# **Algoritmo de Grover y problema 3-SAT**

Camilo Andrés Quintero Rodríguez

Escuela Colombiana de Ingeniería Julio Garavito

camilo.quintero-r@mail.escuelaing.edu.co

11/29/2024

*Este reporte se entrega para cumplir con los requisitos parciales del curso CNYT: Computación Cuántica- 2024-2*

# **Tabla de contenidos**

Tabla de contenidos. 1

1 Introducción. 2

2 Problema 3-SAT. 2

2.1 Problema. 3

2.2 Determinando los valores de verdad. 4

3 Algoritmo de Grover. 5

3.1 Problema. 5

3.2 Implementando el algoritmo de Grover en un computador cuántico. 6

3.3 Resultados tras la implementación del algoritmo de Grover. 9

4 Conclusiones. 12

5 Bibliografía. 12

# **1** **Introducción**

En este informe se presentará la implementación del algoritmo de Grover con el fin de dar solución al problema 3-SAT, este problema es un ejercicio computacional NP-completo. Así mismo, para dar explicación a este problema, determinaremos los valores de verdad que satisfacen un conjunto de cláusulas, el motivo de esto se explicará durante el desarrollo de este informe.

En el presente informe, se expondrá de forma simplificada la teoría del problema 3-SAT, es decir, una explicación de dicho problema, para esto, primero presentaremos la teoría del problema 3-SAT, posteriormente, se presentarán los ejercicios realizados en el desarrollo del taller, estos problemas son de satisfacibilidad booleanos con tres variables cada uno, dichos problemas serán explicados en el desarrollo de este informe, así mismo, se presentará la solución de cada problema.

Con respecto al algoritmo Grover, se presentará la explicación de cómo fue implementado, y a su vez, se presentará la solución que propone el algoritmo para resolver el problema 3-SAT, para eso, primero se presentará la teoría del algoritmo, posteriormente, se explicará cómo el algoritmo de Grover da solución a cada uno de los problemas presentados.

# **2** **Problema 3-SAT**

El problema 3-SAT es la abreviatura del problema 3-satisfacibilidad, este problema es un famoso ejercicio computacional que lo podemos categorizar como un problema NP-completo debido a su complejidad y su poca relación o amistad con los paradigmas computacionales actuales. Este problema consiste en un ejercicio de satisfacibilidad booleana, lo que implica que, su propósito es determinar si existe una asignación de valores de verdad con respecto a un conjunto de variables que participan a su vez de un conjunto de fórmulas booleanas, así que, el objetivo de este problema consiste en determinar las posibles combinaciones que hacen que dichas fórmulas sean verdaderas. Para el caso de este problema 3SAT, las fórmulas se ven expresadas como conjunciones, cada una con exactamente tres variables o sus negaciones.

Este problema se considera como NP-completo, como ya dije antes, gracia a su complejidad, pero a su vez, debemos considerar un factor adicional, y es el crecimiento exponencial de las posibles asignaciones, elemento importante que lo convierte en un problema NP, esto debido a que, el número de asignaciones de valores de verdad a las variables crece de forma exponencial debido al crecimiento a su vez de las variables propuestas, pues como lo hemos visto en anteriores talleres durante el curso de CNYT, si tenemos un conjunto de n variables, podemos decir que existen 2^n posibles combinaciones de valores que pueden hacer la fórmula una expresión satisfacible, esto hace que a medida de que aumente el valor de n, el espacio del problema es insuficiente, esto no es solamente difícil de imaginar, sino que, se convierte en un problema complejo para los algoritmos clásicos.

**2.1** **Problema**

En este caso, se proponen tres ejercicios que se componen por un conjunto de expresiones booleanas, cada expresión consta de tres variables, estos problemas de satisfacibilidad están definidos en forma de conjunciones, por lo tanto nos servirán como ejemplos, pues debemos determinar los valores de las tres variables en cada caso, que satisfacen todas las cláusulas. A continuación se presentan los ejercicios propuestos:

* Ejercicio 1:

Cláusulas:

1. (x O NO y O z)
2. (NO x O y O NO z)
3. (x O y O NO z)

* Ejercicio 2:

Cláusulas:

1. (a O NO b O c)
2. (NO a O NO b O c)
3. (a O b O NO c)

* Ejercicio 3:

Cláusulas:

1. (NO p O q O r)
2. (p O NO q O r)
3. (p O q O NO r)

Para el primer caso, debemos determinar los valores de x,y,z, segundo caso los valores de a,b,c y finalmente para el tercer caso los valores de p,q,r que satisfacen todas las cláusulas según corresponda.

## **2.2** **Determinando los valores de verdad**

Para el ejercicio 1: Problema 3-SAT

| **x** | **y** | **z** | **NOT(x)** | **NOT(y)** | **NOT(z)** | **(x OR NOT y OR z)** | **(NOT x OR y OR NOT z)** | **(x OR y OR NOT z)** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |

Como podemos observar, las posibles valuaciones que satisfacen todas las cláusulas son:

Para el ejercicio 2: Problema 3-SAT

| **a** | **b** | **c** | **NOT(a)** | **NOT(b)** | **NOT(c)** | **(a OR NOT b OR c)** | **(NOT a OR NOT b OR c)** | **(a OR b OR NOT c)** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |

Como podemos observar, las posibles valuaciones que satisfacen todas las cláusulas son:

Para el ejercicio 3: Problema 3-SAT

| **p** | **q** | **r** | **NOT(p)** | **NOT(q)** | **NOT(r)** | **(NOT p OR q OR r)** | **(p OR NOT q OR r)** | **(p OR q OR NOT r)** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |

Como podemos observar, las posibles valuaciones que satisfacen todas las cláusulas son:

# **3** **Algoritmo de Grover**

El algoritmo de Grover es un algoritmo de búsqueda cuántica que nos ofrece una búsqueda rápida para búsquedas no estructuradas. En este caso, este algoritmo encuentra con alta probabilidad la única entrada a una función de caja negra que produce un valor de salida particular.

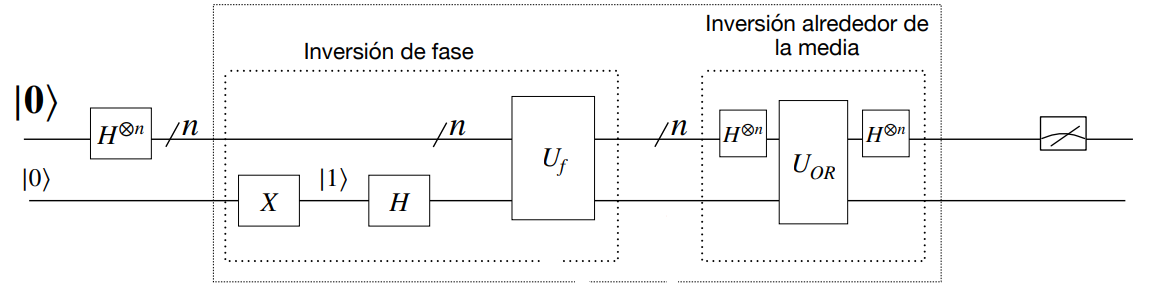
La estrategia del algoritmo de Grover, al igual que en todos los algoritmos cuánticos, es iniciar los qubits en un estado clásico, poner los qubits en superposición, realizar las operaciones en los qubits y finalmente realizar la observación.

## **3.1** **Problema**

El problema que deseamos resolver consiste en la búsqueda algorítmica de un número con una condición particular, para esto, modelamos varias posibilidades para encontrar la que satisface la condición. En este caso, nos dan una función , si evaluamos el peor de los casos, tendríamos que evaluar las 2^n cadenas posibles para satisfacer la condición planteada, eso sucedería si analizamos el problema de búsqueda en un computador clásico, ahora bien, si los planteamos mediante el algoritmo de Grover, en el peor de los casos la complejidad pasa de ser 2^n a .

En este caso, los problemas propuestos anteriormente como ejemplos del problema 3-SAT, serán solucionados mediante el algoritmo de Grover, para esto primero se representará cada una de las variables y su negación como estados cuánticos, posteriormente, usaremos el algoritmo de Grover para amplificar la amplitud de probabilidad de los estados que satisfacen todas las cláusulas y finalmente mediremos el registro cuántico para observar la solución.

El circuito a implementar será el siguiente:

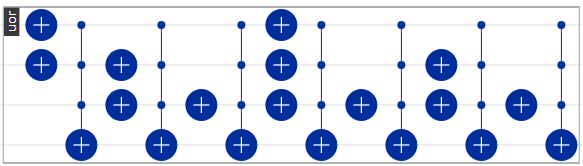


*Figura 1.*

## **3.2** **Implementando el algoritmo de Grover en un computador cuántico**

Mediante la página Composer IBM, creamos los circuitos correspondientes para cada caso como se puede ver a continuación, sin embargo, como se dijo anteriormente, fue necesario repetir el circuito dos veces para cada ejercicio, pues con dos veces fue suficiente para conseguir los estados probabilísticos. A continuación se muestra el circuito para cada ejercicio:

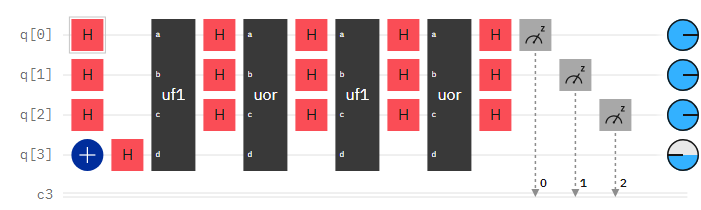
Primero es importante definir el circuito Uor, para definir este circuito, bastó con la formulación de la tabla de verdad de la disyunción entre tres variables, para el primer problema la disyunción sería (x or y or z), para el segundo (a or b or c) y finalmente para el tercero (p or q or r), como se puede evidenciar, la tabla de verdad es igual sin importar el ejercicio, la única valuación que no la satisface es donde todas la variables son falsas, por la tanto, el circuito correspondiente a Uor es:



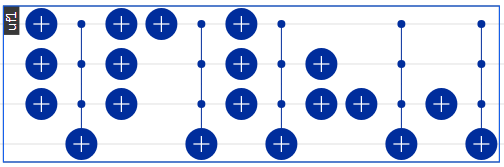
Los circuitos nombrados uf1, uf y uf3, son circuitos que representan la matriz controlada Uf que actúa en el algoritmo de Grover como se puede evidenciar en la *Figura 1,*

estos circuitos se desarrollaron teniendo en cuenta las valuaciones que satisfacen todas las cláusulas presentadas y resaltadas en las tablas anteriores.

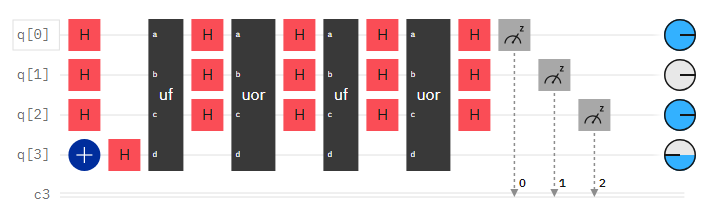
Ahora bien, el circuito correspondiente para las cláusulas con variables xyz es:



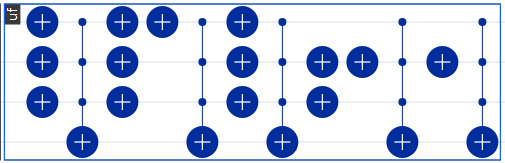
Donde uf1 es:



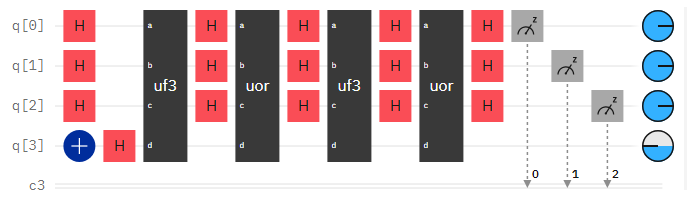
El circuito correspondiente para las cláusulas con variables abc es:



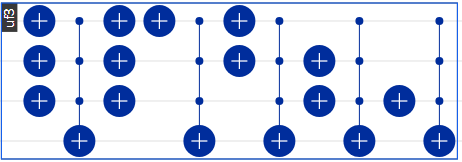
Donde uf es:



Ahora bien, el circuito correspondiente para las cláusulas con variables pqr es:



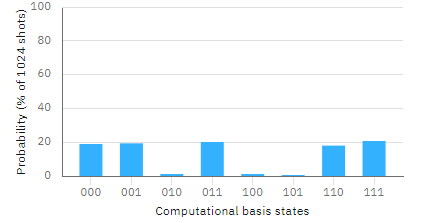
Donde uf3 es:



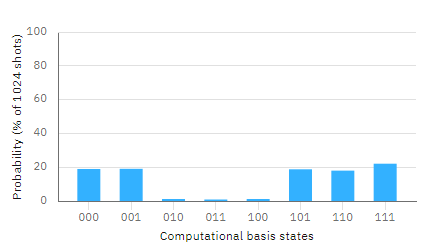
## **3.3** **Resultados tras la implementación del algoritmo de Grover**

En el simulador de IBM obtenemos estas probabilidades, como podemos observar, los estados que mayor probabilidad tienen, coinciden en cada caso con los estados que satisfacen todas las cláusulas:

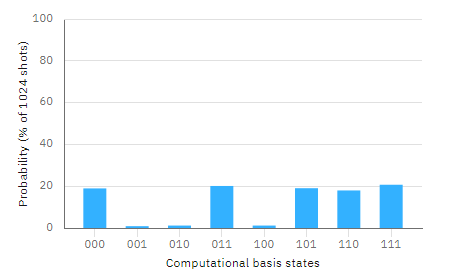
Para las cláusulas con variables xyz:



Para las cláusulas con variables abc:

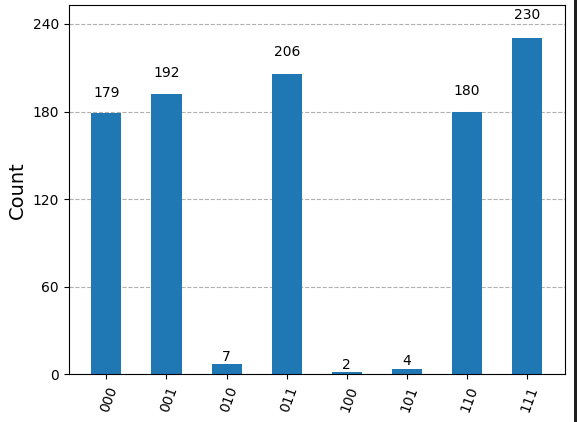


Para las cláusulas con variables pqr:

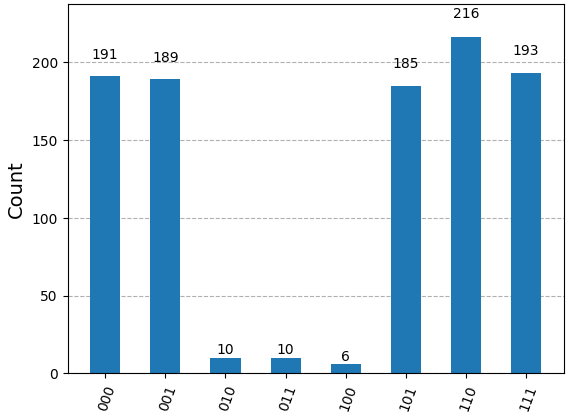


Ahora bien, los resultados obtenidos al ejecutar cada simulación en un editor de código, obtenemos los siguientes resultados que también coinciden con los valores que satisfacen las cláusulas evaluadas en las tablas anteriores:

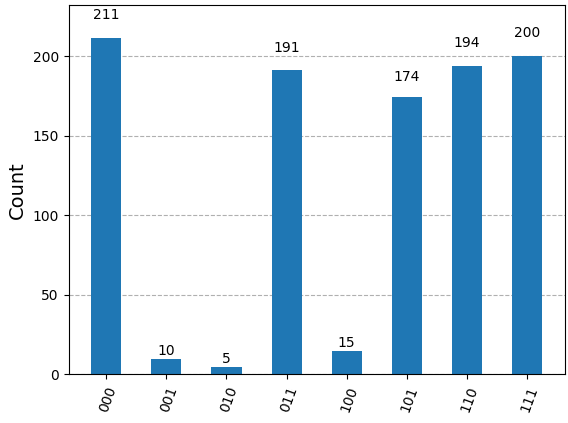
Para las cláusulas con variables xyz:



Para las cláusulas con variables abc:



Para las cláusulas con variables pqr:



# **4** **Conclusiones**

¿Qué presentó?

Se presentó el cálculo de las valuaciones que satisfaces todas las cláusulas, podemos ver cómo determinar qué estados cumplen una condición específica, en este caso satisfacer todas las expresiones booleanas, así mismo, se implementó el algoritmo de Grover para dar solución un problema clásico, el problema 3-SAT. Despues de la implementación teniendo en cuenta la condición anteriormente mencionada, realizamos las simulaciones y hallamos los valores que cumplen dicha condición, esto fue gracias a los conocimientos adquiridos durante el tercer corte del curso de CNYT, pues fue necesario implementar circuitos y así mismo, diseñar estos mismo circuitos mediante la combinación de compuertas y controladores.

¿Qué entendió?

Entendí como funciona el algoritmo de Grover, este algoritmo fue de suma importancia para encontrar una solución óptima para el problema 3-SAT,así mismo fue interesante ver como la combinación de compuertas y la implementación de funciones lógicas mediante circuitos, forman un algoritmo de búsqueda tan preciso.

¿Qué aprendió?

Aprendí a implementar el algoritmo de Grover y como analizar sus resultados, pues hay que tener en cuenta que, cada algoritmo, si bien, su estructura es simular, las funciones lógicas son diferentes y esto causa que obtengamos valores distintos.

¿Qué proyección le ve a lo aprendido?

Este algoritmo es de suma importancia para resolver problemas de búsqueda, especialmente este tipo de problemas como el 3-SAT, pues dan una muy buena solución a problemas de alta complejidad.

# **5** **Bibliografía**

<https://utimaco.com/es/servicio/base-de-conocimientos/criptografia-postcuantica/que-es-el-algoritmo-de-grover>