Primeros algoritmos cuánticos

Clase nº 9 de Computación Cuántica

J. P. Paz, C. Cormick

Depto. de Física, FCEyN, UBA

1er cuatrimestre 2006

En la novena clase se vieron los siguientes algoritmos cuánticos:

- Deutsch-Josza
- Bernstein-Vazirani
- Simon

Las compuertas que se usan

- Los tres algoritmos usan una compuerta U_f que "evalúa" una función f, y la transformada de Hadamard $H^{\otimes n}$.
- Dada $f : \{0,1\}^n \to \{0,1\}^m$, se define U_f como el operador que mapea:

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

(x tiene n bits, y tiene m, $\oplus$ es la suma mod 2 bit a bit).

- La transformada de Hadamard aplica una compuerta H sobre cada qubit:

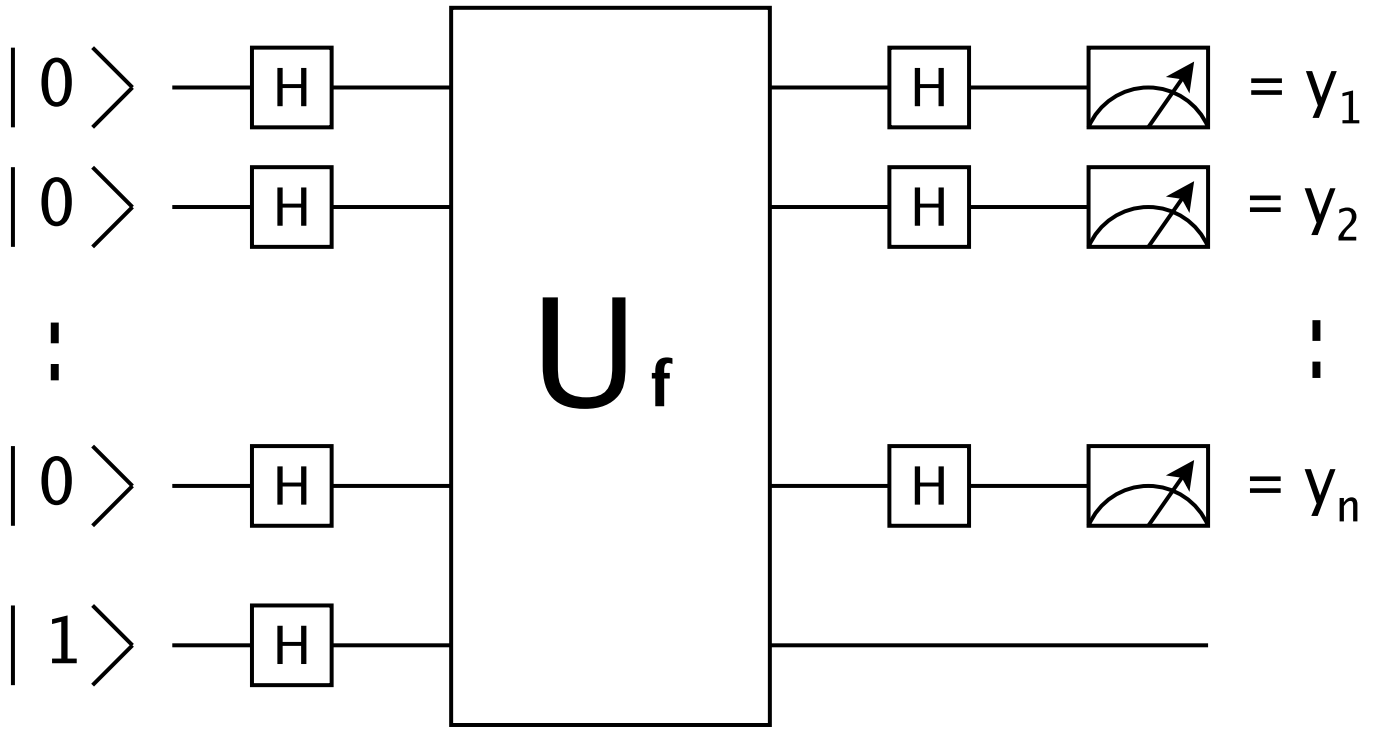
$$H^{\otimes n}|x\rangle = \sum_y (-1)^{x \cdot y}|y\rangle$$

donde $x \cdot y$ es el producto escalar bit a bit.

El algoritmo de Deutsch-Josza

- Es el primer problema que muestra una "superioridad" de las computadoras cuánticas sobre las clásicas.
- Se tiene $f : \{0,1\}^n \longrightarrow \{0,1\}$, con la promesa de que f es constante o balanceada.
  El objetivo es averiguar, con la menor cantidad posible de llamadas a la función, a cuál de los dos casos corresponde.
- Clásicamente es necesario evaluar la función, en el peor caso, un número de veces exponencial en n.
- Cuánticamente basta con una sola vez.

El algoritmo de Deutsch-Josza



El algoritmo de Deutsch-Josza

- $|\psi(t_1)\rangle = (H^{\otimes n}|0\rangle)\ (H|1\rangle) =$

  $= \sum_x (1/\sqrt{2^n})\ |x\rangle\ (1/\sqrt{2})\ (|0\rangle - |1\rangle)$

- $|\psi(t_2)\rangle = U_f\ |\psi(t_1)\rangle =$

  $= \sum_x (1/\sqrt{2^n})\ |x\rangle\ (1/\sqrt{2})\ (|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle) =$

  $= \sum_x (1/\sqrt{2^n})\ (-1)^{f(x)}|x\rangle\ (1/\sqrt{2})\ (|0\rangle - |1\rangle)$

- $|\psi(t_3)\rangle = (H^{\otimes n} \otimes \hat{I})\ |\psi(t_2)\rangle =$

  $= \sum_{x,y} (1/2^n)\ (-1)^{f(x)+x\cdot y}|y\rangle\ (1/\sqrt{2})\ (|0\rangle - |1\rangle)$

- $Prob(y = 0) = \left|(1/2^n)\sum_x (-1)^{f(x)}\right|^2 = \begin{cases} 1 & \text{si } f \text{ es cte.} \\ 0 & \text{si } f \text{ es bal.} \end{cases}$

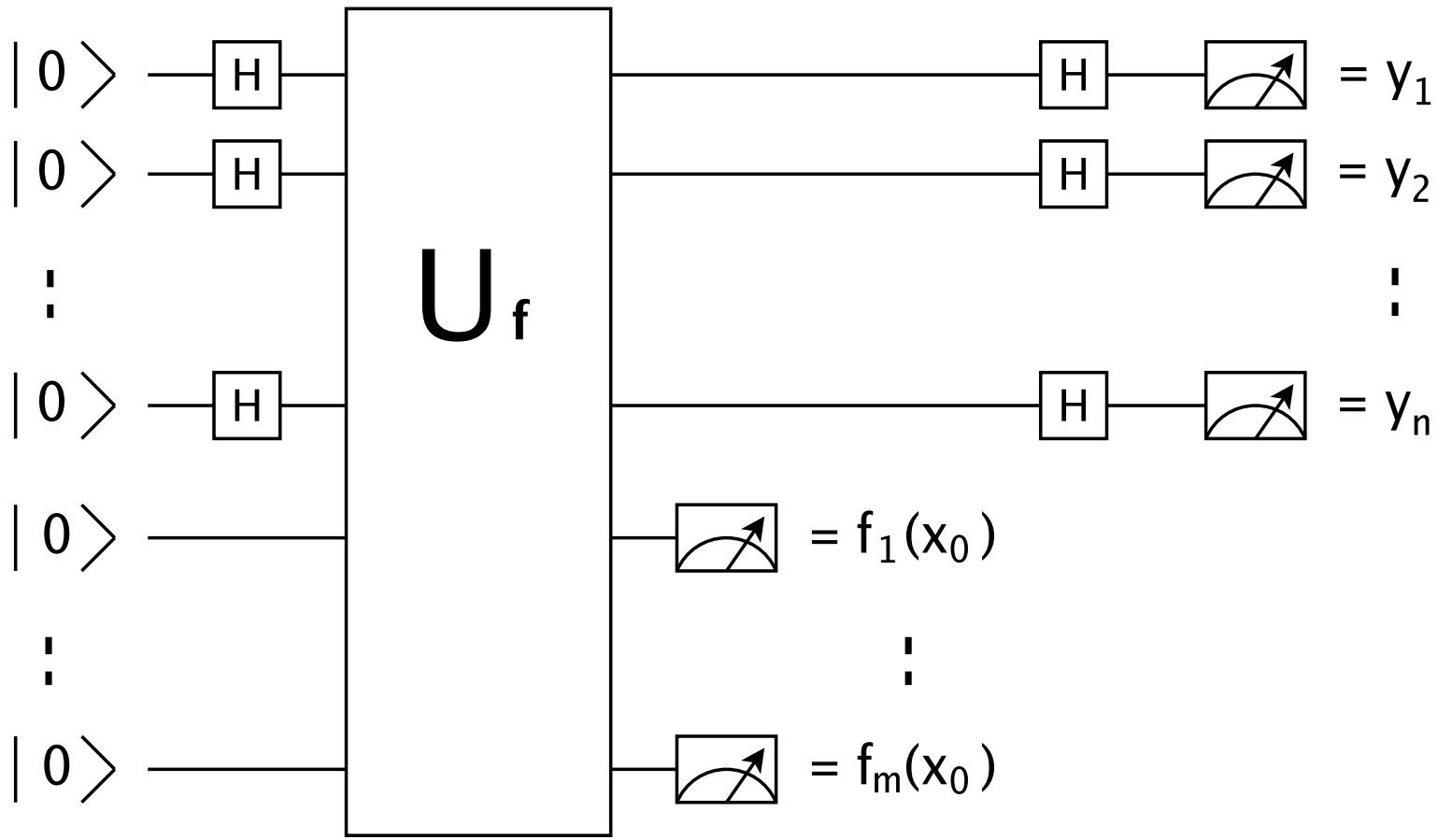
El algoritmo de Bernstein-Vazirani

- Dada una n-upla binaria a, se define $f : \{0,1\}^n \to \{0,1\}$ como $f(x) = x \cdot a$ (producto escalar mod 2).
  El objetivo es hallar a (con la menor cantidad posible de llamadas a la función).
- Clásicamente es necesario evaluar la función n veces.
- Cuánticamente basta con una vez: usando el mismo algoritmo que en el problema de Deutsch-Josza (pero redefiniendo U_f según la nueva f), en la medición final resulta:
  $Prob(y) = \left|(1/2^n)\sum_x (-1)^{x\cdot(a\oplus y)}\right|^2 = \begin{cases} 1 & y = a \\ 0 & y \neq a \end{cases}$

El algoritmo de Simon

- Se tiene $f : \{0,1\}^n \to \{0,1\}^m$ tal que $f(x) = f(x')$ sii $x' = x \oplus a$ (suma bit a bit mod 2). El objetivo es hallar a.
- Clásicamente es necesario evaluar la función un número promedio de veces exponencial en n.
- El algoritmo de Simon es un algoritmo cuántico probabilístico, que resuelve este problema evaluando f un número de veces de orden n.
- Para esto se usa una generalización del circuito de Deutsch-Josza. En la medición final se obtienen siempre resultados y que satisfacen $y \cdot a = 0 \pmod 2$; se repiten los pasos hasta obtener n resultados L.I. y se despeja a.

El algoritmo de Simon



El algoritmo de Simon

- $|\psi(t_1)\rangle = (H^{\otimes n}|0\rangle)\ |0\rangle = \sum_x (1/\sqrt{2^n})\ |x\rangle\ |0\rangle$
- $|\psi(t_2)\rangle = U_f\ |\psi(t_1)\rangle = \sum_x (1/\sqrt{2^n})\ |x\rangle\ |f(x)\rangle$
- Si en la medición se obtiene $f(x_0)$, el estado luego es:
  $|\psi(t_3)\rangle = (1/\sqrt{2})\ (|x_0\rangle + |x_0 \oplus a\rangle)\ |f(x_0)\rangle$
- $|\psi(t_4)\rangle = (H^{\otimes n} \otimes \hat{I})\ |\psi(t_3)\rangle =$
  $= \sum_y (1/\sqrt{2^{n+1}})\ [(-1)^{x_0 \cdot y} + (-1)^{(x_0 \oplus a)\cdot y}]\ |y\rangle\ |f(x_0)\rangle$
- $Prob(y) = \left|(1/\sqrt{2^{n+1}})\ (-1)^{x_0 \cdot y}\ [1 + (-1)^{a \cdot y}]\right|^2 =$
  $= (1/2^{n+1})\ |1 + (-1)^{a \cdot y}|^2 = \begin{cases} 1/2^{n-1} & \text{si } a \cdot y = 0 \\ 0 & \text{si } a \cdot y \neq 0 \end{cases}$

Complejidad

- Evaluamos la complejidad de estos algoritmos como el número de llamadas al "oráculo" que evalúa la función f.

problema	alg. clás. con certeza	alg. clás. con error ac.	alg. cuánt. con certeza	alg. cuánt. con error ac.
Deutsch -Josza	Orden 2^n	Orden n con P(err) $< 2^{-n}$	1	-
Bernstein -Vazirani	n	-	1	-
Simon	Orden 2^n	Orden 2^n	Orden 2^n	Orden n