propiedades explicadas anteriormente, y por lo tanto su conjugada transpuesta es igual a su inversa.

Una puerta cuántica puede ser por ejemplo:

$$|00\rangle \Rightarrow |00\rangle \tag{2}$$

$$|01\rangle \Rightarrow |01\rangle \tag{3}$$

$$|10\rangle \Rightarrow \frac{1}{\sqrt{2}}(|10\rangle + |11\rangle) \tag{4}$$

$$|11\rangle \Rightarrow \frac{1}{\sqrt{2}}(|10\rangle - |11\rangle) \tag{5}$$

Partiendo de esta puerta cuántica y de la siguiente superposición de estados:

$$\frac{1}{\sqrt{2}}|10\rangle - \frac{1}{\sqrt{2}}|11\rangle \tag{6}$$

podemos aplicar las transformaciones unitarias que definen la puerta lógica cuántica (2),(3),(4) o (5). Tras aplicar las transformaciones, la máquina llegará a la superposición de estados:

$$\frac{1}{2}(|10\rangle + |11\rangle) - \frac{1}{2}(|10\rangle - |11\rangle) = |11\rangle \tag{7}$$

Con este ejemplo podemos observar de forma simple el funcionamiento del proceso de computación cuántica. Es interesante resaltar que, aunque el estado $|10\rangle$ ha existido durante el proceso, en la instancia final la probabilidad de amplitud se ha cancelado y por lo tanto, en caso de medir el estado del qubit una vez terminadas las transformaciones $|10\rangle$ no seria observable.

Un *quantum gate array* es un conjunto de puertas cuánticas con çables"lógicos que conectan sus entradas con sus salidas. La entrada de esta estructura, que puede ser alargada añadiendo bits inicializados a 0, pasa a través de una secuencia de puertas cuánticas. Los valores de los bits se observan trás la última puerta, estos valores conforman la salida. Para que estos *array* sean uniformes es necesario añadir dos características más. La primera es el requisito estándar de que el diseño del *array* sea producido en tiempo polinómico en un computador clásico. La segunda es que las entradas en las matrices unitarias que describen las puertas deben ser números computables. Específicamente los primeros log n bits de cada entrada deberían ser computables en tiempo polinómico de n. Esto evita que la información no computable o difícil de computar se esconda en los bits de amplitud de las puertas cuánticas.

2.3. Transformada cuántica de Fourier

Una transfomada de Fourier es una transformación matemática empleada en el cambio de dominio de señales. Esta transformación es útil para cambiar entre el dominio del tiempo o espacio y el de la frecuencia.

Puesto que la computación cuántica trabaja con transformaciones unitarias, es de ayuda crear algunas de estas transformaciones que nos sean de utilidad. La Transfomación cuántica de Fourier no es más que una transformación unitaria realizada en tiempo polinómico en un computador cuántico. Esta transformación se dará como una matriz con tanto filas como columnas indexadas por estados. Estos estados se corresponden con representaciones binarias de enteros en el computador, concretamente se indexarán empezando por 0 a menos que se especifique otra cosa.

Esta transformación parte de un número a, tal que $0 \leq a < q$ para una q cuyo número de bits sea polinómico. La transformación lleva del estado $|a\rangle$ al estado:

$$\frac{1}{q^{\frac{1}{2}}} \sum_{c=0}^{q-1} |c\rangle \, exp(2\pi iac/q) \tag{8}$$

Se aplica la transformación a la matriz unitaria, en (a, c) cuya entrada es $\frac{1}{q^{\frac{1}{2}}} exp(2\pi iac/q)$. Esta transformada de Fourier es la base de muchos algoritmos cuánticos, y a la matriz se le llama A_q.

3. Algoritmo de Shor

El algoritmo de Shor es un algoritmo cuántico útil para factorizar un número N en tiempo $O((logN)^3)$ y espacio O(log$N$). Como se ha expuesto en las secciones anteriores, este algoritmo obtiene una solución en forma de probabilidades, da la solución correcta con elevada probabilidad y la de fallo se puede reducir realizando más iteraciones del algoritmo.

Este algoritmo esta dividido en dos partes, una primera parte clásica que sirve para reducir el problema de descomponer en factores al problema de encontrar el orden y una segunda parte cuántica que consiste en hallar el periodo. Partiendo de un número entero N, deseamos encontrar otro número p entre 1 y N que divida a N.

3.1. Parte clásica

1. Escoge un número pseudo-aleatorio $a < N$.
2. Obtiene el mcd(a,N) mediante el algoritmo de Euclides.
3. Si mcd(a,N) $\neq 1$ se ha encontrado un factor y termina. Si no, realiza una llamada a la parte cuántica para encontrar el periodo r de la función:

$$f(x) = a^x mod N \tag{9}$$

4. Si r es impar, vuelve al paso 1
5. Si $a^{\frac{r}{2}} = -1$ vuelve al paso 1
6. Obtiene los factores de N como el $mcd(a^{\frac{r}{2}} \pm 1, N)$ y finaliza.

Esta primera parte puede correr tanto en una máquina clásica como en una cuántica. Puesto que lo que hace es transformar un problema en otro equivalente, esta puede lanzarse en un computador clásico y este realizar llamadas a la parte cuántica corriendo en otra máquina.

3.2. Parte cuántica

1. Inicializa un par de registros qubits de entrada y salida con $log_2 N$ a:

$$N^{\frac{-1}{2}} \sum_{x=0}^{N-1} |x\rangle |0\rangle \tag{10}$$

2. Construye *f(x)* como función cuántica y la aplica al estado anterior obteniendo:

$$N^{\frac{-1}{2}} \sum_{x=0}^{N-1} |x\rangle |f(x)\rangle \tag{11}$$

3. Aplica la transformada cuántica de Fourier al registro de entrada, alcanzando el siguiente estado:

$$N^{-1} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} e^{\frac{2\pi i x y}{N}} |y\rangle |f(x)\rangle \tag{12}$$

4. Realiza una medición, obtiene un valor y en el registro de entrada y $f(x_0)$ en el de salida.

5. Convierte y/N en una fracción irreductible y extrae el denominador r', candidato a r

6. Comprueba si $f(x) = f(x + r')$, en caso afirmativo termina.

7. Si no, obtiene más candidatos a r usando valores cercanos a y o múltiples de r'. Si cumple la condición anterior termina.

8. Si no vuelve al paso 1 de la parte cuántica.

Esta segunda parte del algoritmo esta diseñada para ser lanzada en una máquina cuántica, a diferencia de la primera parte que puede correr en una clásica. Para realmente acelerar el proceso es necesario disponer de un computador cuántico puesto que la superposición de estados es clave de la reducción de costes.

4. Conclusiones

El algoritmo de Shor es realmente interesante puesto que lo que realmente hace es convertir el problema de factorización de números en un problema de cálculo de frecuencia de onda. Y es aquí dónde cobra interés el computador cuántico ya que al poder estar en múltiples estados simultáneamente gracias a la superposición cuántica, es capaz de hallar esta frecuencia de manera eficiente y por lo tanto de dar solución al problema de factorización de números.

Pese a esto, en la actualidad no se disponen de computadores cuánticos con muchos qubits, por lo tanto hasta el momento no se ha conseguido factorizar nada que no pudiesemos hacer en tiempo útil en un computador clásico. Concretamente el actual servicio en la nube de IBM cuenta con 5 qubits, sin embargo ya hay anunciados dos modelos uno de 20 para finales de año y otro de 50 para un futuro no muy lejano.

Referencias

[1] Michael A. Nielsen and Isaac L. Chuang Quantum Computation and Quantum Information. *Cambridge University Press*, 2000.

[2] Peter W. Shor Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *Computer Society Press*, 1996.

[3] Scott Aaronson The limits of Quantum. *SCIENTIFIC AMERICAN*, 2008.