Procesamiento Cuántico de Información

Ariel Bendersky[1]

[1]Departamento de Computación - FCEyN - Universidad de Buenos Aires

Clase 4

Tarea - Mala onda

Sea $|\psi\rangle$ un estado de un qubit. Demuestre que los tres valores medios $\langle\psi|\,\sigma_x\,|\psi\rangle$, $\langle\psi|\,\sigma_y\,|\psi\rangle$ y $\langle\psi|\,\sigma_z\,|\psi\rangle$ identifican unívocamente el estado $|\psi\rangle$.

Clase 4

- Evolución temporal.
- El modelo de circuitos.
- Conjuntos universales de compuertas.
- Jugando con IBM y Rigetti.

Clase 4

- Evolución temporal.
- El modelo de circuitos.
- Conjuntos universales de compuertas.
- Jugando con IBM y Rigetti.

Clase 4

- Evolución temporal.
- El modelo de circuitos.
- Conjuntos universales de compuertas.
- Jugando con IBM y Rigetti.

Clase 4

- Evolución temporal.
- El modelo de circuitos.
- Conjuntos universales de compuertas.
- Jugando con IBM y Rigetti.

La evolución temporal

Evolución unitaria

La evolución de un sistema cuántico en un estado $|\psi_{inicial}\rangle$ está dada por:

$$|\psi_{inicial}\rangle \longrightarrow |\psi_{final}\rangle = U\,|\psi_{inicial}\rangle$$

donde U es un operador (matriz) unitario: $U^\dagger = U^{-1}$.
Un ingrediente de la computación cuántica es controlar esas operaciones U.

Ejemplo

$|0\rangle \longrightarrow |1\rangle$ se hace mediante

$$U = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Porque $|1\rangle = U\,|0\rangle$.

La evolución temporal

Evolución unitaria

La evolución de un sistema cuántico en un estado $|\psi_{inicial}\rangle$ está dada por:

$$|\psi_{inicial}\rangle \longrightarrow |\psi_{final}\rangle = U\,|\psi_{inicial}\rangle$$

donde U es un operador (matriz) unitario: $U^\dagger = U^{-1}$.
Un ingrediente de la computación cuántica es controlar esas operaciones U.

Ejemplo

$|0\rangle \longrightarrow |1\rangle$ se hace mediante

$$U = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Porque $|1\rangle = U\,|0\rangle$.

La evolución temporal

Importante

Toda operación unitaria se puede implementar. Es decir, se pueden forzar interacciones que hagan que un sistema evolucione de acuerdo a un operador unitario dado.

Computación cuántica

Cómputos cuánticos

Los cómputos cuánticos tienen tres etapas:

- Preparar un estado trivial. Típicamente: $|000...0\rangle = |\bar{0}\rangle$.
- Aplicar una operación unitaria U al estado. Dicha operación es la implementación del *algoritmo cuántico*.
- Realizar una medición trivial. Típicamente, medir proyectivamente en la base computacional.
  - Interpretar el resultado de la medición.

Es decir

$$|\bar{0}\rangle$$

Computación cuántica

Cómputos cuánticos

Los cómputos cuánticos tienen tres etapas:

- Preparar un estado trivial. Típicamente: $|000...0\rangle = |\bar{0}\rangle$.
- Aplicar una operación unitaria U al estado. Dicha operación es la implementación del *algoritmo cuántico*.
- Realizar una medición trivial. Típicamente, medir proyectivamente en la base computacional.
  - Interpretar el resultado de la medición.

Es decir

$$|\bar{0}\rangle$$

Computación cuántica

Cómputos cuánticos

Los cómputos cuánticos tienen tres etapas:

- Preparar un estado trivial. Típicamente: $|000...0\rangle = |\bar{0}\rangle$.
- Aplicar una operación unitaria U al estado. Dicha operación es la implementación del *algoritmo cuántico*.
- Realizar una medición trivial. Típicamente, medir proyectivamente en la base computacional.
  - Interpretar el resultado de la medición.

Es decir

$$U\,|\bar{0}\rangle$$

Computación cuántica

Cómputos cuánticos

Los cómputos cuánticos tienen tres etapas:

- Preparar un estado trivial. Típicamente: $|000...0\rangle = |\bar{0}\rangle$.
- Aplicar una operación unitaria U al estado. Dicha operación es la implementación del *algoritmo cuántico*.
- Realizar una medición trivial. Típicamente, medir proyectivamente en la base computacional.
  - Interpretar el resultado de la medición.

Es decir

$$p_i = \langle\bar{0}| U^\dagger \Pi_i U |\bar{0}\rangle$$

Complejidad cuántica

Dificultad

La complejidad (cantidad necesaria de recursos) de un algoritmo cuántico tiene que ver con la dificultad de implementar U.

Por eso

Exigimos que el estado inicial y la medición sean triviales. Si no, es simple trivializar U:

$$p_i = \langle \bar{0} | U^\dagger \Pi_i U | \bar{0} \rangle = \langle \Psi | \Pi_i | \Psi \rangle = \langle \bar{0} | \tilde{\Pi}_i | \bar{0} \rangle$$

donde $|\Psi\rangle = U|\bar{0}\rangle$ y $\tilde{\Pi}_i = U^\dagger \Pi_i U$. Fijando el estado inicial y la medición metemos toda la complejidad en U.

Complejidad cuántica

Si todos los algoritmos cuánticos consisten en aplicar una operación unitaria. ¿Por qué algunos algoritmos son más complejos –difíciles– que otros?
Los próximos slides veremos eso:

- El modelo de circuitos.
- Conjuntos universales de compuertas cuánticas.

Formalizaremos levemente la noción de complejidad.

Complejidad cuántica

Si todos los algoritmos cuánticos consisten en aplicar una operación unitaria. ¿Por qué algunos algoritmos son más complejos –difíciles– que otros?
Los próximos slides veremos eso:

- El modelo de circuitos.
- Conjuntos universales de compuertas cuánticas.

Formalizaremos levemente la noción de complejidad.

Complejidad cuántica

Si todos los algoritmos cuánticos consisten en aplicar una operación unitaria. ¿Por qué algunos algoritmos son más complejos –difíciles– que otros?
Los próximos slides veremos eso:

- El modelo de circuitos.
- Conjuntos universales de compuertas cuánticas.

Formalizaremos levemente la noción de complejidad.

El modelo de circuitos

El modelo de circuitos

Es simplemente una representación gráfica de las operaciones unitarias.

Ejemplo

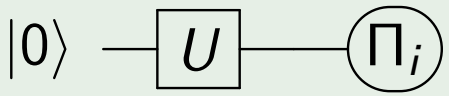
$$p_i = \langle 0 | U^\dagger \Pi_i U | 0 \rangle$$

El modelo de circuitos

El modelo de circuitos

Es simplemente una representación gráfica de las operaciones unitarias.

Ejemplo



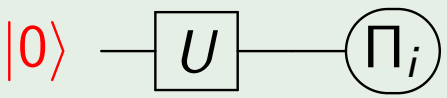
$$p_i = \langle 0 | U^\dagger \Pi_i U | 0 \rangle$$

El modelo de circuitos

El modelo de circuitos

Es simplemente una representación gráfica de las operaciones unitarias.

Ejemplo



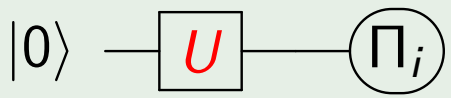
$$p_i = \langle 0 | U^\dagger \Pi_i U | 0 \rangle$$

El modelo de circuitos

El modelo de circuitos

Es simplemente una representación gráfica de las operaciones unitarias.

Ejemplo



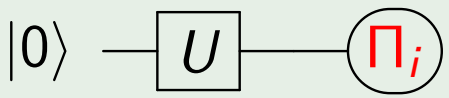
$$p_i = \langle 0 | U^\dagger \Pi_i U | 0 \rangle$$

El modelo de circuitos

El modelo de circuitos

Es simplemente una representación gráfica de las operaciones unitarias.

Ejemplo



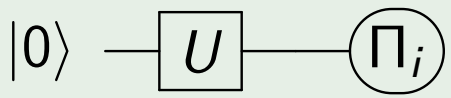
$$p_i = \langle 0 | U^\dagger \Pi_i U | 0 \rangle$$

El modelo de circuitos

El modelo de circuitos

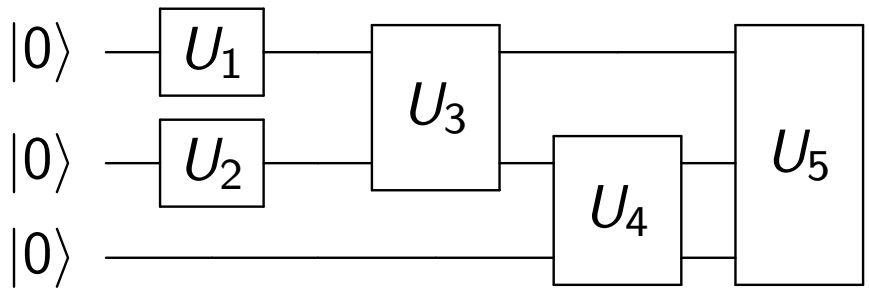
Es simplemente una representación gráfica de las operaciones unitarias.

Ejemplo



$$p_i = \langle 0 | U^\dagger \Pi_i U | 0 \rangle$$

Circuitos de muchos qubits



Se entiende como:

$$U_5^{(1,2,3)}(Id^{(1)} \otimes U_4^{(2,3)})(U_3^{(1,2)} \otimes Id^{(3)})(U_1^{(1)} \otimes U_2^{(2)} \otimes Id^{(3)})\,|000\rangle$$

Algunas unitarias útiles

Unitarias de un solo qubit.

$$Id = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad X = \sigma_X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$Y = \sigma_Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \sigma_Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

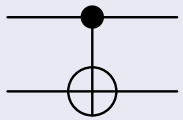
$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

$$T = U_{\pi/8} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$$

Algunas unitarias útiles

Controlled not

Es una compuerta de negación controlada que hace interactuar dos qubits.



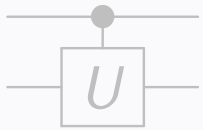
$$CNOT\,|00\rangle = |00\rangle$$

$$CNOT\,|01\rangle = |01\rangle$$

$$CNOT\,|10\rangle = |11\rangle$$

$$CNOT\,|11\rangle = |10\rangle$$

En general: Controlled U



Es una compuerta U controlada.

$$CU\,|00\rangle = |00\rangle$$

$$CU\,|01\rangle = |01\rangle$$

$$CU\,|10\rangle = |1\rangle\,U\,|0\rangle$$

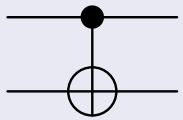
$$CU\,|11\rangle = |1\rangle\,U\,|1\rangle$$

Con $U = X$ se obtiene $CNOT$.

Algunas unitarias útiles

Controlled not

Es una compuerta de negación controlada que hace interactuar dos qubits.



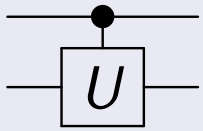
$$CNOT\,|00\rangle = |00\rangle$$

$$CNOT\,|01\rangle = |01\rangle$$

$$CNOT\,|10\rangle = |11\rangle$$

$$CNOT\,|11\rangle = |10\rangle$$

En general: Controlled U



Es una compuerta U controlada.

$$CU\,|00\rangle = |00\rangle$$

$$CU\,|01\rangle = |01\rangle$$

$$CU\,|10\rangle = |1\rangle\,U\,|0\rangle$$

$$CU\,|11\rangle = |1\rangle\,U\,|1\rangle$$

Con $U = X$ se obtiene $CNOT$.

Conjuntos universales de compuertas cuánticas

Conjuntos universales de compuertas cuánticas

Conjuntos universales de compuertas cuánticas

Definición

Decimos que un conjunto $\mathcal{S}$ de compuertas cuánticas es universal si cualquier unitaria U (de cualquier número de qubits) puede aproximarse por una secuencia de aplicaciones de operaciones de $\mathcal{S}$.

Nuestro caso de interés

CNOT y operaciones de un solo qubit son universales.

Conjuntos universales de compuertas cuánticas

Definición

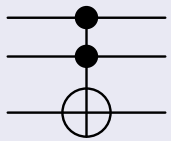
Decimos que un conjunto $\mathcal{S}$ de compuertas cuánticas es universal si cualquier unitaria U (de cualquier número de qubits) puede aproximarse por una secuencia de aplicaciones de operaciones de $\mathcal{S}$.

Nuestro caso de interés

CNOT y operaciones de un solo qubit son universales.

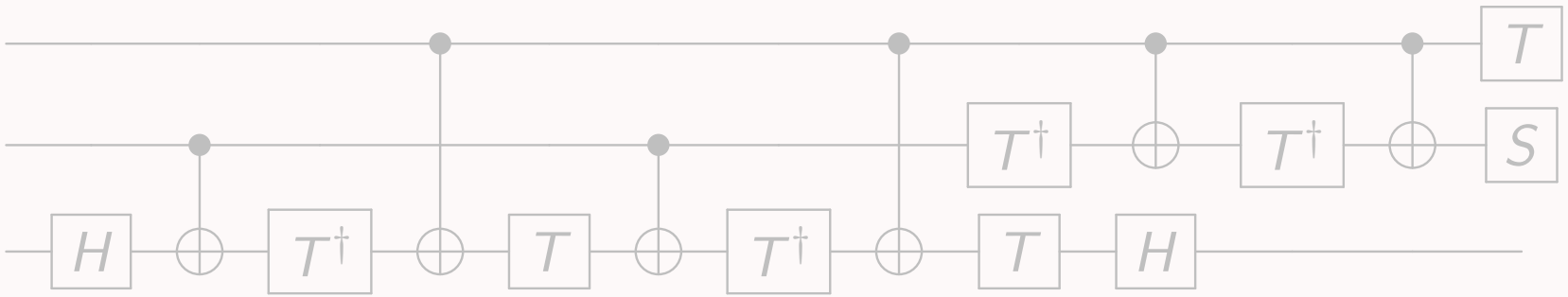
Ideas para la demostración

La compuerta de Toffoli



Niega el tercer qubit cuando los dos primeros tienen un 1.

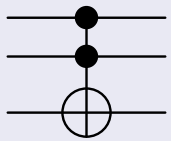
Implementación



Puedo usar Toffoli tranquilo.

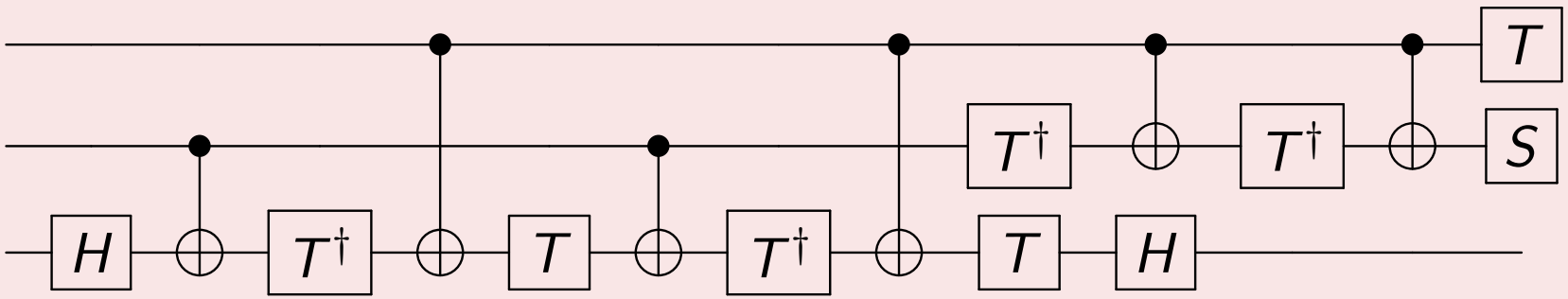
Ideas para la demostración

La compuerta de Toffoli



Niega el tercer qubit cuando los dos primeros tienen un 1.

Implementación



Puedo usar Toffoli tranquilo.

Resultado intermedio

Toda matriz unitaria de NxN puede construirse a partir del producto de matrices de la forma

$$V_i = \begin{pmatrix} 1 & 0 & \cdots & & & \cdots & 0 \\ 0 & 1 & \cdots & & & \cdots & 0 \\ \vdots & \vdots & \ddots & & & & \vdots \\ & & & a & b & & \\ & & & c & d & & \\ \vdots & \vdots & & & & \ddots & \vdots \\ 0 & 0 & \cdots & & & \cdots & 1 \end{pmatrix}$$

llamadas matrices de dos niveles.

Demostración

Primero notemos que para todo vector (x, y) existe una unitaria V tal que:

$$V \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \sqrt{xx^* + yy^*} \\ 0 \end{pmatrix}$$

En efecto,

$$V = \frac{1}{\sqrt{xx^* + yy^*}} \begin{pmatrix} x^* & y^* \\ -y & x \end{pmatrix}$$

cumple eso y es unitaria.

Demostración

De igual forma, para un vector $|\zeta\rangle = (\zeta_1, ..., \zeta_N)$ vale que:

$$V_{N-1}...V_1 |\zeta\rangle = \begin{pmatrix} \langle\zeta|\zeta\rangle \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Si la inversa de U es:

$$U^\dagger = \begin{pmatrix} \zeta_1 & \zeta_2' & \cdots & \zeta_N' \\ \zeta_2 & \ddots & & \\ \vdots & & & \\ \zeta_N & \cdots & & \end{pmatrix}$$

tenemos que $V_{N-1}...V_1U^\dagger$ es:

$$V_{N-1}...V_1U^\dagger = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & U'^\dagger & \\ 0 & & & \end{pmatrix}$$

Repetimos inductivamente y listo.

Lo que falta

Falta ver que cada unitaria de dos niveles se puede escribir como $CNOT$ y unitarias de un solo qubit.

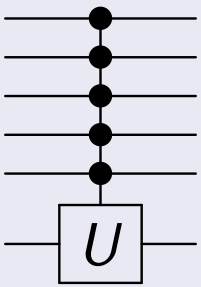
Paso 1 - Idea

Si V es una unitaria de dos niveles, llevar esos niveles al 11..11 y el 11..10. La idea es usar los dos niveles originales, por ejemplo 100101 y 001000 y usar un bit en el que difieran para hacer, con $CNOT$ o $(X \otimes Id)CNOT(X \otimes Id)$ los cambios en cada bit en el que difieren. Finalmente, un swap del qubit que usamos de control con el último.

Lo que falta

Paso 2 - Final

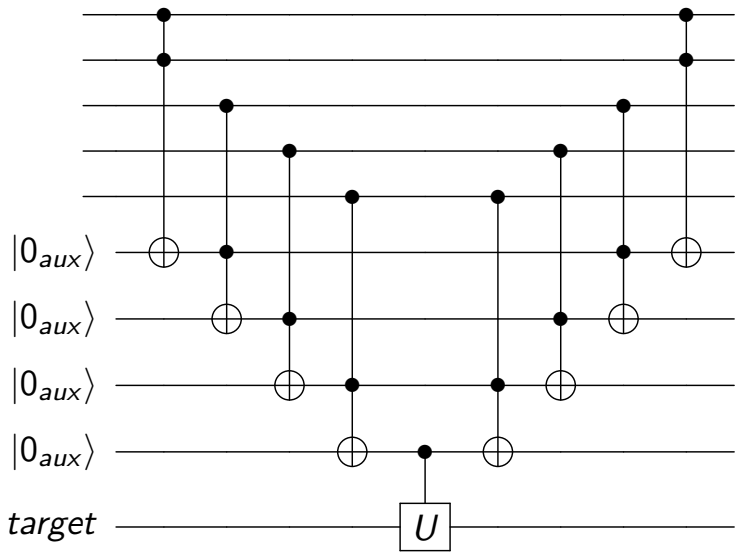
Hacer una unitaria en el último qubit controlada por todos los demás:



¿Cómo se hace con *CNOT* y unitarias de un qubit?

Lo que falta

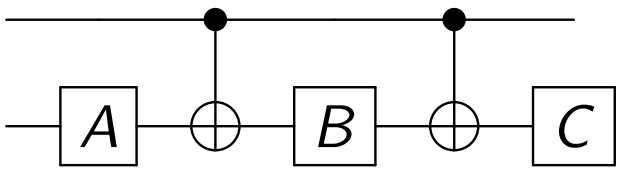
Una forma es agregar $n-1$ qubits auxiliares y usar compuertas de Tofolli (que sabemos implementar).



El truco es que el anteúltimo qubit es 1 si todos los anteriores lo son, por eso lo uso para controlar.

Lo que falta

Como realizar U de manera controlada. Se puede hacer mediante:



En resumen

Resultado principal

Toda unitaria se puede implementar con compuertas *CNOT*, que hacen interactuar los qubits de a dos, y unitarias de un solo qubit.

Pregunta pendiente

¿Hay operaciones unitarias más simples que otras? Sí. Llamaremos complejidad de la implementación de una operación unitaria a la cantidad de compuertas *CNOT* y de un qubit que son necesarias en dicha implementación.
Una definición asintóticamente equivalente es la cantidad de compuertas *CNOT* necesarias.

Importante

Dada una unitaria, no hay un método para saber la mejor manera de implementarla. De ahí que se sigan optimizando algunas unitarias conocidas.

En resumen

Resultado principal

Toda unitaria se puede implementar con compuertas *CNOT*, que hacen interactuar los qubits de a dos, y unitarias de un solo qubit.

Pregunta pendiente

¿Hay operaciones unitarias más simples que otras? Sí. Llamaremos complejidad de la implementación de una operación unitaria a la cantidad de compuertas *CNOT* y de un qubit que son necesarias en dicha implementación.
Una definición asintóticamente equivalente es la cantidad de compuertas *CNOT* necesarias.

Importante

Dada una unitaria, no hay un método para saber la mejor manera de implementarla. De ahí que se sigan optimizando algunas unitarias conocidas.

En resumen

Resultado principal

Toda unitaria se puede implementar con compuertas *CNOT*, que hacen interactuar los qubits de a dos, y unitarias de un solo qubit.

Pregunta pendiente

¿Hay operaciones unitarias más simples que otras? Sí. Llamaremos complejidad de la implementación de una operación unitaria a la cantidad de compuertas *CNOT* y de un qubit que son necesarias en dicha implementación.
Una definición asintóticamente equivalente es la cantidad de compuertas *CNOT* necesarias.

Importante

Dada una unitaria, no hay un método para saber la mejor manera de implementarla. De ahí que se sigan optimizando algunas unitarias conocidas.

El algoritmo de Deutsch

El problema clásico

Sea una función $f : \{0,1\} \to f : \{0,1\}$ desconocida. Se quiere saber si f es constante ($f(0) = f(1)$) o no. ¿Cuántas veces es necesario evaluar f para saber eso?

Solución clásica

Se ve trivialmente que hay que averiguar tanto $f(0)$ como $f(1)$ y, por lo tanto, es necesario evaluar la función dos veces para determinar si la función es constante.

Ventaja cuántica

Evaluando la función f una sola vez podemos determinar si es constante o no.

El algoritmo de Deutsch

El problema clásico

Sea una función $f : \{0,1\} \to f : \{0,1\}$ desconocida. Se quiere saber si f es constante ($f(0) = f(1)$) o no. ¿Cuántas veces es necesario evaluar f para saber eso?

Solución clásica

Se ve trivialmente que hay que averiguar tanto $f(0)$ como $f(1)$ y, por lo tanto, es necesario evaluar la función dos veces para determinar si la función es constante.

Ventaja cuántica

Evaluando la función f una sola vez podemos determinar si es constante o no.

El algoritmo de Deutsch

El problema clásico

Sea una función $f : \{0,1\} \to f : \{0,1\}$ desconocida. Se quiere saber si f es constante ($f(0) = f(1)$) o no. ¿Cuántas veces es necesario evaluar f para saber eso?

Solución clásica

Se ve trivialmente que hay que averiguar tanto $f(0)$ como $f(1)$ y, por lo tanto, es necesario evaluar la función dos veces para determinar si la función es constante.

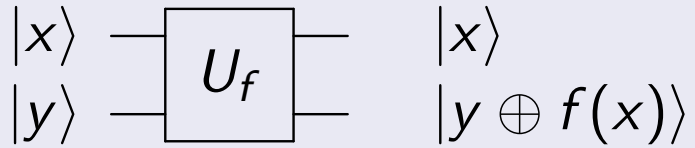
Ventaja cuántica

Evaluando la función f una sola vez podemos determinar si es constante o no.

El algoritmo de Deutsch

Evaluando f

Nos dan una unitaria U_f que codifica f y actúa sobre dos qubits como:



donde $\oplus$ es la suma módulo 2.

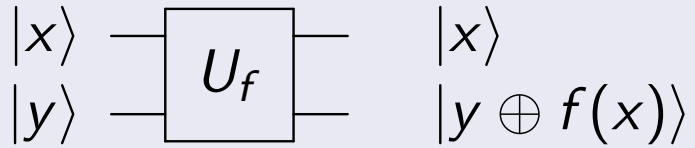
Nota

Eso sirve también para evaluar la función clásicamente.

El algoritmo de Deutsch

Evaluando f

Nos dan una unitaria U_f que codifica f y actúa sobre dos qubits como:

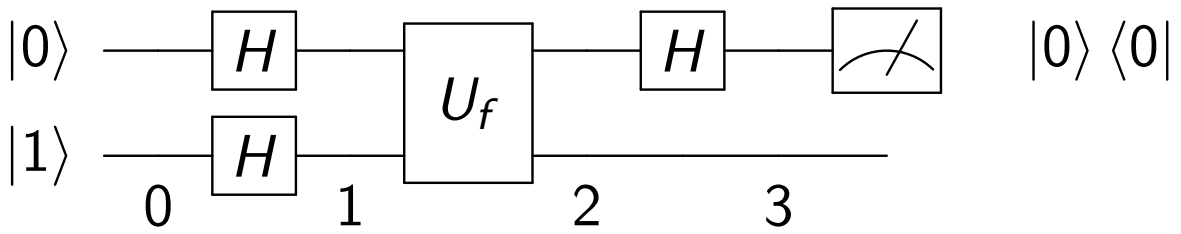


donde $\oplus$ es la suma módulo 2.

Nota

Eso sirve también para evaluar la función clásicamente.

El algoritmo de Deutsch



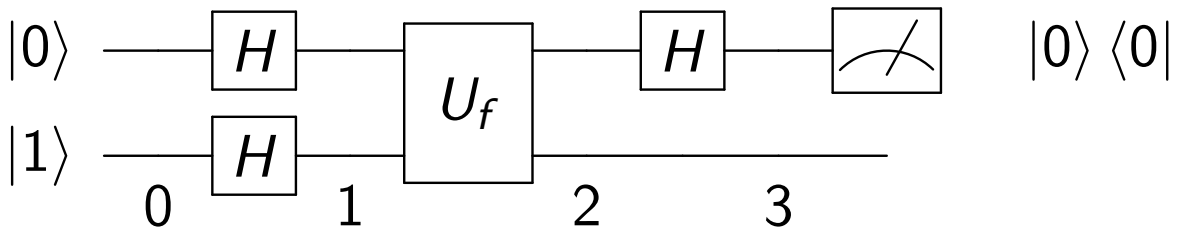
Análisis

$$|\psi_0\rangle = |0\rangle\,|1\rangle$$

$$|\psi_1\rangle = \frac{1}{2}\left(|0\rangle + |1\rangle\right)\left(|0\rangle - |1\rangle\right) = \frac{1}{2}\left(|00\rangle - |01\rangle + |10\rangle - |11\rangle\right)$$

$$|\psi_2\rangle = \frac{1}{2}\left(|0\rangle\,|f(0)\rangle - |0\rangle\,|\neg f(0)\rangle + |1\rangle\,|f(1)\rangle - |1\rangle\,|\neg f(1)\rangle\right)$$

El algoritmo de Deutsch



Análisis

$$|\psi_0\rangle = |0\rangle\,|1\rangle$$

$$|\psi_1\rangle = \frac{1}{2}\left(|0\rangle + |1\rangle\right)\left(|0\rangle - |1\rangle\right) = \frac{1}{2}\left(|00\rangle - |01\rangle + |10\rangle - |11\rangle\right)$$

$$|\psi_2\rangle = \frac{1}{2}\left(|0\rangle\,|f(0)\rangle - |0\rangle\,|\neg f(0)\rangle + |1\rangle\,|f(1)\rangle - |1\rangle\,|\neg f(1)\rangle\right)$$

El algoritmo de Deutsch

Análisis

$$|\psi_2\rangle = \frac{1}{2}\left(|0\rangle\,|f(0)\rangle - |0\rangle\,|\neg f(0)\rangle + |1\rangle\,|f(1)\rangle - |1\rangle\,|\neg f(1)\rangle\right)$$

Si $f(0) = f(1) = c$:

$$|\psi_2^{=}\rangle = \frac{1}{2}\left((|0\rangle + |1\rangle)\,|c\rangle - (|0\rangle + |1\rangle)\,|\neg c\rangle\right) =$$

$$= \frac{1}{2}\left(|0\rangle + |1\rangle\right)\left(|c\rangle - |\neg c\rangle\right)$$

Si $f(0) = c = \neg f(1)$:

$$|\psi_2^{\neq}\rangle = \frac{1}{2}\left(|0\rangle\,|c\rangle - |0\rangle\,|\neg c\rangle + |1\rangle\,|\neg c\rangle - |1\rangle\,|c\rangle\right) =$$

$$= \frac{1}{2}\left(|0\rangle - |1\rangle\right)\left(|c\rangle - |\neg c\rangle\right)$$

El algoritmo de Deutsch

Análisis

$$|\psi_3^{=}\rangle = |0\rangle \frac{1}{\sqrt{2}} (|c\rangle - |\neg c\rangle)$$

$$|\psi_3^{\neq}\rangle = |1\rangle \frac{1}{\sqrt{2}} (|c\rangle - |\neg c\rangle)$$

Midiendo el primer registro determino si la función es constante o no.

Tarea

- Jueguen un poco con IBM Quantum Experience. Pueden crear circuitos y mandarlos a un procesador cuántico.
- Instalen o usen online Quirk, un simulador de circuitos cuánticos.

Hoy vimos

- Evolución temporal.
- Modelo de circuitos.
- Algoritmo de Deutsch.

Hoy vimos

- Evolución temporal.
- Modelo de circuitos.
- Algoritmo de Deutsch.

Hoy vimos

- Evolución temporal.
- Modelo de circuitos.
- Algoritmo de Deutsch.