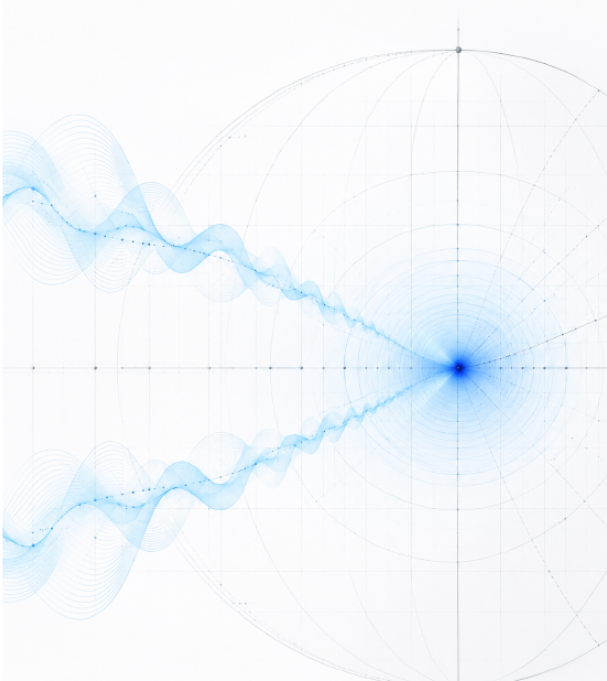
LEZIONE 20

L’algoritmo di Bernstein–Vazirani

Ricostruire una stringa binaria nascosta

Metodi della Fisica Quantistica per l’Ingegneria

Prof. Vincenzo Berardi · Politecnico di Bari



Da Deutsch–Jozsa a Bernstein–Vazirani

Nella Lect19 abbiamo distinto, con una sola query quantistica, una funzione costante da una bilanciata. Ora vogliamo ricostruire una **stringa binaria sconosciuta** di n bit, non soltanto classificare la funzione.

$$\boxed{s = s_1 s_2 \cdots s_n, \qquad s_j \in \{0,1\}.}$$

Il circuito quantistico rimane quello di Deutsch–Jozsa: cambia la funzione implementata dall'oracle, e cambia l'informazione che ricaviamo dalla misura.

Bit classici e stati quantistici: due oggetti diversi

Le stringhe $s = s_1 \cdots s_n$ e $x = x_1 \cdots x_n$ sono entrambe **classiche**:

- s è il parametro **sconosciuto** che vogliamo identificare;
- x è l'**ingresso** che possiamo scegliere per interrogare la funzione.

Ciascuna stringa classica identifica uno stato della **base computazionale** di n qubit:

$$\boxed{x = x_1 \cdots x_n \iff |x\rangle = |x_1\rangle \otimes \cdots \otimes |x_n\rangle \,.}$$

La stessa corrispondenza vale per s. Ad esempio, $s = 101$ identifica $|s\rangle = |1\rangle \otimes |0\rangle \otimes |1\rangle = |101\rangle$. Alla fine dell'algoritmo il registro si trova in $|s\rangle$: la sua misura restituisce la stringa classica s.

La funzione che nasconde la stringa s

Ipotesi del problema di Bernstein–Vazirani: assumiamo che la funzione implementata dall'oracle appartenga alla famiglia

$$\boxed{f_s(x) = s \cdot x = \left(\sum_{j=1}^{n} s_j x_j \right) \bmod 2.}$$

La **forma matematica** della funzione è nota, mentre la stringa $s = s_1 \cdots s_n$ è sconosciuta e determina quale funzione della famiglia sia presente. L'oracle restituisce $f_s(x) \in \{0, 1\}$ per un ingresso scelto x. Ogni coefficiente s_j indica se il bit x_j contribuisce alla somma: $s_j = 0$ lo esclude; $s_j = 1$ lo include.

Perché la somma modulo 2 si scrive con XOR?

Per due bit $a, b \in \{0, 1\}$, l'operazione XOR coincide con la somma modulo 2:

$$\boxed{a \oplus b = (a + b) \bmod 2.}$$

a	b	$(a + b) \bmod 2$	$a \oplus b$
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

Quindi la funzione può essere scritta anche come

$$f_s(x) = s_1 x_1 \oplus s_2 x_2 \oplus \cdots \oplus s_n x_n.$$

Primo esempio: $s = 101$

Per $n = 3$ e $s = 101$ otteniamo

$$\begin{aligned} f_{101}(x_1x_2x_3) &= (1 \cdot x_1 + 0 \cdot x_2 + 1 \cdot x_3) \bmod 2 \\ &= (x_1 + x_3) \bmod 2 = x_1 \oplus x_3. \end{aligned}$$

x	000	100	010	001	110	101
$f_{101}(x)$	0	1	0	1	1	0

Il secondo ingresso non influisce sull'uscita, perché $s_2 = 0$.

Secondo esempio: $s = 011$

Se la stringa nascosta è $s = 011$, allora

$$f_{011}(x_1x_2x_3) = (0 \cdot x_1 + 1 \cdot x_2 + 1 \cdot x_3) \bmod 2 = x_2 \oplus x_3.$$

x	000	100	010	001	011	111
$f_{011}(x)$	0	0	1	1	0	0

In questo caso è il **primo** ingresso a non contribuire; la funzione dipende da x_2 e x_3.

Terzo esempio: $s = 111$

Con $s = 111$, tutti e tre gli ingressi partecipano alla parità:

$$f_{111}(x_1x_2x_3) = (x_1 + x_2 + x_3) \bmod 2 = x_1 \oplus x_2 \oplus x_3.$$

x	000	001	010	011	100	101	110	111
$f_{111}(x)$	0	1	1	0	1	0	0	1

Due contributi uguali a 1 producono 0 modulo 2, mentre tre contributi uguali a 1 producono 1.

Come interrogare la funzione per trovare s?

Supponiamo che s sia ignota e $n = 3$. Interroghiamo la funzione con ingressi che contengono **un solo bit uguale a** 1:

$$\boxed{f_s(100) = s_1, \qquad f_s(010) = s_2, \qquad f_s(001) = s_3.}$$

Se le risposte sono rispettivamente $1, 0, 1$, possiamo ricostruire

$$\boxed{s = 101.}$$

Questo è un procedimento **classico**: ogni interrogazione fornisce un bit della stringa sconosciuta.

Generalizzazione: n interrogazioni classiche

Per n bit, indichiamo con e_j l'ingresso che ha 1 nella posizione j e 0 nelle altre.

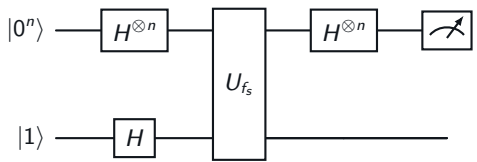
$$e_1 = 10\cdots0, \qquad e_2 = 01\cdots0, \qquad e_n = 00\cdots1.$$

Poiché soltanto il termine $s_j x_j$ è diverso da zero,

$$\boxed{f_s(e_j) = s \cdot e_j = s_j.}$$

Ripetendo la procedura per $j = 1, \ldots, n$, recuperiamo tutti i bit di s in esattamente n query.

Il circuito: vista d'insieme



La linea superiore rappresenta un registro di n qubit; la misura finale riguarda soltanto tale registro. La sequenza è quella della Lect19: la sola sostituzione è $U_f \mapsto U_{f_s}$.

La sola novità: la fase dipende linearmente da s

Riprendiamo il circuito della Lect19, con l'ancilla in $|-\rangle$. La stessa regola di *phase kickback* ora fornisce, poiché $f_s(x) = s \cdot x$,

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{s \cdot x} |x\rangle |-\rangle .$$

La differenza rispetto a Deutsch–Jozsa è la **struttura lineare delle fasi**. Per decodificarle usiamo, senza ridimostrarla, la relazione della Lect19:

$$H^{\otimes n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y} |y\rangle .$$

Dimostrazione: le Hadamard finali restituiscono $|s\rangle$

$$|\psi_3\rangle = \frac{1}{2^n} \sum_{y\in\{0,1\}^n} \left[\sum_{x\in\{0,1\}^n} (-1)^{(s\oplus y)\cdot x} \right] |y\rangle\,|-\rangle = \sum_{y\in\{0,1\}^n} A_y\,|y\rangle\,|-\rangle\,,$$

$$A_y = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{(s\oplus y)\cdot x}.$$

Se $y = s$, tutti i termini valgono 1 e $A_s = 1$. Se $y \neq s$, esiste un indice j con $(s \oplus y)_j = 1$: i termini per x e $x \oplus e_j$ hanno segno opposto, e $A_y = 0$.

$$\boxed{A_y = \delta_{ys} \quad\Longrightarrow\quad |\psi_3\rangle = |s\rangle\,|-\rangle\,, \qquad P(s) = 1.}$$

Una lettura alternativa: fattorizzazione delle fasi

Dal momento che $s \cdot x = \left(\sum_j s_j x_j\right) \bmod 2$, il registro dopo l'oracle soddisfa

$$\frac{1}{\sqrt{2^n}} \sum_x (-1)^{s\cdot x} |x\rangle = \bigotimes_{j=1}^{n} \frac{|0\rangle + (-1)^{s_j} |1\rangle}{\sqrt{2}}$$

$$= \bigotimes_{j=1}^{n} \begin{cases} |+\rangle, & s_j = 0, \\ |-\rangle, & s_j = 1. \end{cases}$$

Il bit s_j è dunque codificato nel segno relativo tra $|0\rangle$ e $|1\rangle$ del qubit j.

Esempio $s = 101$: quali termini cambiano segno?

x	000	001	010	011	100	101	110	111
$f_s(x)$	0	1	0	1	1	0	1	0
$(-1)^{f_s(x)}$	$+$	$-$	$+$	$-$	$-$	$+$	$-$	$+$

Ne segue lo stato del registro di ingresso

$$\frac{1}{\sqrt{8}}\big(|000\rangle - |001\rangle + |010\rangle - |011\rangle - |100\rangle + |101\rangle - |110\rangle + |111\rangle\big).$$

L'ancilla rimane nello stato $|-\rangle$ ed è omessa dalla formula per brevità.

Esempio $s = 101$: fattorizzazione dello stato

Il registro di ingresso dopo l'oracle può essere scritto come

$$|\phi_2\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$
$$= |-\rangle\,|+\rangle\,|-\rangle\,.$$

Il primo e il terzo qubit sono in $|-\rangle$ perché $s_1 = s_3 = 1$; il secondo è in $|+\rangle$ perché $s_2 = 0$.

Esempio $s = 101$: misura finale

Le tre Hadamard finali trasformano

$$H^{\otimes 3} |-\rangle |+\rangle |-\rangle = (H|-\rangle) \otimes (H|+\rangle) \otimes (H|-\rangle)$$
$$= |1\rangle \otimes |0\rangle \otimes |1\rangle = |101\rangle .$$

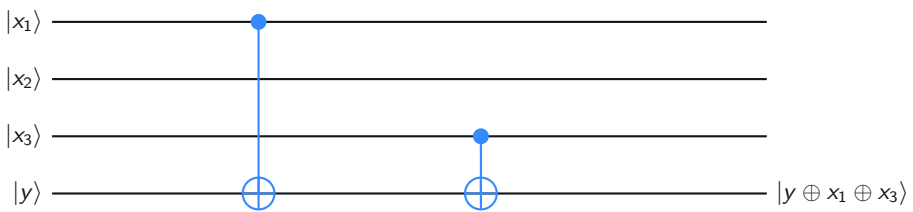
La misura restituisce con probabilità uno

$$\boxed{101 = s.}$$

Per ottenere il risultato, il circuito ha interrogato l'oracle una volta sola.

Circuito esplicito dell'oracle per $s = 101$

Poiché $f_s(x) = x_1 \oplus x_3$, l'oracle può essere costruito con due porte CNOT, entrambe dirette all'ancilla.



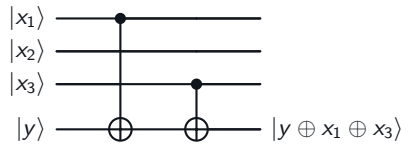
Le CNOT non coinvolgono x_2: ciò riflette il fatto che $s_2 = 0$.

Come si costruisce l'oracle per $s = 101$?

L'oracle deve eseguire

$$U_{f_s} |x\rangle |y\rangle = |x\rangle |y \oplus f_s(x)\rangle .$$

Per $s = 101$, abbiamo $f_s(x) = x_1 \oplus x_3$: bastano due CNOT con la stessa ancilla come bersaglio.



$$y \xrightarrow{\text{CNOT}(x_1)} y \oplus x_1 \xrightarrow{\text{CNOT}(x_3)} y \oplus x_1 \oplus x_3.$$

Come si costruisce l'oracle per una stringa qualsiasi?

Per $s = s_1 \cdots s_n$, inseriamo una **CNOT** controllata da x_j e diretta all'ancilla **solo quando** $s_j = 1$.

$$\boxed{y \longmapsto y \oplus s_1 x_1 \oplus \cdots \oplus s_n x_n = y \oplus f_s(x).}$$

Stringa s	Qubit di controllo delle CNOT
100	x_1
101	x_1, x_3
011	x_2, x_3
111	x_1, x_2, x_3

Il numero di CNOT coincide con il numero di bit uguali a 1 in s. Una singola *query* all'oracle non equivale necessariamente a una sola porta fisica.

Complessità: confronto classico e quantistico

Per algoritmi **esatti**, nel modello a oracle, si ottiene

$$\boxed{Q_{\mathrm{C}}^{\mathrm{det,exact}}(n) = n, \qquad Q_{\mathrm{Q}}^{\mathrm{exact}}(n) = 1.}$$

Procedura	Query	Risultato
Classica deterministica	n	n bit di s, esatti
Quantistica	1	n bit di s, esatti

Si tratta di una separazione **lineare** in numero di query, non esponenziale.

Perché una misura restituisce n bit?

Il registro finale è precisamente uno stato della base computazionale:

$$|\psi_3\rangle = |s_1 s_2 \cdots s_n\rangle\,|-\rangle\,.$$

La misura del registro di ingresso produce una stringa classica di n bit, cioè s, con probabilità uno.

Il procedimento non estrae tutti i valori di f_s dalla sovrapposizione: **l'interferenza concentra tutta la probabilità nell'unico risultato desiderato**.

Questo è possibile grazie alla struttura lineare promessa per f_s.

Deutsch–Jozsa e Bernstein–Vazirani: stesso circuito

L'**unica differenza circuitale essenziale** è la funzione implementata da U_f; cambia, di conseguenza, il problema che la misura risolve.

	Deutsch–Jozsa	Bernstein–Vazirani
Oracle	f costante o bilanciata	$f_s(x) = s \cdot x$
Domanda	Quale delle due classi?	Qual è la stringa s?
Misura	0^n se costante; altro altrimenti	La stringa s
Query	1	1

Se $s = 0^n$, f_s è costante; se $s \neq 0^n$, f_s è bilanciata. Bernstein–Vazirani legge l'*intero risultato* che Deutsch–Jozsa usa soltanto per decidere la classe.

Un'applicazione: identificare gli ingressi di una parità

Un circuito digitale con cinque ingressi produce

$$f(x_1, x_2, x_3, x_4, x_5) = x_1 \oplus x_3 \oplus x_5.$$

Il sottoinsieme di ingressi attivi è codificato dalla stringa

$$\boxed{s = 10101.}$$

- Una procedura classica deterministica identifica tutti e cinque i coefficienti con cinque query.
- Con accesso coerente all'oracle, Bernstein–Vazirani identifica s in una query ideale.

Conclusioni

1. La promessa $f_s(x) = s \cdot x \pmod 2$ codifica una stringa ignota di n bit.
2. Una query quantistica trasforma questa informazione in segni delle ampiezze.
3. Le Hadamard finali cancellano tutte le ampiezze tranne quella di $|s\rangle$.
4. La misura del registro restituisce l'intera stringa s con probabilità uno.

$$|0^n\rangle\,|1\rangle \xrightarrow{H^{\otimes(n+1)},\ U_{f_s},\ H^{\otimes n}} |s\rangle\,|-\rangle\,, \qquad Q_{\mathrm{Q}} = 1.$$