
Fundamentos de lenguajes de programación cuántica

Día 2:
~ Algoritmos cuánticos ~

Alejandro Díaz-Caro

Universidad Nacional de Quilmes

22º Escuela de Verano de Ciencias Informáticas
Río Cuarto, Córdoba – 9 al 14 de febrero de 2015

Algoritmos más conocidos y criptografía

- ▶ Deutsch
- ▶ Deutsch-Jotza
- ▶ Grover
- ▶ BB84

Algoritmo de Deutsch

Objetivo:

Dado un “oráculo” U_f que implementa la función $f : \{0, 1\} \rightarrow \{0, 1\}$, **determinar si f es constante o no**

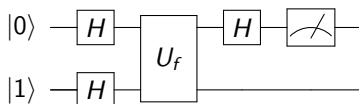
$$U_f|x, y\rangle = |x, y \oplus f(x)\rangle$$

Algoritmo de Deutsch

Objetivo:

Dado un “oráculo” U_f que implementa la función $f : \{0, 1\} \rightarrow \{0, 1\}$, **determinar si f es constante o no**

$$U_f|x, y\rangle = |x, y \oplus f(x)\rangle$$

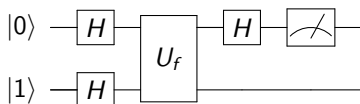


Algoritmo de Deutsch

Objetivo:

Dado un “oráculo” U_f que implementa la función $f : \{0, 1\} \rightarrow \{0, 1\}$, **determinar si f es constante o no**

$$U_f|x, y\rangle = |x, y \oplus f(x)\rangle$$



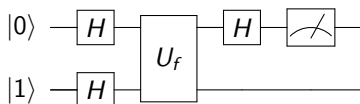
$|01\rangle \xrightarrow{\text{Deutsch alg.}} \dots$

Algoritmo de Deutsch

Objetivo:

Dado un "oráculo" U_f que implementa la función $f : \{0, 1\} \rightarrow \{0, 1\}$, **determinar si f es constante o no**

$$U_f|x, y\rangle = |x, y \oplus f(x)\rangle$$



$$|01\rangle \xrightarrow{\text{Deutsch alg.}} \pm |f(0) \oplus f(1)\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] \begin{cases} \xrightarrow{\text{Si es constante}} \pm |0\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] \\ \xrightarrow{\text{Sino}} \pm |1\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] \end{cases}$$

Algoritmo de Deutsch-Jozsa

Objetivo:

Dado un "oráculo" U_f que implementa la función $f : \{0, 1\}^n \rightarrow \{0, 1\}$, **determinar si f es constante o balanceada**

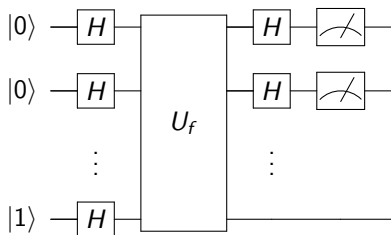
$$U_f |\bar{x}, y\rangle = |\bar{x}, y \oplus f(\bar{x})\rangle$$

Algoritmo de Deutsch-Jozsa

Objetivo:

Dado un "oráculo" U_f que implementa la función $f : \{0, 1\}^n \rightarrow \{0, 1\}$, **determinar si f es constante o balanceada**

$$U_f |\bar{x}, y\rangle = |\bar{x}, y \oplus f(\bar{x})\rangle$$

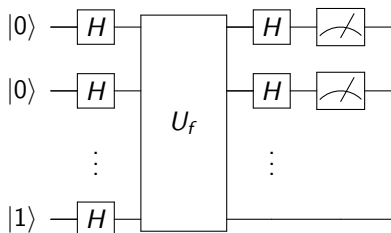


Algoritmo de Deutsch-Jozsa

Objetivo:

Dado un "oráculo" U_f que implementa la función $f : \{0, 1\}^n \rightarrow \{0, 1\}$, **determinar si f es constante o balanceada**

$$U_f |\bar{x}, y\rangle = |\bar{x}, y \oplus f(\bar{x})\rangle$$



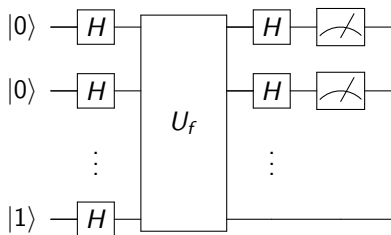
$$|0\rangle^{\otimes n} |1\rangle \xrightarrow{\text{D-J alg.}} \dots \rightarrow$$

Algoritmo de Deutsch-Jozsa

Objetivo:

Dado un "oráculo" U_f que implementa la función $f : \{0, 1\}^n \rightarrow \{0, 1\}$, **determinar si f es constante o balanceada**

$$U_f |\bar{x}, y\rangle = |\bar{x}, y \oplus f(\bar{x})\rangle$$



$$|0\rangle^{\otimes n} |1\rangle \xrightarrow{\text{D-J alg.}} \begin{cases} \text{Si es constante} & \pm |0\rangle^{\otimes n} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] \\ \text{Si es balanceada} & \pm |\psi\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] \end{cases}$$

donde $|\psi\rangle$ no incluye $|0\rangle^{\otimes n}$

Algoritmo de búsqueda de Grover

Preliminares: Oráculo

$$U_f|x, y\rangle = |x, y \oplus f(x)\rangle$$

Tomar $y = |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ entonces

$$\begin{aligned}U_f|x, y\rangle &= U_f\left(|x\rangle \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right) = \frac{1}{\sqrt{2}}(U_f|x, 0\rangle - U_f|x, 1\rangle) \\&= \frac{1}{\sqrt{2}}(|x, f(x)\rangle - |x, 1 \oplus f(x)\rangle) = |x\rangle \frac{1}{\sqrt{2}}(|f(x)\rangle - |1 \oplus f(x)\rangle) \\&= (-1)^{f(x)}|x, y\rangle\end{aligned}$$

Algoritmo de búsqueda de Grover

Preliminares: Oráculo

$$U_f|x, y\rangle = |x, y \oplus f(x)\rangle$$

Tomar $y = |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ entonces

$$\begin{aligned}U_f|x, y\rangle &= U_f\left(|x\rangle \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right) = \frac{1}{\sqrt{2}}(U_f|x, 0\rangle - U_f|x, 1\rangle) \\&= \frac{1}{\sqrt{2}}(|x, f(x)\rangle - |x, 1 \oplus f(x)\rangle) = |x\rangle \frac{1}{\sqrt{2}}(|f(x)\rangle - |1 \oplus f(x)\rangle) \\&= (-1)^{f(x)}|x, y\rangle\end{aligned}$$

U_f no modifica y ... lo omitimos

Oráculo

$$U|x\rangle = (-1)^{f(x)}|x\rangle$$

Algoritmo de búsqueda de Grover

Preliminares: Inversión sobre el promedio

$$|\phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{\bar{x} \in \{0,1\}^n} |\bar{x}\rangle$$

Algoritmo de búsqueda de Grover

Preliminares: Inversión sobre el promedio

$$|\phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{\bar{x} \in \{0,1\}^n} |\bar{x}\rangle = \begin{pmatrix} \frac{1}{\sqrt{2^n}} \\ \vdots \\ \frac{1}{\sqrt{2^n}} \end{pmatrix}_{2^n}$$

Algoritmo de búsqueda de Grover

Preliminares: Inversión sobre el promedio

$$|\phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{\bar{x} \in \{0,1\}^n} |\bar{x}\rangle = \begin{pmatrix} \frac{1}{\sqrt{2^n}} \\ \vdots \\ \frac{1}{\sqrt{2^n}} \end{pmatrix}_{2^n}$$

Inversión sobre el promedio

$$G = 2|\phi\rangle\langle\phi| - I$$

$$\begin{pmatrix} \frac{2}{2^n} - 1 & \frac{2}{2^n} & \cdots & \frac{2}{2^n} \\ \frac{2}{2^n} & \frac{2}{2^n} - 1 & \cdots & \frac{2}{2^n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{2}{2^n} & \frac{2}{2^n} & \cdots & \frac{2}{2^n} - 1 \end{pmatrix}_{2^n \times 2^n}$$

Algoritmo de búsqueda de Grover

Preliminares: Inversión sobre el promedio

$$|\phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{\bar{x} \in \{0,1\}^n} |\bar{x}\rangle = \begin{pmatrix} \frac{1}{\sqrt{2^n}} \\ \vdots \\ \frac{1}{\sqrt{2^n}} \end{pmatrix}_{2^n}$$

Inversión sobre el promedio

$$G = 2|\phi\rangle\langle\phi| - I$$

$$\begin{pmatrix} \frac{2}{2^n} - 1 & \frac{2}{2^n} & \cdots & \frac{2}{2^n} \\ \frac{2}{2^n} & \frac{2}{2^n} - 1 & \cdots & \frac{2}{2^n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{2}{2^n} & \frac{2}{2^n} & \cdots & \frac{2}{2^n} - 1 \end{pmatrix}_{2^n \times 2^n} \quad G \begin{pmatrix} \sum_{\bar{x} \in \{0,1\}^n} a_{\bar{x}} |\bar{x}\rangle \end{pmatrix} \\ = \sum_{\bar{x} \in \{0,1\}^n} (2A - a_{\bar{x}}) |\bar{x}\rangle$$

donde A es el promedio de los $a_{\bar{x}}$

Algoritmo de búsqueda de Grover

El algoritmo

Objetivo:

Localizar el \bar{x}_0 tal que $f(\bar{x}_0) = 1$

1. Aplicar Hadamard a $|0\rangle^{\otimes n}$
2. Aplicar el oráculo U
3. Aplicar la inversión sobre el promedio G
4. Repetir pasos 2 y 3 durante $\left\lfloor \frac{\pi}{4\arcsen(\sqrt{\frac{1}{2^n}})} \right\rfloor$ iteraciones
(cálculo del número óptimo de iteraciones, en el apunte, sección 2.3.4)

EXPLICACIÓN PASO A PASO EN EL PIZARRÓN
(Y EJEMPLO)

Aplicación criptográfica

One-time pad, un método clásico infalible...

b_1	b_2	$b_1 \oplus b_2$
1	1	0
1	0	1
0	1	1
0	0	0

Aplicación criptográfica

One-time pad, un método clásico infalible...

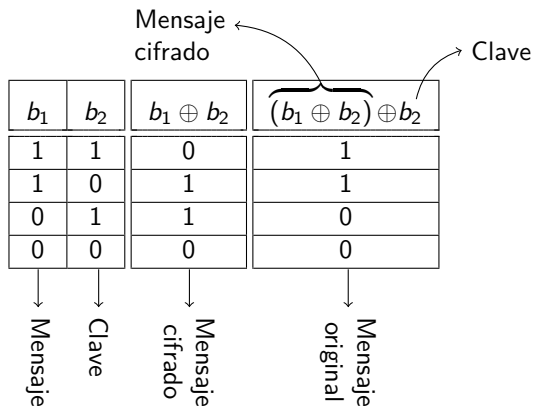
b_1	b_2	$b_1 \oplus b_2$
1	1	0
1	0	1
0	1	1
0	0	0

↓ ↓ ↓

Mensaje Clave Mensaje
cifrado

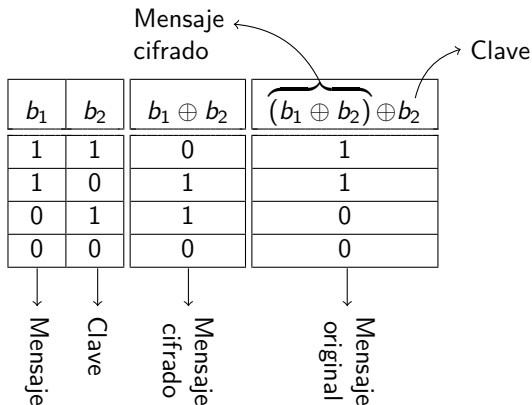
Aplicación criptográfica

One-time pad, un método clásico infalible...



Aplicación criptográfica

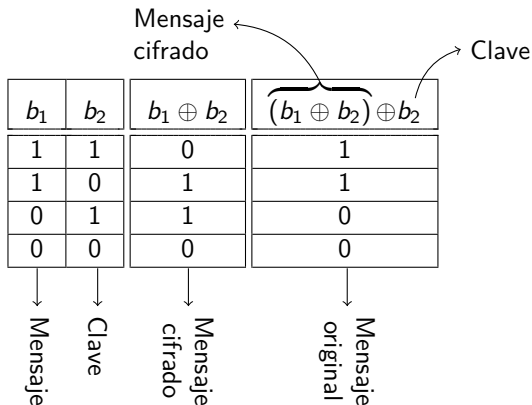
One-time pad, un método clásico infalible...



Probabilidad de adivinar el mensaje original a partir del cifrado: $\frac{1}{2^n}$

Aplicación criptográfica

One-time pad, un método clásico infalible...



Probabilidad de adivinar el mensaje original a partir del cifrado: $\frac{1}{2^n}$

¡Igual que la posibilidad de adivinar el mensaje original sin ninguna información extra!

Aplicación criptográfica

One-time pad, un método clásico infalible...

Entonces, si es tan simple y seguro... ¿porqué no es utilizado?

Aplicación criptográfica

One-time pad, un método clásico infalible...

Entonces, si es tan simple y seguro... ¿porqué no es utilizado?

- ▶ Largo del mensaje = largo de la clave (para 100% de seguridad)
- ▶ Clave de encriptación y desencriptación iguales (y secretas)
 - ▶ Dificultad para distribuir las claves

Aplicación criptográfica

One-time pad, un método clásico infalible...

Entonces, si es tan simple y seguro... ¿porqué no es utilizado?

- ▶ Largo del mensaje = largo de la clave (para 100% de seguridad)
- ▶ Clave de encriptación y desencriptación iguales (y secretas)
 - ▶ Dificultad para distribuir las claves

Ahí entra el método BB84: es un método de *distribución* de claves de manera segura.

Aplicación criptográfica

QKD-BB84

Objetivo: Crear y transmitir una clave de manera segura

Esquema	+	×
Base	$\{ 0\rangle, 1\rangle\}$	$\{ +\rangle, -\rangle\}$
Codif.	$0 = 0\rangle$ $1 = 1\rangle$	$0 = -\rangle$ $1 = +\rangle$

1. A: secuencia aleatoria de 0s o 1s y elección aleatoria de esquemas para c/bit
2. B: elección aleatoria del esquema de medición para cada bit recibido
3. A: transmite la sucesión de esquemas empleada
4. B: informa en qué casos coincidieron
5. La clave queda definida por los bits donde se usaron los mismos esquemas
6. Intercambio de hashes para verificación

Aplicación criptográfica

QKD-BB84: Ejemplo

$$+ : 0 = |0\rangle, \quad 1 = |1\rangle \qquad \times : 0 = |-\rangle, \quad 1 = |+\rangle$$

1. A: secuencia aleatoria de 0s o 1s y elección aleatoria de esquemas para c/bit
2. B: elección aleatoria del esquema de medición para cada bit recibido
3. A: transmite la sucesión de esquemas empleada
4. B: informa en qué casos coincidieron
5. La clave queda definida por los bits donde se usaron los mismos esquemas
6. Intercambio de hashes para verificación

Bits de A	1	0	0	1	0	0	0	1
Esquemas de A	×	+	+	×	×	+	×	+
Valores de A	$ +\rangle$	$ 0\rangle$	$ 0\rangle$	$ +\rangle$	$ -\rangle$	$ 0\rangle$	$ -\rangle$	$ 1\rangle$
Esquemas de B	+	×	+	×	+	+	×	×
Valores de B	$ 0\rangle$	$ +\rangle$	$ 0\rangle$	$ +\rangle$	$ 1\rangle$	$ 0\rangle$	$ -\rangle$	$ -\rangle$

Aplicación criptográfica

QKD-BB84: Ejemplo

$$+ : 0 = |0\rangle, \quad 1 = |1\rangle \qquad \times : 0 = |-\rangle, \quad 1 = |+\rangle$$

1. A: secuencia aleatoria de 0s o 1s y elección aleatoria de esquemas para c/bit
2. B: elección aleatoria del esquema de medición para cada bit recibido
3. A: transmite la sucesión de esquemas empleada
4. B: informa en qué casos coincidieron
5. La clave queda definida por los bits donde se usaron los mismos esquemas
6. Intercambio de hashes para verificación

Bits de A	1	0	0	1	0	0	0	1
Esquemas de A	×	+	+	×	×	+	×	+
Valores de A	$ +\rangle$	$ 0\rangle$	$ 0\rangle$	$ +\rangle$	$ -\rangle$	$ 0\rangle$	$ -\rangle$	$ 1\rangle$
Esquemas de B	+	×	+	×	+	+	×	×
Valores de B	$ 0\rangle$	$ +\rangle$	$ 0\rangle$	$ +\rangle$	$ 1\rangle$	$ 0\rangle$	$ -\rangle$	$ -\rangle$
Coincidencias			✓	✓		✓	✓	

Aplicación criptográfica

QKD-BB84: Ejemplo

$$+ : 0 = |0\rangle, \quad 1 = |1\rangle \qquad \times : 0 = |-\rangle, \quad 1 = |+\rangle$$

1. A: secuencia aleatoria de 0s o 1s y elección aleatoria de esquemas para c/bit
2. B: elección aleatoria del esquema de medición para cada bit recibido
3. A: transmite la sucesión de esquemas empleada
4. B: informa en qué casos coincidieron
5. La clave queda definida por los bits donde se usaron los mismos esquemas
6. Intercambio de hashes para verificación

Bits de A	1	0	0	1	0	0	0	1
Esquemas de A	×	+	+	×	×	+	×	+
Valores de A	$ +\rangle$	$ 0\rangle$	$ 0\rangle$	$ +\rangle$	$ -\rangle$	$ 0\rangle$	$ -\rangle$	$ 1\rangle$
Esquemas de B	+	×	+	×	+	+	×	×
Valores de B	$ 0\rangle$	$ +\rangle$	$ 0\rangle$	$ +\rangle$	$ 1\rangle$	$ 0\rangle$	$ -\rangle$	$ -\rangle$
Coincidencias			✓	✓		✓	✓	
Clave			0	1		0	0	

Aplicación criptográfica

QKD-BB84: Inviolabilidad (teórica)

Agregamos un espía: C

Aplicación criptográfica

QKD-BB84: Inviolabilidad (teórica)

Agregamos un espía: C

- ▶ A envía 0 con esquema $\times: |-\rangle$

Aplicación criptográfica

QKD-BB84: Inviolabilidad (teórica)

Agregamos un espía: C

- ▶ A envía 0 con esquema \times : $|-\rangle$
- ▶ Si C usa esquema $+$, el estado pasa a $|0\rangle$ o $|1\rangle$

Aplicación criptográfica

QKD-BB84: Inviolabilidad (teórica)

Agregamos un espía: C

- ▶ A envía 0 con esquema \times : $|-\rangle$
- ▶ Si C usa esquema $+$, el estado pasa a $|0\rangle$ o $|1\rangle$
- ▶ Si B usa esquema \times , obtiene $|-\rangle$ con probabilidad $\frac{1}{2}$ y $|+\rangle$ con probabilidad $\frac{1}{2}$

Aplicación criptográfica

QKD-BB84: Inviolabilidad (teórica)


Agregamos un espía: C

- ▶ A envía 0 con esquema \times : $|-\rangle$
- ▶ Si C usa esquema $+$, el estado pasa a $|0\rangle$ o $|1\rangle$
- ▶ Si B usa esquema \times , obtiene $|-\rangle$ con probabilidad $\frac{1}{2}$ y $|+\rangle$ con probabilidad $\frac{1}{2}$
- ▶ Mientras más bits se envían, la probabilidad de no detectar a C decrece exponencialmente:

Bits	Probabilidad
1 bit	$3/4 = 0,75$
8 bits	$(3/4)^8 = 0,10011$
128 bits	$(3/4)^{128} = 1,018 \times 10^{-16}$
1Mb	$(3/4)^{1024} = 1,155 \times 10^{-128}$
1MB	$(3/4)^{8192} = 3,17 \times 10^{-1024}$

Aplicación criptográfica

QKD-BB84 en la vida real



T: +41 22 301 83 71 | E: info@idquantique.com


- Random Number Generation
- Quantum-Safe Crypto
- Photon Counting
- Company
- News
- Contacts

Quantum Safe Crypto

High performance network encryption, quantum key generation and quantum key distribution (QKD) technologies.

Home > Quantum-Safe Crypto > Quantum Key Distribution

Cerberis Quantum Key Distribution (QKD) Server



ID Quantique's Cerberis solution is the ultimate in quantum-safe cryptography.

Combined with IDQ's **Centauris** high-speed layer 2 encryptors, it guarantees long-term protection of data into the quantum era, when the massive processing power of quantum computers will break today's public key exchange mechanisms.

The Cerberis quantum key distribution (QKD) platform generates secure shared keys...

PRODUCTS

- Centauris L2 encryption (pdf)
- Cerberis QKD Server (pdf)
- Clavis² QKD Research Platform (pdf)

QKD USER CASES

- Gigabit Ethernet Government Network
- 10G Ethernet Encryption for Disaster Recovery Center
- Colt QKD as a Service

WHITE PAPERS

Ejercicio

En una lista de 1000 elementos,

1. ¿Cuál es el número óptimo de iteraciones del algoritmo de Grover?
2. ¿Cuál es la probabilidad de error?