# Quantum Computing

Fourier Sampling, Algoritmo di Simon e Base di Fourier

## 1 Parallelismo quantistico

Un passo fondamentale di quasi tutti gli algoritmi quantistici consiste nel calcolare una data funzione $f$ implementata mediante un circuito quantistico $U_f$ su una sovrapposizione di tutti i valori dell'input. In questo modo è possibile ottenere tutti i possibili valori di $f(x)$ con una sola applicazione di $U_f$. Questo effetto si chiama *parallelismo quantistico*. Il circuito $U_f$, detto *oracolo*, viene di solito visto come una scatola nera il cui funzionamento interno non si conosce, ma che si può invocare per ottenere un effetto noto. In generale, la procedura si può schematizzare nei seguenti passaggi. Supponiamo che $f$ sia una funzione su $n$ bits che produce un valore su $k$ bits, cioè $f : \{0,1\}^n \mapsto \{0,1\}^k$. Allora si prepara in input un registro di $n$ qubits $|00\ldots0\rangle$ e si applica la Trasformata di Walsh-Hadamard ottenendo la sovrapposizione equiprobabile

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

di tutti i vettori della base computazionale per lo spazio degli stati $\mathbb{C}^{2^n}$. L'input per il circuito $U_f$ è costituito da questa sovrapposizione di tutti gli input per $f$ e un registro di $k$ qubits $|0\rangle$ destinato a contenere il risultato. Si ottiene quindi per linearità

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

Misurare a questo punto non permette tuttavia di sfruttare il parallelismo quantistico, in quanto si otterrebbe un solo risultato e per di più random

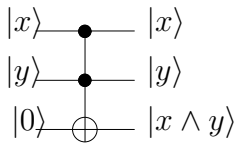


Figure 1: Circuito che realizza l'and di $x$ e $y$

(non è possibile stabilire a priori quale risultato sarà ottenuto per effetto della misurazione). Per trarre pieno vantaggio dal parallelismo quantistico si sfruttano delle tecniche particolari. Abbiamo già visto un esempio nell'algoritmo di Deutsch-Jozsa, dove una singola invocazione dell'oracolo $U_f$ permetteva di stabilire con probabilità 1 se la funzione $f$ era costante o bilanciata.

**Nota 1.1** *Poiché $n$ qubit permettono di lavorare simultaneamente su $2^n$ stati, il parallelismo quantistico si differenzia dal parallelismo classico che deve sempre raggiungere un compromesso tra spazio e tempo impiegati. Nel caso quantistico è invece possibile sfruttare una quantità esponenziale di spazio computazionale in uno spazio fisico di dimensioni lineari.*

**Esercizio 1.2 (`AND` logico)** *Considera il circuito di Toffoli per calcolare l'*`AND` *reversibile di due bits (v. Figura 1). Cosa si ottiene applicando il circuito alla sovrapposizione di tutti i possibili inputs $|x\rangle$ e $|y\rangle$?*

La seguente procedura è nota come *Fourier Sampling*: Si prepara un registro di $n$ qubit in uno stato $|\phi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$; si calcola $H^{\otimes n}|\phi\rangle$; infine si misura lo stato risultante $\sum_{x\in\{0,1\}^n} \hat{\alpha}_x |x\rangle$ per ottenere il risultato $y$ con probabilità $|\hat{\alpha}_y|^2$.

Il circuito che realizza l'algoritmo di Deutsch-Josza (vedi Appunti delle lezioni) implementa questa procedura per produrre lo stato $|\phi\rangle = \frac{1}{\sqrt{2^n}} \sum_x (-1)^{f(x)} |x\rangle$ e poi misurare $H^{\otimes n}|\phi\rangle$, ottenendo la soluzione.

**Esercizio 1.3 (Fourier Sampling)** *Applicando il Fourier Sampling allo stato*

$$\frac{1}{2\sqrt{2}}[|000\rangle - |001\rangle + |010\rangle - |011\rangle + |100\rangle - |101\rangle + |110\rangle - |111\rangle],$$

*quali sono i possibili risultati della misurazione?*

**Esercizio 1.4 (Algoritmo di Simon)** *Un altro esempio di utilizzo del parallelismo quantistico in associazione con il Fourier sampling per testare le proprietà di una data trasformazione unitaria $U_f$ è l'algoritmo di Simon per risolvere il seguente problema.*

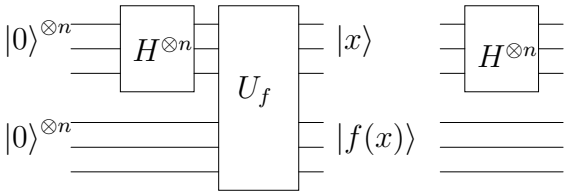


Figure 2: Circuito di Simon

**Problema di Simon** *Supponiamo di disporre di una scatola nera $U_f$ che calcola una funzione $f : \{0,1\}^n \mapsto \{0,1\}^n$ con la proprietà che per una data stringa $s \in \{0,1\}^n$ si ha che per ogni $x \neq y$,*

$$f(x) = f(y) \text{ se e solo se } x \oplus y = s.$$

*L'algoritmo permette di calcolare $s$ usando il circuito (simile a quello di Deutsch-Jozsa) in Figura 2.*

*Verificare che l'output del circuito è $y$ tale che $y \cdot s = 0$ e che $y$ è uniformemente distribuito su tutti i possibili outputs.*

**Dimostrazione** Dopo l'applicazione della trasformata di Hadamard si ottiene lo stato

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

e dopo l'invocazione all'oracolo $U_f$

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

A questo punto misuriamo il secondo registro ottenendo un risultato $|f(z)\rangle$. Poiché i soli due valori dell'input su cui $f$ assume il valore $f(z)$ sono $z$ e $z \oplus s$, il primo registro collasserà dopo la misurazione nello stato

$$\frac{1}{\sqrt{2}}[|z\rangle + |z \oplus s\rangle].$$

Misurare questo registro non ci darebbe alcuna informazione utile. Applichiamo invece nuovamente la trasformata di Hadamard, ottenendo

$$\begin{array}{l} \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} \frac{(-1)^{z\cdot y}}{\sqrt{2}} |y\rangle + \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} \frac{(-1)^{(z+s)\cdot y}}{\sqrt{2}} |y\rangle = \\ \frac{1}{\sqrt{2^{n+1}}} \sum_{y=0}^{2^n-1} [(-1)^{z\cdot y} + (-1)^{z\cdot y}(-1)^{s\cdot y}] |y\rangle = \\ \frac{1}{\sqrt{2^{n+1}}} \sum_{y=0}^{2^n-1} (-1)^{z\cdot y}[1 + (-1)^{s\cdot y}] |y\rangle . \end{array}$$

Quindi, se $s \cdot y = 1$ l'ampiezza di probabilità di $|y\rangle$ è 0, mentre tutti gli $y$ tali che $r \cdot y = 0$ hanno la stessa ampiezza con segni differenti. Pertanto il risultato di una misurazione darà a questo punto uno degli $y$ tali che $s \cdot y = 0$ con la stessa probabilità.

L'algoritmo di Simon consiste nell'applicare questo circuito $n-1$ volte per ottenere $n-1$ vettori $y^1, y^2, \ldots y^{(n-1)}$. Se questi $y^i$ sono linearmente indipendenti, allora risolvendo il sistema lineare $y^i \cdot s = 0$ si ottiene $s$. Altrimenti, si applica il circuito ancora una volta fino a quando non si ottengono $n-1$ vettori linearmente indipendenti.

**Esercizio 1.5** *Verificare che il circuito per l'algoritmo di Simon permette di trovare $n-1$ vettori tali che $s \cdot y = 0$ e linearmente indipendenti con probabilità costante e pari a $\frac{1}{4}$.*

**Esercizio 1.6** *Supponiamo di voler risolvere il problema di Simon $n = 4$. Se eseguendo 4 volte l'algoritmo di Simon abbiamo ottenuto 0000, 1101, 1010 e 0110, qual' la stringa $s$? (Si assuma che $s$ sia diversa dalla stringa 0000).*

**Esercizio 1.7** *Applicare l'algoritmo di Simon alla funzione $f(x) : \{0,1\}^2 \to \{0,1\}^2$, definita da:*

$$\begin{aligned} f(00) = f(01) &= 00 \\ f(01) = f(11) &= 01. \end{aligned}$$

**Esercizio 1.8 (Base di Fourier)** *La base computazionale per un registro di $n$ qubit è formata dai vettori unitari in $\mathbb{C}^{2^n}$ corrispondenti a ciascuna configurazione classica del sistema e indicati mediante le rappresentazioni binarie (su $n$ bits) dei numeri da 0 a $2^n - 1$.*

*Un'altra base importante per lo spazio degli stati generati da $n$ qubit è la* base di Fourier*. I vettori di questa base sono etichettati dagli elementi in $\{0,1\}^n$. Quindi per ogni $u \in \{0,1\}^n$:*

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

*dove $x \cdot u = \sum_j x_j u_j$. Nota che una misurazione in questa base fa ottenere ciascun $x$ con la stessa probabilità $1/2^n$.*

1. *Verificare che i vettori della base di Fourier sono ortonormali.*

2. *Costruire un circuito che trasforma ciascun vettore della base computazionale $|u\rangle$ nel vettore della base di Fourier $|\chi_u\rangle$.*