Quantum Computing

Alessandra Di Pierro

Appunti del corso - A.A. 2001-2002

1 Esercitazione 3: 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 é 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 puó invocare per ottenere un effetto noto. In generale, la procedura si puó schematizzare nei seguenti passaggi. Supponiamo che f sia una funzione su n bits che produce un valore su k bits, cioé $f : \{0,1\}^n \mapsto \{0,1\}^k$. Allora si prepara in input un registro di n qubits $|00\dots 0\rangle$ 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 é 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 linearitá

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

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

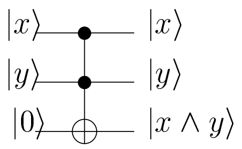

Figure 1: Circuito che realizza l'and di x e y

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

Nota 1.1 *Poiché n qubits 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 é invece possibile sfruttare una quantitá 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 $|y\rangle$?*

Esercizio 1.3 (Algoritmo di Simon) *Un altro esempio di utilizzo del parallelismo quantistico per testare le proprietá di una data trasformazione unitaria U_f é l'algoritmo di Simon per trovare il periodo di una funzione. In questo caso U_f calcola la funzione $f : \{0,1\}^n \mapsto \{0,1\}^n$ tale che*

$$f(x) = f(x + r), \text{ per ogni } x \in \{0,1\}^n.$$

L'algoritmo permette di calcolare r usando un circuito simile a quello di Deutsch-Jozsa e mostrato in Figura 2.

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

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

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

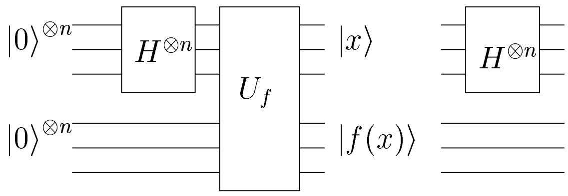

Figure 2: Circuito di Simon

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 $z+r$, il primo registro collasserá dopo la misurazione allo stato

$$\frac{1}{\sqrt{2^n}} |z\rangle + \frac{1}{\sqrt{2^n}} |z+u\rangle .$$

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

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

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

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 $y^i \cdot r = 0$ si ottiene r. Altrimenti, si apllica il circuito ancora una volta fino a quando non si ottengono $n-1$ vettori linearmente indipendenti.

Esercizio 1.4 *Verificare che il circuito per l'algoritmo di Simon permette di trovare $n-1$ vettori tali che $r \cdot y = 0$ e linearmente indipendenti con probabilitá costante e pari a $\frac{1}{4}$.*

Esercizio 1.5 (Base di Fourier) *La base computazionale per un registro di n qubits é 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 qubits é 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 probabilitá $1/2^n$.

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 $|\chi_u\rangle$.*