Quantum Computing

Appunti delle lezioni

Alessandra Di Pierro

Introduzione

1 Fisica e Computazione

Un processo di calcolo è essenzialmente un processo fisico che viene eseguito su una macchina il cui funzionamento obbedisce a certe leggi fisiche. La teoria classica della computazione si basa su un modello astratto di macchina universale (la Macchina di Turing Universale) che funziona secondo un insieme di regole e di principi enunciati nel 1936 da Alan Turing ed elaborati successivamente da John von Neumann negli anni '40. Questi principi sono rimasti essenzialmente immutati da allora, nonostante gli enormi progressi tecnologici che permettono oggi di produrre dispositivi di gran lunga più potenti rispetto a quelli che si potevano realizzare nella prima metà del ventesimo secolo. La tacita assunzione alla base di questi principi è che una macchina di Turing idealizza un dispositivo meccanico di computazione (con una memoria potenzialmente infinita) che obbedisce alle leggi della fisica classica.

La computazione quantistica nasce come un paradigma alternativo basato sui principi della *meccanica quantistica*. Questi sono gli unici in grado di giustificare i fenomeni fisici che avvengono a livello microscopico, come per esempio all'interno di un atomo. Questi fenomeni saranno imprescindibili nella costruzione di computers elettronici in un futuro ormai prossimo se la legge di Moore continuerà a valere, come ci si aspetta. Questa legge formulata già negli anni Sessanta prevedeva che la potenza di un computer sarebbe raddoppiata una volta ogni due anni. In pratica questa legge ha dimostrato fino ad oggi la sua validità, e attualmente effetti quantistici incominciano ad interferire nel funzionamento dei dispositivi elettronici man mano che le loro dimensioni diventano più piccole.

L'idea di realizzare un modello di computazione come un sistema quantistico isolato cominciò ad affacciarsi agli inizi degli anni Ottanta, quando P. Benioff, partendo da considerazioni precedentemente elaborate da C. Bennet, definì la Macchina di Turing reversibile: una computazione può sempre essere eseguita in modo da ritornare allo stato inziale ripercorrendo all'indietro i vari passi di computazione.

Successivamente R. Feynman domostrò che nessuna Macchina di Turing classica poteva simulare certi fenomeni fisici senza incorrere in un rallentamento esponenziale delle sue prestazioni. Al contario, un "simulatore quantistico universale" avrebbe potuto effettuare la simulazione in maniera più efficiente.

Nel 1985 D. Deutsch formalizzò queste idee nella sua Macchina di Turing Quantistica Universale, che rappresenta in teoria della calcolabità quantistica esattamente quello che la Macchina di Turing Universale rappresenta per la calcolabità classica e ha portato alla concezione moderna di *computazione quantistica*.

Naturalmente gli effetti dell'introduzione del nuovo modello di calcolo si sono fatti sentire anche nel campo della complessità computazionale, (come previsto da Feynman), provocando il cambiamento della nozione di "trattabilità". Infatti, nel 1994 P. Shor dimostra che il problema della fattorizzazione dei numeri primi (classicamente considerato intrattabile) si può risolvere efficientemente (cioè in tempo polinomiale) con un algoritmo quantistico.

Queste considerazioni unite a quelle di tipo tecnologico accennate precedentemente, hanno portato all'affermarsi di un campo di ricerca oggi noto come teoria dell'informazione e della computazione quantistica. Concentrandoci su quest'ultima, ne studieremo le differenze fondamentali con il paradigma classico. Queste derivano essenzialmente dai principi della teoria quantistica che regolano il mondo dell'infinitamente piccolo. In particolare, avremo a che fare con i tre fenomeni, tanto fondamentali quanto poco intuitivi, della teoria quantistica su cui la computazione quantistica si basa in maniera essenziale e che determinano la sua enorme potenzialià di calcolo: il *principio di sovrapposizione degli stati*, il *principio di misurazione* e il fenomeno dell'*entanglement*.

2 Quantum bit

Il concetto fondamentale della computazione classica è il *bit*. La computazione quantistica si basa su un concetto analogo, il *quantum bit*, o in

breve *qubit*, di cui descriviamo nel seguito le proprietà fondamentali sottolineandone le differenze con il bit classico.

2.1 Interpretazione matematica

Descriveremo un qubit come un oggetto matematico astratto che gode di certe particolari proprietà. La natura fisica di tale oggetto verrà chiarita successivamente osservando la corrispondenza tra le proprietà di un qubit con quelle di un qualsiasi sistema quantistico a due stati. Richiamiamo innanzitutto alcune definizioni e notazioni alla base del modello matematico del qubit.

2.1.1 Spazi vettoriali

Lo spazio vettoriale reale a due dimensioni $\mathbb{R}^2$ è l'insieme dei vettori colonna della forma

$$v = \begin{pmatrix} a \\ b \end{pmatrix}$$

dove $a, b \in \mathbb{R}$ sono numeri reali. La norma di v è data da $\|v\| = \sqrt{a^2 + b^2}$. Il trasposto di v è il vettore riga $v^T = (a, b)$. Il prodotto scalare di due vettori

$$v_1 = \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} \text{ e } v_2 = \begin{pmatrix} a_2 \\ b_2 \end{pmatrix}$$

è dato da $v_1 \cdot v_2 \stackrel{\text{def}}{=} v_1^T v_2 = (a_1, b_1) \begin{pmatrix} a_2 \\ b_2 \end{pmatrix} = a_1 a_2 + b_1 b_2 = \|v_1\| \|v_2\| \cos\theta$, dove θ è l'angolo tra v_1 e v_2. Se $v_1 \cdot v_2 = 0$, allora i due vettori sono detti ortogonali.

I vettori $\{v_i \in \mathbb{R}^2 \mid i = 1, 2, \ldots, k\}$ sono detti linearmente indipendenti se

$$a_1 v_1 + a_2 v_2 + \ldots + a_k v_k = 0, \ a_i \in \mathbb{R},$$

implica che $a_i = 0$ per ogni $i = 1, 2, \ldots, k$. Altrimenti sono detti linearmente dipendenti.

Una base di $\mathbb{R}^2$ è un qualsiasi insieme di vettori linearmente indipendenti tali che ogni altro vettore in $\mathbb{R}^2$ si può esprimere come combinazione lineare dei vettori nell'insieme. Ogni coppia di vettori v_1 e v_2 linearmente indipendenti forma una base per $\mathbb{R}^2$. Si dice che v_1 e v_2 formano una base ortonormale per $\mathbb{R}^2$ se $\|v_1\| = \|v_2\| = 1$ e $v_1 \cdot v_2 = 0$. I due vettori

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

formano una base ortonormale per $\mathbb{R}^2$ detta la base standard di $\mathbb{R}^2$.

Esercizio 2.1 *Estendere le definizioni e proprietà date per* $\mathbb{R}^2$ *a* $\mathbb{R}^d, d \in \mathbb{N}$.

Analogamente, lo spazio vettoriale complesso a due dimensioni $\mathbb{C}^2$ è l'insieme dei vettori colonna della forma

$$w = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

con $\alpha, \beta \in \mathbb{C}$. La norma di w è data da $\|w\| = \sqrt{|\alpha|^2 + |\beta|^2}$, dove $|z|$ è il modulo del numero complesso z. Il coniugato complesso di w è il vettore riga $w^\dagger = (\alpha^*, \beta^*)$. In analogia con il prodotto scalare di due vettori reali, si definisce il prodotto scalare (o inner product) di due vettori complessi

$$w_1 = \begin{pmatrix} \alpha_1 \\ \beta_1 \end{pmatrix} \; w_2 = \begin{pmatrix} \alpha_2 \\ \beta_2 \end{pmatrix}$$

come

$$(w_1, w_2) \stackrel{\text{def}}{=} w_1^\dagger w_2 = (\alpha_1^*, \beta_1^*) \begin{pmatrix} \alpha_2 \\ \beta_2 \end{pmatrix} = \alpha_1^* \alpha_2 + \beta_1^* \beta_2.$$

Lo spazio vettoriale $\mathbb{C}^2$ con il suo prodotto scalare è detto spazio di Hilbert a due dimensioni.

Le definizioni di indipendenza lineare, base e base ortonormale sono analoghe a quelle per $\mathbb{R}^2$. I due vettori

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \; \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

formano una base ortonormale per $\mathbb{C}^2$ (detta base canonica).

2.2 Notazione di Dirac

Per rappresentare gli elementi di uno spazio vettoriale complesso è conveniente usare una notazione detta *notazione di Dirac* dal nome del famoso fisico inglese, pioniere della teoria quantistica, che la introdusse. Essa rappresenta la notazione standard in meccanica quantistica. Secondo questa notazione $|v\rangle$ (o *ket*) indica un generico elemento dello spazio vettoriale e $|i\rangle$ indica l'i-esimo elemento della base ortonormale canonica.

Se $|v\rangle = \sum_i \alpha_i |i\rangle$ e $|w\rangle = \sum_i \beta_i |i\rangle$, allora il loro prodotto scalare

$$(v, w) = (\alpha_1^*, \ldots, \alpha_d^*) \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_d \end{pmatrix},$$

si indica con $\langle v|\, w\rangle$, e in generale il vettore riga $(\alpha_1^*, \ldots, \alpha_d^*)$ si denota con $\langle v|$ o *bra* (così che $\langle v|\, w\rangle$ formano un *braket*).

L'utilità di questa notazione sarà particolarmente evidente quando studieremo la misurazione quantistica e in particolare gli operatori di proiezione.

2.2.1 Qubit come vettore unitario complesso

Lo stato di un bit classico viene descritto mediante i valori 0 e 1. Il modo piu' diretto per rappresentare lo stato di un qubit è mediante un vettore unitario in uno spazio vettoriale complesso a due dimensioni. I vettori $|0\rangle$ e $|1\rangle$ formano una base ortonormale per questo spazio vettoriale, nota come *base computazionale standard*. Usando la notazione classica dell'algebra lineare, possiamo rappresentare $|0\rangle$ con il vettore colonna $(1,0)^T$ e $|1\rangle$ con il vettore $(0,1)^T$, dove T indica il trasposto. Gli stati $|0\rangle$ e $|1\rangle$ di un qubit si possono vedere come i corrispondenti degli stati 0 e 1 di un bit classico. La differenza tra bits e qubits sta nel fatto che un qubit si può trovare anche in altri stati diversi da $|0\rangle$ e $|1\rangle$. Infatti, ogni combinazione lineare

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,,$$

dove α e β sono numeri complessi tali che $|\alpha|^2 + |\beta|^2 = 1$, è un possibile stato per un qubit. In notazione algebrica, il vettore $|\psi\rangle$ corrisponde a

$$\alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} \equiv \begin{pmatrix} \alpha \\ \beta \end{pmatrix}.$$

Tali stati sono spesso chiamati *sovrapposizioni* (superpositions).

2.2.2 Principio di misurazione

Mentre per un bit classico possiamo sempre determinarne lo stato e stabilire con precisione se è 0 o 1, per un qubit non possiamo determinare con altrettanta precisione il suo stato quantistico, cioè i valori di α e β. La meccanica quantistica ci dice che quando misuriamo un qubit possiamo solo ottenere lo stato $|0\rangle$ con una probabilità pari a $|\alpha|^2$ oppure lo stato $|1\rangle$ con una probabilità pari a $|\beta|^2$. Per questo motivo i valori α e β sono chiamati *ampiezze di probabilità* e la somma $|\alpha|^2 + |\beta|^2$ deve essere 1. Geometricamente questo significa che gli stati di un qubit sono vettori normalizzati di lunghezza 1 (o vettori unitari).

Abbiamo quindi stabilito che un qubit si può trovare in un numero di stati che è infinitamente maggiore di quello dei possibili stati di un bit classico. Si vedrà successivamente che la realizzazione fisica di un qubit non permette di osservare direttamente questi stati: la "misurazione" di un qubit

darà sempre come risultato o lo stato $|0\rangle$ oppure lo stato $|1\rangle$. Si noti tuttavia che i risultati delle misurazioni dipendono strettamente dalle proprietà specifiche dello stato su cui si sono effettuate operazioni di trasformazioni. In questo risiede essenzialmente la potenza del calcolo quantistico.

Esempio 2.2 *Un qubit si può trovare nello stato*

$$\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle,$$

fino al momento in cui viene osservato. Nel momento in cui lo misuriamo, il risultato sarà 0 *nel 50% dei casi e* 1 *nel rimanente 50% dei casi. Cioè otterremo il risultato* 0 *con probabilità* $(1/\sqrt{2})^2 = 1/2$ *e il risultato* 1 *con probabilità* $(1/\sqrt{2})^2 = 1/2$.

2.2.3 Cambio di base

Una qualsiasi base di $\mathbb{C}^2$ può essere vista come una base computazionale.

Per esempio i due qubits $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ e $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ formano anch'essi una base computazionale. Si calcola facilmente che:

$$|0\rangle = \frac{1}{\sqrt{2}}(|+\rangle + |-\rangle) \text{ e } |1\rangle = \frac{1}{\sqrt{2}}(|+\rangle - |-\rangle).$$

Quindi un qubit arbitrario $\alpha|0\rangle + \beta|1\rangle$ può essere espresso nella nuova base come:

$$\begin{aligned} \alpha|0\rangle + \beta|1\rangle &= \tfrac{\alpha}{\sqrt{2}}(|+\rangle + |-\rangle) + \tfrac{\beta}{\sqrt{2}}(|+\rangle - |-\rangle) \\ &= \tfrac{\alpha+\beta}{\sqrt{2}}|+\rangle + \tfrac{\alpha-\beta}{\sqrt{2}}|-\rangle. \end{aligned}$$

Si possono anche fare misurazioni rispetto ad una base diversa dalla base standard $\{|0\rangle, |1\rangle\}$. In questo caso il qubit misurato collasserà a uno degli stati della base computazionale considerata. Nell'esempio visto prima, questi corrispondono a $|+\rangle$ e $|-\rangle$.

2.3 Interpretazione geometrica

Una visualizzazione utile di un qubit si può ottenere mediante un'interpretazione geometrica che associa gli stati di un qubit ai punti sulla superficie di una sfera di raggio unitario. Il polo sud della sfera corrisponde a 1 e il polo nord a 0. Le altre locazioni sono le sovrapposizioni quantistiche di 0 e 1. Questa sfera è nota come la *sfera di Bloch* ed è rappresentata in Figura 1. Molte delle operazioni su un singolo qubit che studieremo possono essere descritte all'interno di questa sfera, che ci aiuta cosi' a coglierne il significato intuitivo.

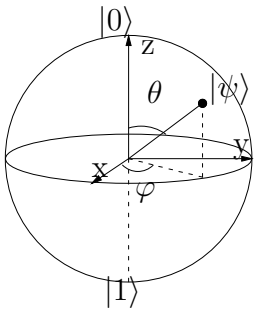

Figure 1: La sfera di Bloch

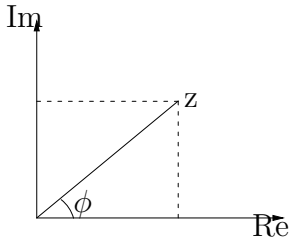

Figure 2: Il piano complesso: $z = a + ib$

Esiste una corrispondenza biunivoca tra un generico stato di un qubit

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,,$$

e un punto sulla sfera unitaria in $\mathbb{R}^3$ rappresentato come

$$\cos(\theta/2)\,|0\rangle + e^{i\varphi}\sin(\theta/2)\,|1\rangle\,,$$

dove θ e φ sono numeri reali (le coordinate sferiche del punto).
Per vedere questa corrispondenza ricordiamo prima la definizione di numero complesso.
Un numero complesso ha la forma $z = a + ib$, dove $i = \sqrt{-1}$, $a = Re(z)$ è la parte reale di z, $b = Im(z)$ è la parte immaginaria di z e $a, b \in \mathbb{R}$ sono numeri reali. La *norma* o *modulo* di z è $|z| = \sqrt{a^2 + b^2}$. Il *coniugato* di z è $z^* = a - ib$. Si denota con $\mathbb{C}$ l'insieme dei numeri complessi.
Un numero complesso $z \in \mathbb{C}$ si può vedere come un punto nel piano complesso in Figura 2.

Due rappresentazioni equivalenti di z come punto in questo piano sono:

Coordinate cartesiane: z è il punto di coordinate a (sull'asse reale Re) e b (sull'asse immaginario Im), cioè $z = a + ib$, con $a, b \in \mathbb{R}$.

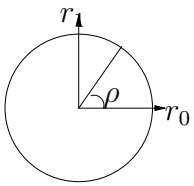

Figure 3: Il cerchio unitario di equazione $r_0^2 + r_1^2 = 1$

Coordinate polari: se ϕ è l'angolo che il vettore z forma con l'asse Re e $r = \sqrt{a^2 + b^2}$ è il modulo di z, allora $a = r\cos(\phi)$ e $b = r\sin(\phi)$. Quindi z è individuato dalle coordinate (r, ϕ), cioè $z = r(\cos(\phi) + i\sin(\phi))$. Dalla formula di Eulero

$$e^{i\phi} = \cos(\phi) + i\sin(\phi),$$

si ottiene quindi la rappresentazione di z come $re^{i\phi}$.

In un qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, i valori α e β sono numeri complessi tali che $|\alpha|^2 + |\beta|^2 = 1$. Usando la descrizione di α e β in coordinate polari possiamo scrivere $|\psi\rangle$ come

$$|\psi\rangle = r_0 e^{i\phi_0}|0\rangle + r_1 e^{i\phi_1}|1\rangle,$$

con

$$r_0^2 + r_1^2 = 1. \tag{1}$$

Notiamo che l'equazione (1) descrive i punti del cerchio unitario in $\mathbb{R}^2$ (cf. Figura 3). Possiamo quindi rappresentare i moduli di α e β mediante l'angolo ρ, ponendo

$$r_0 = \cos(\rho) \text{ e } r_1 = \sin(\rho).$$

Ponendo $\rho = \theta/2$, otteniamo l'espressione

$$|\psi\rangle = \cos(\theta/2)e^{i\phi_0}|0\rangle + \sin(\theta/2)e^{i\phi_1}|1\rangle,$$

con $0 \leq \theta \leq \pi$, o equivalentemente

$$|\psi\rangle = e^{i\gamma}(\cos(\theta/2)|0\rangle + e^{i\varphi}\sin(\theta/2)|1\rangle),$$

con $\varphi = \phi_1 - \phi_0$ e $\gamma = \phi_0$, $0 \leq \varphi \leq 2\pi$.

Vedremo che da un punto di vista fisico il fattore $e^{i\gamma}$ (detto fase globale) si può ignorare in quanto non ha *effetti osservabili*, cioè dal punto di vista osservazionale i due stati $e^{i\gamma}|\psi\rangle$ e $|\psi\rangle$ sono identici (dal principio

di misurazione quantistica). Notiamo infine che l'angolo sferico θ che un punto sulla sfera unitaria in $\mathbb{R}^3$ forma con l'asse z soddisfa esattamente la stessa condizione $0 \leq \theta \leq \pi$ dell'angolo θ nella rappresentazione del qubit $\cos(\theta/2)\,|0\rangle + e^{i\varphi}\sin(\theta/2)\,|1\rangle$. Anche l'angolo φ in questa rappresentazione varia nello stesso intervallo $0 \leq \varphi \leq 2\pi$ dell'angolo che la proiezione di un vettore unitario nella sfera di Bloch sul piano (x, y) forma con l'asse x.

Quindi esiste effettivamente una corrispondenza biunivoca tra i qubits rappresentati come

$$|\psi\rangle = \cos(\theta/2)\,|0\rangle + e^{i\varphi}\sin(\theta/2)\,|1\rangle\,,$$

e i punti sulla sfera di Bloch.

Poichè possiamo codificare nel numero θ stringhe di bits di lunghezza arbitraria, l'informazione classica che può contenere un qubit sembrerebbe infinita. Tuttavia l'unico modo per estrarre l'informazione contenuta in un qubit è attraverso una misurazione. Secondo le leggi della meccanica quantistica il risultato di tale misurazione è sempre un singolo bit classico – 0 oppure 1 – con probabilità che dipende dalla "latitudine" del qubit.

2.4 Interpretazione fisica

La descrizione astratta di un qubit come un vettore in uno spazio bi- dimensionale complesso ha un corrispondente nel mondo reale. In particolare, un qualsiasi sistema fisico con almeno due livelli di energia discreti e sufficientemente separati è un candidato appropriato per rappresentare un qubit. Per realizzare fisicamente un qubit i tre approcci più comuni sono quelli basati su:

- le due diverse polarizzazione di un fotone;
- l'allineamento di uno spin nucleare in un campo magnetico uniforme;
- due livelli di energia[1] di un elettrone che orbita in un singolo atomo.

Ad esempio possiamo considerare il sistema costituito dall'atomo di idrogeno H^2. In questo sistema, lo stato $|0\rangle$ del qubit può essere rappresentato dal primo livello di energia ($n = 0$), corrispondente allo stato

[1] In un atomo i livelli di energia dei vari elettroni sono discreti. Due di essi possono essere selezionati per rappresentare i valori logici 0 e 1. Questi livelli corrispondono a specifici stati di eccitazione degli elettroni nell'atomo.

[2] Negli esperimenti in laboratorio, gli atomi usati sono tipicamente quelli di rubidio e berillio.

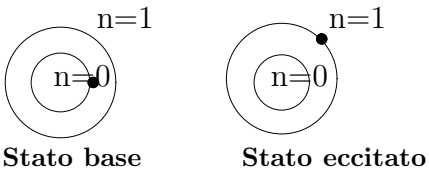

Figure 4: Qubit rappresentato da un elettrone in un atomo di idrogeno

base dell'elettrone, e lo stato $|1\rangle$ dal secondo livello di energia ($n = 1$) corrispondente allo stato eccitato dell'elettrone, come in Figura 4. Il passaggio dell'elettrone da uno stato all'altro può essere realizzato sottoponendo l'elettrone ad un impulso laser di appropriata intensità, durata e lunghezza d'onda. Riducendo opportunamente la durata, si può realizzare il passaggio di un elettrone inizialmente nello stato $|0\rangle$ ad uno stato che si trova "a metà" tra $|0\rangle$ e $|1\rangle$, corrispondente allo stato $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ dell'esempio 2.2.

Come illustrato nell'Esempio 2.2, quando osserviamo un qubit, il risultato può solo essere 0 oppure 1. Inoltre, la misurazione che abbiamo fatto *cambia* lo stato del qubit facendolo collassare dalla sua sovrapposizione di $|0\rangle$ e $|1\rangle$ allo stato specifico consistente con il risultato della misurazione. Queste proprietà si spiegano mediante i principi della meccanica quantistica.

3 Registri quantistici

Con due bits classici possiamo formare quattro possibili stati: 00, 01, 10, 11. In generale, con n bits è possibile costruire 2^n stati distinti. Quanti stati si possono ottenere con n qubits? Lo spazio degli stati generato da un sistema di n qubits ha dimensione 2^n: ogni vettore normalizzato in questo spazio rappresenta un possibile stato computazionale, che chiameremo *registro quantistico a n qubits*. Questa crescita esponenziale nel numero dei qubits delle dimensioni dello spazio degli stati suggerisce la potenziale capacità di un computer quantistico di elaborare informazioni ad una velocità esponenzialmente superiore a quella di un computer classico. Si noti che per $n = 200$ si ottiene un numero che è più grande del numero di atomi nell'universo.

Formalmente un registro quantistico di n qubits è un elemento dello spazio di Hilbert 2^n-dimensionale, $\mathbb{C}^{2^n}$, con base computazionale formata da 2^n registri a n qubits

$$|i_1\rangle \otimes |i_2\rangle \otimes \ldots \otimes |i_n\rangle$$

con $i_j \in \{0,1\}$, $1 \leq j \leq n$. Per convenienza, questo vettore della base si scrive $|i_1\rangle |i_2\rangle \dots |i_n\rangle$ oppure semplicemente $|i_1 i_2 \dots i_n\rangle$.

Consideriamo il caso di due qubits. In analogia con il singolo qubit, possiamo costruire la base computazionale dello spazio degli stati come formata dai vettori $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. Come osservato precedentemente, $|x, y\rangle$ è un'abbreviazione di $|x\rangle \otimes |y\rangle$, il prodotto tensore di x e y. In notazione algebrica questi vettori corrispondono quindi a

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Un registro quantistico a due qubits è una sovrapposizione della forma:

$$|\psi\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle ,$$

con la condizione di normalizzazione $\sum_{i \in \{0,1\}^2} |\alpha_i|^2 = 1$.

Analogamente al caso di un singolo qubit, il risultato di una misurazione sarà uno degli stati $i \in \{0,1\}^2$ con probabilità $|\alpha_i|^2$.

In un sistema di n qubits possiamo anche misurare solo un sottoinsieme degli n qubits. Per esempio, nel caso di un registro a due qubit possiamo misurare il primo qubit ottenendo come risultato 0 con probabilità $|\alpha_{00}|^2 + |\alpha_{01}|^2$. Dopo aver effettuato la misurazione lo stato collasserà a

$$\frac{\alpha_{00} |00\rangle + \alpha_{01} |01\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}}.$$

Nota che lo stato viene rinormalizzato per avere lunghezza 1.

Esercizio 3.1 *Scrivere le basi computazionali per un registro quantistico a* 3 *qubits e a* 4 *qubits.*

3.1 Prodotto tensore

Il prodotto tensore è un'operazione che combina spazi vettoriali per formare spazi vettoriali più grandi. La costruzione generale per spazi vettoriali complessi di dimensione finita è definita come segue.

Nota che per ogni intero positivo m, lo spazio vettoriale complesso m-dimensionale $\mathbb{C}^m$ ha come base standard

$$b_1^m, b_2^m, \dots, b_m^m,$$

dove il vettore colonna di dimensione m, b_j^m, ha componenti tutte nulle tranne la j-sima che è 1. Quindi, ogni vettore $u \in \mathbb{C}^m$ si può scrivere come $\sum_{j=1}^{m} u_j b_j^m$, per qualche $u_j \in \mathbb{C}$:

$$u = \begin{pmatrix} u_1 \\ \vdots \\ u_j \\ \vdots \\ u_m \end{pmatrix}$$

Dati due spazi vettoriali $\mathbb{C}^k$ e $\mathbb{C}^l$, si definisce il prodotto tensore come la funzione

$$\otimes : \mathbb{C}^k \times \mathbb{C}^l \to \mathbb{C}^{kl},$$

con

$$v \otimes w = \begin{pmatrix} v_1 w \\ \vdots \\ v_j w \\ \vdots \\ v_k w \end{pmatrix}$$

dove per ogni $1 \leq j \leq k$, $v_j w$ è la moltiplicazione del vettore colonna $w \in \mathbb{C}^l$ per lo scalare $v_j \in \mathbb{C}$.

Per definizione, il prodotto tensore soddisfa le seguenti proprietà. Indichiamo con z un arbitrario scalare in $\mathbb{C}$ e con V e W due generici spazi di Hilbert di dimensione k e l rispettivamente.

1. Per ogni $|v\rangle \in V$ e $|w\rangle \in W$,

$$z(|v\rangle \otimes |w\rangle) = (z\,|v\rangle) \otimes |w\rangle = |v\rangle \otimes (z\,|w\rangle).$$

2. Per ogni $|v_1\rangle, |v_2\rangle \in V$ e $|w\rangle \in W$,

$$(|v_1\rangle + |v_2\rangle) \otimes |w\rangle = |v_1\rangle \otimes |w\rangle + |v_2\rangle \otimes |w\rangle\,.$$

3. Per ogni $|v\rangle \in V$ e $|w_1\rangle, |w_2\rangle \in W$,

$$|v\rangle \otimes (|w_1\rangle + |w_2\rangle) = |v\rangle \otimes |w_1\rangle + |v\rangle \otimes |w_2\rangle\,.$$

Esercizio 3.2 *Dimostra la proprietà di bilinearità:*

$$(\alpha v + \alpha' v') \otimes (\beta w + \beta' w') = \\ \alpha\beta v \otimes w + \alpha\beta' v \otimes w' + \alpha'\beta v' \otimes w + \alpha'\beta' v' \otimes w',$$

dove $\alpha, \beta, \alpha', \beta' \in \mathbb{C}$, $v, v' \in \mathbb{C}^k$ *e* $w, w' \in \mathbb{C}^l$.

Esercizio 3.3 *Dimostra che:*

$$b_i^k \otimes b_j^l = b_{(i-1)l+j}^{kl}.$$

Esercizio 3.4 *Dimostra che per ogni* $v, v' \in \mathbb{C}^k$ *e* $w, w' \in \mathbb{C}^l$,

$$\langle v \otimes w | v' \otimes w' \rangle = \langle v | v' \rangle \langle w | w' \rangle.$$

3.1.1 Prodotto tensore di matrici

Dati due operatori lineari con matrici di rappresentazione

$$M : \mathbb{C}^k \mapsto \mathbb{C}^k \text{ e } N : \mathbb{C}^l \mapsto \mathbb{C}^l,$$

rispetto alle basi standard di $\mathbb{C}^k$ e $\mathbb{C}^l$, il prodotto tensore di M ed N è l'operatore lineare su $\mathbb{C}^{kl}$ con matrice di rappresentazione

$$M \otimes N : \mathbb{C}^{kl} \mapsto \mathbb{C}^{kl}$$

definita da

$$M \otimes N = \begin{bmatrix} M_{11}N & M_{12}N & \dots & M_{1k}N \\ M_{21}N & M_{22}N & \dots & M_{2k}N \\ \vdots & \vdots & \vdots & \vdots \\ M_{k1}N & M_{k2}N & \dots & M_{kk}N \end{bmatrix},$$

dove M_{ij} è l'elemento di indici i, j della matrice M e $M_{ij}N$ è la matrice $l \times l$ ottenuta moltiplicando N per il numero complesso M_{ij}.

Esempio 3.5 *Il prodotto tensore delle matrici*

$$M = \begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix} \text{ e } N = \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix}$$

è la matrice

$$M \otimes N = \begin{bmatrix} 0 & 1 & 0 & 3 \\ -1 & 2 & -3 & 6 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 2 \end{bmatrix}.$$

Esercizio 3.6 *Dimostrare le seguenti proprietà:*

1. $(M \otimes N)(v \otimes w) = (Mv) \otimes (Nw)$.
2. $(\alpha M + \alpha' M') \otimes (\beta N + \beta' N') =$
   $\alpha\beta M \otimes N + \alpha\beta' M \otimes N' + \alpha'\beta M' \otimes N + \alpha'\beta' M' \otimes N'$.
3. $(M \otimes N)(M' \otimes N') = (MM') \otimes (NN')$.
4. $$\begin{aligned} (M \otimes N)^* &= M^* \otimes N^*, \\ (M \otimes N)^T &= M^T \otimes N^T, \\ (M \otimes N)^\dagger &= M^\dagger \otimes N^\dagger. \end{aligned}$$
5. *Se M ed N sono unitarie (invertibili), allora anche $M \otimes N$ è unitaria (invertibile).*

3.2 Stati "entangled"

Una proprietà importante dei registri quantistici a n qubits è che non è sempre possibile decomporli negli stati dei qubit componenti. Gli stati di questo tipo sono detti *entangled* e godono di proprietà che non si possono ritrovare in nessun oggetto della fisica classica. I membri di una collezione entangled non hanno un proprio stato individuale, solo l'intera collezione corrisponde a uno stato ben definito. Gli stati entangled si comportano come se fossero strettamente connessi l'uno all'altro indipendentemente dalla distanza che li separa. Ad esempio, una misurazione di uno dei due stati di una coppia entangled fornisce simultaneamente informazioni riguardo all'altro. Questa proprietà è alla base di soluzioni di problemi in information-processing che non possono essere riprodotte classicamente. Un esempio che vedremo nel seguito è la realizzazione di circuiti quantistici per il teletrasporto di uno stato quantistico da una locazione all'altra.

Esempio 3.7 (Entanglement) *Lo stato $|00\rangle + |11\rangle$ non può essere fattorizzato nel prodotto tensore di due qubits indipendenti, cioè non esistono a_1, a_2, b_1, b_2 tali che*

$$|00\rangle + |11\rangle = (a_1 |0\rangle + b_1 |1\rangle) \otimes (a_2 |0\rangle + b_2 |1\rangle).$$

Esercizio 3.8 *Dimostrare l'affermazione dell'Esempio 3.7.*

Introduzione ai circuiti quantistici

4 Porte logiche quantistiche

Abbiamo studiato la descrizione quantistica degli stati di una computazione. Vediamo ora come questi stati evolvono dando luogo ad una computazione quantistica.

Analogamente ai computer classici, un computer quantistico è formato da circuiti quantistici costituiti da *porte logiche quantistiche* elementari.

Nel caso classico esiste un'unica porta logica (non banale) a un bit, la porta `NOT`, che implementa l'operazione logica di negazione definita mediante una tabella di verità in cui $1 \to 0$ e $0 \to 1$.

Per definire un'operazione analoga su un qubit, non possiamo limitarci a stabilire la sua azione sugli stati di base $|0\rangle$ e $|1\rangle$, ma dobbiamo specificare anche come deve essere trasformato un qubit che si trova in una sovrapposizione degli stati $|0\rangle$ e $|1\rangle$. Intuitivamente, il `NOT` dovrebbe scambiare i ruoli dei due stati fondamentali e trasformare $\alpha|0\rangle + \beta|1\rangle$ in $\beta|0\rangle + \alpha|1\rangle$. Chiaramente $|0\rangle$ si trasformerebbe in $|1\rangle$ e $|1\rangle$ in $|0\rangle$. L'operazione che implementa questo tipo di trasformazione è un'operazione *lineare* e, come vedremo, questa è una proprietà generale della meccanica quantistica che si giustifica sperimentalmente. Un modo conveniente per rappresentare operazioni lineari è per mezzo di matrici.

4.1 Rappresentazione di operatori lineari

Una funzione $L : \mathbb{C}^2 \mapsto \mathbb{C}^2$ si dice lineare se per ogni $a_1, a_2 \in \mathbb{C}$ e $v_1, v_2 \in \mathbb{C}^2$:

$$L(a_1 v_1 + a_2 v_2) = a_1 L(v_1) + a_2 L(v_2).$$

Dato un vettore $u \in \mathbb{C}^2$, il duale di u è la funzione lineare $L_u : \mathbb{C}^2 \mapsto \mathbb{C}$ definito da $L_u(w) = (u, w) = u^\dagger w$. Spesso si identifica L_u con $u^\dagger$. Nella notazione di Dirac il duale di un vettore $|\psi\rangle$ è denotato da $\langle\psi|$ e il prodotto scalare di $|\psi\rangle$ e $|\phi\rangle$ da $\langle\psi|\phi\rangle$. Per vettori a norma 1 il significato di $L_u(w) = \langle u|w\rangle$ è la proiezione di w nella direzione di u.

Esercizio 4.1 *Dimostrare che la somma di due funzioni lineari è lineare.*

Esercizio 4.2 *Dimostrare che la funzione lineare $|\psi\rangle\langle\phi| : \mathbb{C}^2 \mapsto \mathbb{C}^2$ definita da $|\psi\rangle\langle\phi|(|x\rangle) = \langle\phi|x\rangle|\psi\rangle$ è lineare.*

La matrice di rappresentazione di una funzione lineare L nella base computazionale $|0\rangle$ e $|1\rangle$ si definisce a partire da $L(|0\rangle)$ e $L(|1\rangle)$ come segue. Supponiamo che

$$L(|0\rangle) = a_{11}|0\rangle + a_{21}|1\rangle$$

e

$$L(|1\rangle) = a_{12}|0\rangle + a_{22}|1\rangle .$$

Allora possiamo scrivere L come

$$L = a_{11}|0\rangle\langle 0| + a_{21}|1\rangle\langle 0| + a_{12}|0\rangle\langle 1| + a_{22}|1\rangle\langle 1| .$$

La matrice

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

è la matrice di rappresentazione di L nella base computazionale $|0\rangle$ e $|1\rangle$.

4.1.1 Cambio di base

Il passaggio da una base ad un'altra è una trasformazione lineare la cui matrice si costruisce come segue. Supponiamo che i vettori $|\psi_1\rangle$ e $|\psi_2\rangle$ formino una base per $\mathbb{C}^2$ e che in tale base risulti $|0\rangle = b_{11}|\psi_1\rangle + b_{21}|\psi_2\rangle$ e $|1\rangle = b_{12}|\psi_1\rangle + b_{22}|\psi_2\rangle$. Allora la matrice del cambio di base è data da:

$$B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} .$$

Il vettore con rappresentazione $(\alpha, \beta)^T$ nella base $|0\rangle$ e $|1\rangle$, avrà coordinate $B(\alpha, \beta)^T$ rispetto alla base $|\psi_1\rangle$ e $|\psi_2\rangle$.

Esempio 4.3 *Considera la base formata dai due qubits* $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ *e* $|-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$. *La matrice del cambio di base da* $|0\rangle$, $|1\rangle$ *a* $|+\rangle$, $|-\rangle$ *è data da*

$$B = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} .$$

Esercizio 4.4 *Verificare che la rappresentazione di L nella nuova base è BAB^{-1}*

Esercizio 4.5 *Dimostrare che la rappresentazione di* `NOT` *nella base* $|0\rangle$ *e* $|1\rangle$ *è la matrice*

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} .$$

Qual è la sua rappresentazione nella base $|+\rangle$, $|-\rangle$*?*

4.2 Matrici unitarie

Data una matrice $n \times m$ A, la trasposta A^T è definita da $(A^T)_{ij} = (A)_{ji}$. La coniugata A^* di A è la matrice $(A^*)_{ij} = (A)^*_{ij}$. La matrice aggiunta $A^\dagger$ di A è la matrice $A^\dagger = (A^T)^*$.

Una matrice A è detta unitaria se $A^\dagger = A^{-1}$, dove A^{-1} è l'inversa di A, cioè $AA^{-1} = I$ (I è la matrice identità, $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$).

Esercizio 4.6 *Dimostrare che le matrici*

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}\ e\ Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

sono unitarie. X, Y e Z sono dette le matrici di Pauli.

Esercizio 4.7 *Dimostrare che*

$$\langle u|Av\rangle = \langle A^\dagger u|v\rangle.$$

Dedurre la seguente proprietà delle matrici unitarie: M è unitaria se e solo se preserva i prodotti scalari, cioè se e solo se $\langle Mu|Mv\rangle = \langle u|v\rangle$, per ogni $u, v \in \mathbb{C}^2$.

4.3 Porte logiche quantistiche a un qubit

La matrice corrispondente al `NOT` quantistico è chiamata per motivi storici X ed è definita da:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Si può infatti verificare che l'applicazione di X a un qubit $\alpha|0\rangle + \beta|1\rangle$ (scritto in notazione vettoriale) è:

$$X \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}.$$

In generale un'operazione su un singolo qubit può essere specificata da una matrice 2×2. Tuttavia non tutte le matrici 2×2 definiscono operazioni "lecite" su qubits. Ricordiamo che la condizione di normalizzazione richiede che $|\alpha^2 + \beta^2| = 1$ in qualsiasi stato quantistico $\alpha|0\rangle + \beta|1\rangle$. La stessa condizione deve valere anche per lo stato che si ottiene dopo aver effettuato l'operazione. La proprietà delle matrici che garantisce la trasformazione di un vettore unitario in un vettore che è ancora unitario è l'*unitarietà*.

Teorema 4.8 *Una funzione lineare trasforma un qubit in un qubit (cioè preserva vettori normalizzati) se e solo se è unitaria.*

Dimostrazione Esercizio.

Contrariamente al caso classico in cui possiamo definire una sola operazione non banale su un singolo bit, nel caso quantistico esistono molte operazioni non banali su un singolo qubit. Oltre al `NOT` due operazioni importanti che useremo in seguito sono la porta Z:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

che agisce solo sulla componente $|1\rangle$ scambiandone il segno, e la porta di *Hadamard*:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Quest'ultima operazione è una delle più utili e viene usata molto spesso nella definizione di circuiti quantistici. Il suo effetto è quello di trasformare uno stato base in una sovrapposizione che risulta, dopo una misurazione nella base computazionale, essere 0 oppure 1 con uguale probabilità. Ad esempio, applicando H a $|0\rangle$ si ottiene:

$$H \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

cioè lo stato $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$. L'effetto di H si può quindi vedere come un `NOT` eseguito a metà in modo che lo stato risultante non è ne' 0 ne' 1, bensì una sovrapposizione coerente dei due stati di base. Per questo motivo H viene spesso chiamata la "radice quadrata di `NOT`". Nota che questa espressione ha un significato solo fisico! Da un punto di vista algebrico, H^2 *non* è la matrice X. Con un semplice calcolo si può infatti verificare che H^2 è l'identità e quindi applicando due volte H ad uno stato lo lascia inalterato.

Nella sfera di Bloch l'operazione H corrisponde ad una rotazione di 90° della sfera intorno all'asse y seguita da una riflessione attraverso il piano (x, y). In Figura 5 è illustrato l'effetto dell'applicazione di H al qubit $|0\rangle$. Si può provare a visualizzare l'effetto di H sul qubit $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$. Per effetto della rotazione e della successiva riflessione attraverso il piano x, y si otterrà nuovamente $|0\rangle$.

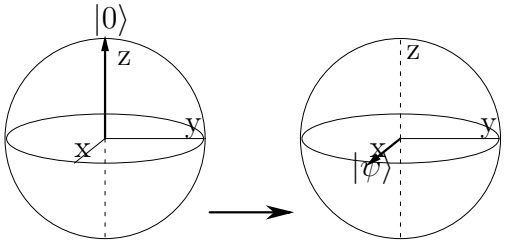

Figure 5: Visualizzazione della porta di Hadamard applicata all'input $|0\rangle$. L'output è $|\psi\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$.

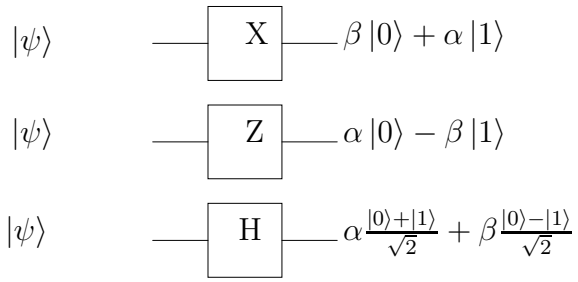

Figure 6: Le porte X, Z e H

Le porte logiche a un qubit X, Z e H sono rappresentate graficamente come in Figura 6.

Le matrici X, Z e la matrice unitaria

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix},$$

sono chiamate *matrici di Pauli* e rappresentano rispettivamente le componenti x, z, y dello spin di un elettrone.

Si può dimostrare che per ogni matrice unitaria U esistono numeri reali α, β, δ e γ tali che

$$U = \begin{bmatrix} e^{i(\alpha-\beta/2-\delta/2)}\cos(\frac{\gamma}{2}) & -e^{i(\alpha-\beta/2+\delta/2)}\sin(\frac{\gamma}{2}) \\ e^{i(\alpha+\beta/2-\delta/2)}\sin(\frac{\gamma}{2}) & e^{i(\alpha+\beta/2+\delta/2)}\cos(\frac{\gamma}{2}) \end{bmatrix}.$$

(esercizio)

4.4 Porte logiche quantistiche a più qubits

Le operazioni su registri quantistici di due o più qubits sono necessarie per descrivere le trasformazioni di stati composti e in particolare dei cosiddetti

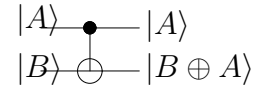

Figure 7: Porta `CNOT`

stati *entangled*. Abbiamo visto che non sempre un registro di due qubits può essere decomposto nel prodotto tensore dei singoli qubits componenti. Di conseguenza non possiamo in questi casi simulare un'operazione sui due qubits mediante operazioni su ciascun qubit componente.

Anche le operazioni su registri di qubits corrispondono a operazioni unitarie come nel caso di un singolo qubit.

Le più importanti porte logiche che implementano operazioni su due bits classici sono le porte `AND`, `OR`, `XOR`, `NAND` e `NOR`. Le porte `NOT` e `AND` formano un insieme *universale*, cioè qualsiasi funzione booleana si può realizzare mediante una combinazione di queste due operazioni. Per lo stesso motivo, il `NAND` costituisce un insieme universale. Nota che `XOR` da solo o anche insieme con `NOT` non è universale: poichè preserva la parità totale dei bits, solo un sottinsieme delle funzioni booleane sono rappresentabili mediante questa operazione.

L'analogo quantistico di `XOR` è la porta `CNOT` (controlled-`NOT`) che opera su due qubits: il primo è chiamato qubit di *controllo* e il secondo è il qubit *target*. Se il controllo è zero allora il target è lasciato inalterato; se il controllo è uno, allora il target viene negato, cioè:

$$|00\rangle \mapsto |00\rangle\,, |01\rangle \mapsto |01\rangle\,, |10\rangle \mapsto |11\rangle\,, |11\rangle \mapsto |10\rangle\,.$$

Equivalentemente, `CNOT` si può vedere come la trasformazione

$$|A, B\rangle \mapsto |A, B \oplus A\rangle\,,$$

dove A è il qubit di controllo, B è il target e $\oplus$ è la somma modulo 2, cioè l'operazione classica `XOR`.

La porta `CNOT` è rappresentata graficamente dal circuito in Figura 7.

La rappresentazione come matrice unitaria è:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

dove la prima colonna descrive la trasformazione del vettore della base computazionale $|00\rangle$, la seconda quella del vettore $|01\rangle$, la terza di $|10\rangle$ e la quarta di $|11\rangle$.

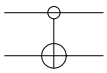

Figure 8: Variante della porta `CNOT`

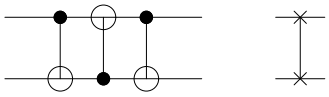

Figure 9: Circuito che scambia due qubits e simbolo schematico

Esercizio 4.9 *Considera un'operazione analoga al* `CNOT` *dove il target è negato se il qubit di controllo è zero anzicchè uno. Questa operazione è rappresentata come in Figura 8. Qual è la matrice che la rappresenta?*

È importante osservare che il `CNOT`, come tutte le trasformazioni unitarie, è invertibile: dall'output si può sempre ottenere l'input. Questo non è vero per le porte logiche `XOR` e `NAND`: in generale le operazioni classiche sono *irreversibili*.

La porta `CNOT` e le porte a un qubit rappresentano i prototipi di tutte le porte logiche quantistiche. Infatti, mostreremo in seguito l'universalità di queste operazioni.

5 Circuiti quantistici

Un semplice esempio di circuito quantistico è dato in Figura 9. Il circuito realizza lo scambio degli stati di due qubits. Dato in input il registro di due qubits $|a, b\rangle$, viene effettuato un `CNOT` con qubit di controllo a. Come risultato b viene rimpiazzato da $a \oplus b$. Quest'ultimo viene preso come controllo di un secondo `CNOT` con target a. L'effetto è che a viene sostituito da $a \oplus (a \oplus b) = b$. Un ultimo `CNOT` con controllo b e target $a \oplus b$ realizza infine lo scambio rimpiazzando $a \oplus b$ con a. Data una qualsiasi operazione unitaria U su n qubits, si può definire il circuito *controlled-U* come la naturale estensione della porta `CNOT` (cf. Figura 10). La linea con il pallino nero indica il qubit di controllo, mentre i qubits target sono gli n inputs di U. Secondo questa convenzione il *controlled-*`NOT` non è altro che un *controlled-U* con $U = X$.

Un'altra operazione importante è rappresentata dal simbolo in Figura 11 e consiste nella misurazione di un qubit $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$. Come sappiamo il risultato è un bit classico M (indicato con una doppia linea) che sarà 0 oppure 1 con probabilità rispettivamente $|\alpha|^2$ e $|\beta|^2$.

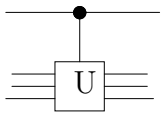

Figure 10: Porta controlled-U

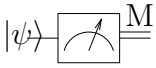

Figure 11: Simbolo di circuito per la misurazione

5.1 No-cloning

È possibile costruire un circuito che fa la copia di un qubit? Si potrebbe pensare di utilizzare un `CNOT` con qubit di controllo contenente il qubit $|x\rangle$ da copiare e il target posto inizialmente a $|0\rangle$. Il risultato sarebbe la copiatura di x nel target. In realtà questo è vero per bit classici (o per gli stati della base computazionale) ma non per un generico qubit $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$. Infatti, consideriamo il circuito in Figura 12, costituito da un `CNOT` che ha come input i qubits $|\psi\rangle$ (controllo) e $|0\rangle$ (target), cioè il registro $|\psi\rangle\,|0\rangle$.

Il nostro obiettivo è ottenere il risultato $|\psi\rangle\,|\psi\rangle$. Osserviamo che

$$|\psi\rangle\,|\psi\rangle = a^2\,|00\rangle + ab\,|01\rangle + ab\,|10\rangle + b^2\,|11\rangle\,.$$

Questo stato è in generale diverso dal risultato del nostro circuito $a\,|00\rangle + b\,|11\rangle$ a meno che $ab = 0$. Quindi questo circuito non effettua la copia di un qubit. In generale, l'impossibilità di copiare un qubit è una proprietà dei sistemi quantistici nota come teorema del *no cloning*:

Teorema 5.1 *Non esiste una trasformazione unitaria M tale che $M\,|\psi\rangle\,|0\rangle = |\psi\rangle\,|\psi\rangle$, per ogni stato $|\psi\rangle$.*

Dimostrazione Supponiamo che esista M tale che $M\,|\psi\rangle\,|0\rangle = |\psi\rangle\,|\psi\rangle$, per ogni qubit $|\psi\rangle$. Allora possiamo scegliere due qubits $|\psi\rangle$ e $|\phi\rangle$ tali che $0 < \langle\psi|\phi\rangle < 1$. Ad esempio, si può prendere $|\psi\rangle = |0\rangle$ e $|\phi\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$. Allora otteniamo che

$$M\,|\psi\rangle\,|0\rangle = |\psi\rangle\,|\psi\rangle \text{ e } M\,|\phi\rangle\,|0\rangle = |\phi\rangle\,|\phi\rangle\,.$$

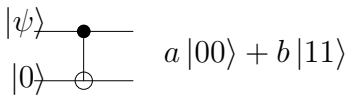

Figure 12: Circuito quantistico per "copiare" un qubit

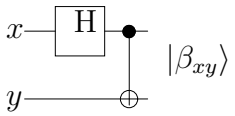

Figure 13: Circuito quantistico per la creazione degli stati di Bell

Facciamo ora il prodotto scalare membro a membro delle due equazioni. Siccome M è unitaria e quindi preserva i prodotti scalari (cf. Esercizio 4.7), e per la proprietà distributiva del prodotto tensore rispetto al prodotto scalare (cf. Esercizio 3.4 Lezione 1) otteniamo che

$$\langle\psi|\phi\rangle\langle\psi|\phi\rangle = \langle\psi|\phi\rangle\langle 0|0\rangle = \langle\psi|\phi\rangle,$$

contraddicendo l'ipotesi $0 < \langle\psi|\phi\rangle < 1$. Quindi M non può esistere.

5.2 Esempi di circuiti quantistici

Descriviamo due circuiti un po' più complicati di quelli visti precedentemente: il primo permette di trasformare i quattro stati computazionali di due qubits in altrettanti stati che sono detti *stati di Bell* o *coppie EPR*; il secondo usa questi stati per realizzare il *teletrasporto* di un qubit. Questi due esempi mostrano come costruire stati computazionali che non hanno alcuna controparte classica e usarli per dar luogo a fenomeni paradossali secondo le leggi della fisica classica. Tali stati sono quelli che abbiamo chiamato *entangled*.

5.2.1 Stati di Bell

Abbiamo visto che la porta `CNOT` può essere usata per creare stati che sono entangled. Il circuito in Figura 13 genera per ogni stato della base computazionale $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ un particolare stato entangled. Questi stati, che indichiamo con $\beta_{00}, \beta_{10}, \beta_{01}, \beta_{11}$, sono chiamati stati di Bell o EPR da Bell, Einstein, Podolsky e Rosen che per primi ne scoprirono le straordinarie proprietà. In particolare, Einstein, Podolsky e Rosen usarono questi stati in un esperimento che nelle loro intenzioni doveva dimostrare che la meccanica quantistica non era in grado di dare una descrizione completa della realtà. Il paradosso che veniva fuori da questo esperimento era che l'interazione tra queste coppie di stati quantistici dava luogo a un fenomeno che violava i principi fondamentali della teoria della relatività. La spiegazione "classica" che essi proponevano fu successivamente smentita da Bell.

Il circuito trasforma il primo qubit in una sovrapposizione che viene poi usata come input di controllo per `CNOT`. Il target viene quindi invertito solo quando il controllo è 1. Pertanto i vettori in input vengono trasformati come segue:

$$\begin{aligned}
|00\rangle &\mapsto |\beta_{00}\rangle \equiv (|00\rangle + |11\rangle / \sqrt{2}) \\
|01\rangle &\mapsto |\beta_{01}\rangle \equiv (|01\rangle + |10\rangle / \sqrt{2}) \\
|10\rangle &\mapsto |\beta_{10}\rangle \equiv (|00\rangle - |11\rangle / \sqrt{2}) \\
|11\rangle &\mapsto |\beta_{11}\rangle \equiv (|01\rangle - |10\rangle / \sqrt{2})
\end{aligned}$$

5.2.2 Teletrasporto quantistico

Il teletrasporto quantistico è una tecnica per trasportare stati quantistici da un posto ad un altro sfruttando solo la trasmissione di bits classici. Questa tecnica è stata scoperta nel 1993 e la sua validità è stata poi confermata da vari risultati sperimentali. In un recente esperimento realizzato all'università di Ginevra e riportato sulla rivista *Nature* N. 421 del 30 gennaio 2003, è stato effettuato il teletrasporto di un qubit tra due laboratori posti a 55 metri di distanza e sfruttando un canale standard di telecomunicazione di 2 Km.

Per capire il tipo di problemi che permette di risolvere, immaginiamo una situazione in cui una persona che chiamiamo Alice deve far conoscere lo stato di un qubit ad un'altra persona che chiameremo Bob. Alice non conosce lo stato del qubit e per il teorema del no-cloning sappiamo che non le è possibile farne una copia. Inoltre Alice può solo mandare a Bob informazione classica, cioè i valori 0 e 1 di un bit classico. In questa situazione sembrebbe impossibile trasmettere il qubit a Bob. Vediamo come ciò è possibile grazie alle proprietà degli stati entangled. L'ipotesi fondamentale è che Bob e Alice possiedono ciascuno un qubit di una coppia EPR generata precedentemente. Alice può operare sul suo qubit e Bob può fare altrettanto sulla sua parte della coppia EPR. Il circuito in Figura 14 illustra come avviene la trasmissione di un qubit $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ di cui si ignorano le ampiezze α e β, da parte di Alice a Bob. Lo stato di input del circuito è $|\psi_0\rangle = |\psi\rangle\,|\beta_{00}\rangle$.

Alice combina $|\psi\rangle$ con la sua metà della coppia EPR e misura i suoi due qubits dopo aver applicato le porte `CNOT` e Hadamard. I due bit che ottiene dopo la misurazione vengono mandati attraverso un canale di comunicazione classico a Bob, il quale sarà in grado di ricostruire lo stato $|\psi\rangle$ sfruttando l'informazione classica ricevuta da Alice e la sua metà della coppia EPR.

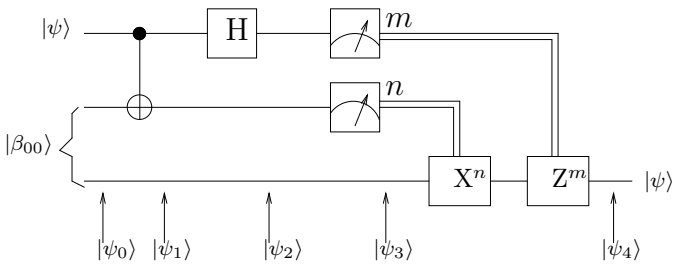

Figure 14: Circuito quantistico per il teletrasporto di un qubit

Nel circuito in figura, le prime due linee corrispondono ai qubits usati da Alice, mentre l'ultima linea corrisponde al qubit posseduto da Bob. L'input è

$$|\psi_0\rangle = |\psi\rangle\,|\beta_{00}\rangle = \frac{1}{\sqrt{2}}[\alpha\,|0\rangle\,(|00\rangle + |11\rangle) + \beta\,|1\rangle\,(|00\rangle + |11\rangle)],$$

dove lo stato $|\beta_{00}\rangle \equiv (|00\rangle + |11\rangle\,/\sqrt{2})$ occupa il secondo qubit di Alice e il qubit di Bob. Come risultato di `CNOT` applicato ai suoi due qubits, Alice ottiene

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}[\alpha\,|0\rangle\,(|00\rangle + |11\rangle) + \beta\,|1\rangle\,(|10\rangle + |01\rangle)].$$

Viene quindi applicato Hadamard al primo qubit, $|\psi\rangle$, ottenendo

$$|\psi_2\rangle = \frac{1}{2}[\alpha(|0\rangle + |1\rangle)(|00\rangle + |11\rangle) + \beta(|0\rangle - |1\rangle)(|10\rangle + |01\rangle)],$$

che si può scrivere equivalentemente come

$$|\psi_2\rangle = \frac{1}{2}[|00\rangle\,(\alpha\,|0\rangle + \beta\,|1\rangle) + |01\rangle\,(\alpha\,|1\rangle + \beta\,|0\rangle) + |10\rangle\,(\alpha\,|0\rangle - \beta\,|1\rangle) + |11\rangle\,(\alpha\,|1\rangle - \beta\,|0\rangle)].$$

A questo punto Alice misura i suoi due qubits ottenendo una delle quattro coppie di bits:

$$00\ 01\ 10\ 11.$$

Per effetto della misurazione anche il qubit di Bob collasserà nello stato corrispondente al risultato della misurazione, cioè:

$$\begin{aligned} 00 &\mapsto \alpha\,|0\rangle + \beta\,|1\rangle \\ 01 &\mapsto \alpha\,|1\rangle + \beta\,|0\rangle \\ 10 &\mapsto \alpha\,|0\rangle - \beta\,|1\rangle \\ 11 &\mapsto \alpha\,|1\rangle - \beta\,|0\rangle \end{aligned}$$

Nella figura questo stato è indicato con $|\psi_3\rangle$.

Alice comunica i due bit mn ottenuti a Bob mediante un canale classico. Bob è ora in grado di ottenere il qubit $|\psi\rangle$ applicando al suo qubit il circuito $X^n Z^m$, dove

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Introduzione alla meccanica quantistica

Fino ad ora abbiamo parlato di sistemi quantistici, stati quantistici, evoluzione e misurazione di stati quantistici, ecc., ma non abbiamo ancora definito questi termini in modo formale. Un modello matematico che ci permette di farlo è la teoria quantistica.

6 I postulati della meccanica quantistica

La meccanica quantistica fornisce la descrizione più accurata e completa delle leggi che governano il mondo fisico. Il formalismo matematico su cui si basa e la realtà fisica che descrive sono messi in relazione mediante alcuni postulati fondamentali.

Postulato 1 Ogni sistema fisico isolato ha associato uno spazio di Hilbert complesso, detto *spazio degli stati* del sistema. Il sistema è completamente descritto dal suo *vettore di stato*, che è un vettore unitario nello spazio degli stati.

Il sistema fisico isolato più semplice è il *qubit*. Lo spazio di Hilbert associato è $\mathbb{C}^2$. La base computazionale formata da $|0\rangle$ e $|1\rangle$ è una base ortonormale e la condizione che ogni vettore $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$ (con $a, b \in \mathbb{C}$) sia un vettore unitario è espressa equivalentemente da $|a|^2 + |b|^2 = 1$, oppure mediante il prodotto interno da $\langle\psi|\,\psi\rangle = 1$. Ricordiamo che a e b sono chiamate le *ampiezze* rispettivamente del vettore $|0\rangle$ e del vettore $|1\rangle$. Osserviamo inoltre che uno stato di un sistema quantistico è in realtà una classe di equivalenza di vettori che differiscono per moltiplicazione di uno scalare complesso non nullo; il vettore unitario è il rappresentante di questa classe. Quindi lo stato $e^{i\theta}\,|\psi\rangle$, dove $|\psi\rangle$ è uno stato e θ è un numero reale, è equivalente a $|\psi\rangle$. Il fattore $e^{i\theta}$ è detto *fattore di fase globale*. Questa identificazione degli stati con fasi globali diverse è giustificata dal fatto che le statistiche di *misurazione* (cf. Postulato 3) di questi stati sono le stesse. Un altro tipo di fattore, detto *fase relativa*, è invece fisicamente significativo e due stati che differiscono

per un fattore di fase relativa non possono essere identificati. Ad esempio, identifichiamo $a\,|\varphi\rangle + b\,|\psi\rangle$ con $e^{i\theta}(a\,|\varphi\rangle + b\,|\psi\rangle)$ ma non con $a\,|\varphi\rangle + e^{i\theta}b\,|\psi\rangle$.

Postulato 2 L'evoluzione di un sistema quantistico *chiuso* è descritto da una *trasformazione unitaria*: lo stato $|\psi\rangle$ del sistema al tempo t_1 è in relazione con lo stato $|\psi'\rangle$ del sistema al tempo t_2 mediante un operatore unitario U che dipende solo da t_1 e t_2,

$$\left|\psi'\right\rangle = U\,|\psi\rangle\,.$$

Per il semplice sistema quantistico rappresentato da un qubit abbiamo visto vari esempi di operatori unitari: le matrici di Pauli,

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \text{ e } Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

e la porta di Hadamard

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

X è detta anche porta logica `NOT` e Y corrisponde a iZX.
Una versione più raffinata di questo postulato descrive l'evoluzione di un sistema quantistico chiuso in *tempo continuo*. Diamo questa versione nel Postulato 2' per completezza, ma non ne faremo uso negli argomenti che tratteremo successivamente, in cui faremo sempre uso della formulazione unitaria della dinamica quantistica.

Postulato 2' L'evoluzione nel tempo dello stato di un sistema quantistico chiuso è descritta dall'*equazione di Schrödinger*

$$i\hbar\frac{d|\psi\rangle}{dt} = H\,|\psi\rangle\,,$$

dove $\hbar$è una costante fisica, nota come la *costante di Planck*, il cui valore deve essere determinato sperimentalmente. Hè un operatore Hermitiano fissato, noto come l'*Hamiltoniano* del sistema chiuso.

In generale, trovare l'Hamiltoniano di un sistema, cioè l'operatore che ne descrive completamente la dinamica, è un problema molto difficile che richiede un numero sostanziale di esperimenti per poter essere risolto. Noi non ci porremo questo tipo di problemi; per i nostri scopi ci basterà sapere che ogni operatore unitario U (del tipo di quelli che useremo per descrivere

una computazione quantistica) può essere realizzato come una soluzione dell'equazione di Shröndinger e che esiste una corrispondenza biunivoca tra la descrizione della dinamica di un sistema a *tempo discreto* mediante operatori unitari e la descrizione a *tempo continuo* mediante gli Hamiltoniani.

Postulato 3 Ogni *osservabile* M (cioè ogni proprietà di un sistema fisico che può essere misurata) è rappresentato da un operatore Hermitiano sullo spazio degli stati del sistema che viene osservato e viceversa. M ha la forma

$$M = \sum_m m P_m,$$

dove P_m è la proiezione sul sottospazio degli autovettori di M corrispondenti all'autovalore m (ricordiamo che gli autovalori di operatori Hermitiani sono reali, cf. Capitolo 8). I possibili risultati di una misura dell'osservabile M corrispondono agli autovalori m di M. Dopo una misura di M nello stato ψ, la probabilità di ottenere il risultato m è

$$p(m) = \langle\psi| P_m |\psi\rangle = \| P_m |\psi\rangle \|^2.$$

Lo stato del sistema immediatamente dopo una misura con risultato m è

$$\frac{P_m |\psi\rangle}{\sqrt{p(m)}}.$$

Gli operatori P_m soddisfano l'equazione di completezza

$$\sum_m P_m = I.$$

L'equazione di completezza esprime il fatto che la somma delle probabilità dei risultati di una misura deve essere 1. Infatti:

$$1 = \sum_m p(m) = \sum_m \langle\psi| P_m |\psi\rangle = \langle\psi|\psi\rangle .$$

Questo tipo di misurazione viene spesso chiamata *misurazione proiettiva completa* (o misurazione di von Neumann) perché l'osservabile M è determinato da un qualsiasi insieme di operatori di proiezione ortogonali P_m che soddisfano la relazione di completezza e tali che $P_m P_{m'} = \delta_{mm'} P_m$. Misurare nella base $\{|m\rangle\}$, dove $\{|m\rangle\}$ è una base ortonormale, significa eseguire una misurazione proiettiva con operatori di proiezione $|m\rangle\langle m|$.

Esempio 6.1 (qubit) *Consideriamo l'osservabile* $M \equiv 0P_0 + 1P_1$ *e misuriamolo nello stato* $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. *Otterremo il risultato* 0 *con probabilità*

$$p(0) = \|P_0|\psi\rangle\|^2 = \||0\rangle\langle 0|\psi\rangle\|^2 = |a|^2$$

e analogamente

$$p(1) = \|P_1|\psi\rangle\|^2 = \||1\rangle\langle 1|\psi\rangle\|^2 = |b|^2.$$

Inoltre, se abbiamo osservato 0 *allora lo stato* $|\psi\rangle$ *diventerà* $|0\rangle = |0\rangle\langle 0|\psi\rangle$, *mentre se il risultato era* 1 *diventerà* $|1\rangle = |1\rangle\langle 1|\psi\rangle$.

Il risultato della misura di un osservabile M in uno stato $|\psi\rangle$ è in generale aleatorio. Una formula utile per determinare il valore medio del risulato della misura di M si ottiene mediante la nozione nota in teoria della probabilità come *media* o *speranza*, $\mathbf{E}$, di una variabile aleatoria (nel nostro caso l'osservabile M), corrispondente alla media su tutti i possibili risultati, pesata dalle probabilità associate:

$$\begin{aligned}
\mathbf{E}_{|\psi\rangle}(M) &= \sum_m mp(m) \\
&= \sum_m m\langle\psi|P_m|\psi\rangle \\
&= \langle\psi|\left(\sum_m mP_m\right)|\psi\rangle \\
&= \langle\psi|M|\psi\rangle .
\end{aligned}$$

Il valore medio di un osservabile M si indica spesso con la notazione $< M >$.

Esempio 6.2 *Si vuole misurare l'osservabile* Z *nello stato* $|\psi\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$. *Poichè* Z *ha autovalori* $+1$ *e* -1 *con corrispondenti autovettori* $|0\rangle$ *e* $|1\rangle$, *si ha che*

$$Z = |0\rangle\langle 0| - |1\rangle\langle 1| .$$

I possibili risultati sono quindi 1 *e* -1 *con probabilità* $p(1) = \langle\psi|0\rangle\langle 0|\psi\rangle = \frac{1}{2}$ *e* $p(-1) = \langle\psi|1\rangle\langle 1|\psi\rangle = \frac{1}{2}$, *con valore medio* $< Z >= 0$. *Lo stato dopo la misurazione risulterà*

$$\begin{aligned}
\textit{se } m = 1 \qquad & \sqrt{2}(|0\rangle\langle 0|\psi\rangle) = |0\rangle \\
\textit{se } m = -1 \qquad & \sqrt{2}(|1\rangle\langle 1|\psi\rangle) = |1\rangle .
\end{aligned}$$

Postulato 4 Lo spazio degli stati di un sistema fisico composto è il prodotto tensore degli spazi degli stati dei sistemi fisici componenti. Se il sistema è composto da n sottosistemi e il componente i-esimo si trova nello stato $|\psi_i\rangle$, allora lo stato del sistema totale è $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \ldots \otimes |\psi_n\rangle$.

Questo postulato descrive come costruire lo spazio degli stati di un sistema quantistico composto da due o più sistemi fisici distinti a partire dagli spazi degli stati dei sistemi componenti.

Esercizio 6.3 *Dato un sistema di due qubits, dimostrare che il valore medio dell'osservabile $X_1 Z_2$ misurato nello stato $(|00\rangle + |11\rangle)/\sqrt{2}$ è zero. La notazione X_1 e Z_2 indica rispettivamente l'operatore di Pauli X applicato al primo qubit e l'operatore di Pauli Z applicato al secondo qubit.*

Il Postulato 4 permette di definire la nozione di *entanglement* che, insieme al problema della misurazione, rappresenta una delle proprietà più discusse dei sistemi quantistici. Uno stato di un sistema composto si dice "entangled" se non può essere scritto come prodotto dei suoi sistemi componenti.

Esercizio 6.4 *Considera il seguente stato di un registro di due qubit:*

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$

Dimostrare che non esistono stati $|a\rangle$ e $|b\rangle$ dei due qubit componenti tali che $|\psi\rangle = |a\rangle\,|b\rangle$.

Come già visto nell'esempio del teletrasporto, gli stati entangled possono essere utilizzati per ottenere dei risultati sorprendenti (come la trasmissione dello stato di un qubit attraverso due soli bit di informazione classica). Nella sezione 7 viene presentato un altro esempio in cui l'entanglement viene utilizzato per comunicare due bit di informazione classica attraverso la trasmissione di un solo qubit. L'esempio illustra anche l'applicazione del Postulato 3.

6.1 Il problema della misurazione quantistica

Il problema della misurazione quantistica diede luogo ad accesi dibattiti sin dalla nascita della teoria quantistica nel 1920. John von Neumann introdusse il primo trattamento assiomatico rigoroso della meccanica quantistica nel 1955, intervenendo in maniera decisiva anche sul problema della misurazione

e fornendo una spiegazione chiara ai vari paradossi che erano stati introdotti per sostenere l'inadeguatezza della teoria quantistica.

Secondo la formalizzazione di von Neumann il processo di misurazione avviene in due fasi. Nella prima fase l'operatore Hermitiano che rappresenta l'osservabile viene applicato allo stato del sistema generando uno stato entangled. In una seconda fase avviene il cosiddetto "quantum leap" o "riduzione di stato", cioè il salto dallo stato enatngled allo stato corrispondente ad uno degli autovettori dell'osservabile. Questa riduzione è nondeterministica e conseguentemente non c'e' modo di prevedere quale dei risultati sarà ottenuto prima che il processo di misurazione abbia termine. In altre parole, per un osservabile non è mai possibile stabilire in maniera definita il valore che verrà misurato. La meccanica quantistica fornisce tuttavia delle informazioni statistiche sui possibili risultati di una misurazione secondo quella che è nota come l'*interpretazione statistica di Born*. Attraverso misurazioni fatte su copie del sistema opportunamente preparate, è possibile stabilire la distribuzione probabilistica dei risultati. Il significato di *probabilità* di un risultato va inteso secondo l'interpretazione data in teoria delle probabilità come *frequenza relativa*: la probabilità di un risultato e' il rapporto tra il numero delle volte che l'esperimento ha successo (cioè si ottiene quel risultato) e il numero totale degli esperimenti fatti, purché si ripeta l'esperimento un numero sufficientemente grande di volte.

7 Un'applicazione: Codifica superdensa

Alice deve comunicare a Bob l'informazione contenuta in due bit classici, (cioè un numero tra $0, 1, 2, 3$) e deve farlo trasmettendo un solo qubit. Poichè la misurazione di un qubit può dare come risultato solo un bit di informazione classica (0 o 1), questo compito sembrerebbe impossibile. In realtà il problema può essere risolto attraverso l'uso di una coppia EPR. Supponiamo che inizialmente Alice e Bob siano in possesso rispettivamente del primo e del secondo qubit della coppia entangled

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$

Alice applica al suo qubit una delle trasformazioni I, X, iY, Z a seconda del numero che vuole trasmettere. In particolare, lo stato $|\psi\rangle$ viene trasformato nel seguente modo:

$$0 \quad : \quad |\psi\rangle \to (I \otimes I)\,|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

$$
\begin{aligned}
1 \quad &: \quad |\psi\rangle \to (X \otimes I)\,|\psi\rangle = \frac{|10\rangle + |01\rangle}{\sqrt{2}} \\
2 \quad &: \quad |\psi\rangle \to (iY \otimes I)\,|\psi\rangle = \frac{-|10\rangle + |01\rangle}{\sqrt{2}} \\
3 \quad &: \quad |\psi\rangle \to (Z \otimes I)\,|\psi\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}
\end{aligned}
$$

I quattro stati risultanti formano una base ortonormale nota come *base di Bell*. Allora Alice non deve fare altro che inviare il suo qubit a Bob, il quale può ora determinare i due bits che Alice voleva trasmettergli, semplicemente attraverso una misura nella base di Bell. Più precisamente, indicando con $|\beta_0\rangle, |\beta_1\rangle, |\beta_2\rangle, |\beta_3\rangle$ i quattro stati di Bell, per ricevere l'informazione Bob deve misurare un osservabile del tipo

$$M \equiv \sum_{i=0}^{3} i\,|\beta_i\rangle\langle\beta_i|$$

nello stato dei due qubit in suo possesso. Nota che il risultato della misura ha sempre probabilità 1. Infatti, la probabilità di ottenere i nello stato $|\beta_j\rangle$ è data da

$$p(i) = \langle\beta_j|\beta_i\rangle\langle\beta_i|\beta_j\rangle = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases}$$

Esercizio 7.1 *Verificare che Bob può determinare i due bits attraverso misurazioni su un singolo qubit. (Suggerimento: misurare il secondo qubit e applicare Hadamard al primo qubit dello stato risultante).*

8 Richiami di Algebra Lineare

8.1 Prodotto interno e spazi di Hilbert

Dato uno spazio vettoriale V, una funzione $(\cdot,\cdot) : V \times V \mapsto \mathbb{C}$ è un *prodotto interno* (o *prodotto scalare*) se soddisfa i seguenti requisiti:

- $(|v\rangle, |v\rangle) \geq 0$
- $(|v\rangle, |v\rangle) = 0$ se e solo se $v = 0$
- $(|v\rangle, |w\rangle) = (|w\rangle, |v\rangle)^*$
- $(|v\rangle, \sum_i a_i |w_i\rangle) = \sum_i a_i (|v\rangle, |w_i\rangle)$.

Esercizio 8.1 *Verificare che la funzione definita da*

$$((y_1, y_2, \ldots, y_n), (z_1, z_2, \ldots, z_n)) = \sum_{i=1}^{n} y_i^* z_i,$$

dove $(y_1, y_2, \ldots, y_n)$ *e* $(z_1, z_2, \ldots, z_n)$ *sono vettori in* $\mathbb{C}^n$*,è un prodotto interno.*

Dalla definizione segue che ogni prodotto interno soddisfa la seguente proprietà:

$$(\sum_i a_i \left|w_i\right\rangle, \left|v\right\rangle) = \sum_i a_i^* (\left|w_i\right\rangle, \left|v\right\rangle).$$

In notazione di Dirac, il prodotto interno del vettore $\left|v\right\rangle$ con il vettore $\left|w\right\rangle$è denotato da $\left\langle v\right| \left.w\right\rangle$. Mediante il prodotto interno si può definire la *norma* di un vettore come

$$\|v\| = \sqrt{\left\langle v\right|}.$$

Uno *spazio di Hilbert*, è uno spazio vettoriale V dotato di prodotto interno e completo rispetto alla metrica indotta dalla norma $\|\cdot\|$. Per completezza si intende che tutte le sequenze di Cauchy di vettori in V convergono a un limite in V. Questa proprietà è significativa nel caso di spazi a dimensioni infinite, poichè per spazi vettoriali di dimensioni finite è sempre soddisfatta. In computazione quantistica, gli spazi vettoriali con cui si ha a che fare sono sempre di dimensioni finite. Pertanto, per i nostri scopi, il termine "spazio di Hilbert" sarà del tutto equivalente a "spazio vettoriale con prodotto interno". Inoltre, ci riferiremo di solito ad uno spazio vettoriale V, intendendo implicitamente che Vè uno spazio di Hilbert.

8.2 Basi ortonormali

Un vettore $\left|v\right\rangle$ in uno spazio vettoriale V si dice *vettore unitario* o *normalizzato* se la sua normaè 1, cioè $\|v\| = 1$. I vettori $\left|v\right\rangle$ e $\left|w\right\rangle$ si dicono *ortogonali* se il loro prodotto internoè zero, cioè $\left\langle v\right| \left.w\right\rangle = 0$. Un insieme di vettori $\{\left|i\right\rangle\}$ con indice i si dice *ortonormale* se per ogni i, $\{\left|i\right\rangle\}$ è un vettore unitario e vettori distinti sono a due a due ortogonali, cioè

$$\left\langle i\right| \left.j\right\rangle = \delta_{ij} = \{ \begin{array}{ll} 1 & i = j \\ 0 & i \neq j \end{array}$$

È sempre possile trasformare una qualsiasi base per uno spazio vettoriale V in una base ortonormale. Il metodo per farlo si chiama *procedura di*

Gram-Schmidt e non lo esamineremo in dettaglio. Grazie a questa procedura possiamo assumere che le basi che considereremo d'ora in poi siano sempre ortonormali.

8.3 Operatori lineari e prodotto esterno

Una rappresentazione utile di un operatore lineare è quella mediante *prodotto esterno*. Dati $|v\rangle \in V$ e $|w\rangle \in W$, definiamo l'operatore lineare $|w\rangle\langle v| : V \mapsto W$ associa ad ogni vettore $|v'\rangle \in V$ il vettore in W risultante dalla moltiplicazione scalare di w per il prodotto interno $\langle v|v'\rangle$, cioè

$$|w\rangle\langle v|\left(|v'\rangle\right) \equiv |w\rangle\langle v|v'\rangle = \langle v|v'\rangle|w\rangle .$$

Ogni base ortonormale $\{|i\rangle\}$ per uno spazio vettoriale V soddisfa la *relazione di completezza*

$$\sum_i |i\rangle\langle i| = I.$$

Infatti, per ogni $|v\rangle \in V$, $|v\rangle = \sum_i v_i|i\rangle$, si ha che

$$\left(\sum_i |i\rangle\langle i|\right)|v\rangle = \sum_i |i\rangle\langle i|v\rangle = \sum_i v_i|i\rangle = |v\rangle ,$$

poichè $\langle i|v\rangle = v_i$ per ogni i.
Utilizzando la relazione di completezza si ottiene la seguente rappresentazione per operatori lineari che sarà molto utile nel seguito.
Dato un operatore lineare $L : V \mapsto W$ e date le basi ortonormali $\{|v_i\rangle\}$ per V e $\{|w_j\rangle\}$ per W, possiamo scrivere L come

$$L = I_W L I_V,$$

dove I_V e I_W sono rispettivamente l'identità in V e in W. Da questa espressione si ottiene la rappresentazione in forma di prodotto esterno di L. Infatti,

$$I_W L I_V = \sum_{ij} |w_j\rangle\langle w_j|L|v_i\rangle\langle v_i| = \sum_{ij} \langle w_j|L|v_i\rangle |w_j\rangle\langle v_i| .$$

Nota che i coefficienti $\langle w_j|L|v_i\rangle$ corrispondono agli elementi della matrice di rappresentazione di L rispetto alla base di input $\{|v_i\rangle\}$ e alla base di output $\{|w_j\rangle\}$. In particolare, $L|v_i\rangle$ è la trasformazione dell'i-esimo vettore della base di input in un vettore in W la cui coordinata j-esima è data da $\langle w_j|L|v_i\rangle$. Tale elemento è pertanto l'elemento della colonna i e della riga j della matrice per L.

8.4 Autovalori e autovettori

Un *autovettore* di un operatore lineare L su uno spazio vettoriale V è un vettore non nullo $|v\rangle \in V$ tale che $L|v\rangle = v|v\rangle$, dove v è un numero complesso detto l'*autovalore* di L corrispondente a $|v\rangle$. Gli autovalori di L sono le soluzioni dell'equazione caratteristica $c(\lambda) = 0$, dove $c(\lambda) \equiv det|L - \lambda I|$, è il determinante della matrice $L - \lambda I$. L'*autospazio* corrispondente ad un autovalore v è il sottospazio di V formato da tutti i vettori che hanno autovalore v. Una *rappresentazione diagonale* per un operatore L è $L = \sum_i \lambda_i |i\rangle\langle i|$, dove i vettori $|i\rangle$ formano un insieme ortonormale di autovettori per L con autovalori λ_i. Un operatore si dice diagonalizzabile se ammette una rappresentazione diaganale.

Esercizio 8.2 *Trovare autovettori, autovalori e rappresentazione diagonale delle matrici di Pauli*

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}\ e\ Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

8.5 Operatori aggiunti e Hermitiani

Dato un operatore lineare L su uno spazio di Hilbert V, esiste un unico operatore lineare $L^\dagger$ tale che per tutti i vettori $|v\rangle, |w\rangle \in V$,

$$(|v\rangle, L|w\rangle) = (L^\dagger|v\rangle, |w\rangle).$$

L'operatore $L^\dagger$ si chiama l'aggiunto di L. Per convenzione, $|v\rangle^\dagger \equiv \langle v|$.

Dalla definizione seguono le seguenti proprietà:

- $(\sum_i a_i L_i)^\dagger = \sum_i a_i^* L_i^\dagger$.
- $(L^\dagger)^\dagger = L$.

In rappresentazione matriciale, l'operatore aggiunto corrisponde alla matrice trasposta coniugata di L: $L^\dagger = (L^*)^T$.

Un operatore *Hermitiano* è un operatore L tale che $L^\dagger = L$.

Gli operatori di *proiezione* sono una classe importante di operatori Hermitiani, definiti nel seguente modo. Dato uno spazio vettoriale V di dimensione d, si considera un sottospazio di V di dimensione k. Allora, se $\{|1\rangle, \ldots, |d\rangle\}$ è una base ortonormale per V, il sottoinsieme $\{|1\rangle, \ldots, |k\rangle\}$ è una base ortonormale per W. Si definisce quindi

$$P \equiv \sum_{i=1}^{k} |i\rangle\langle i|$$

come la proiezione sul sottospazio W. Gli operatori di proiezione sono idempotenti: $P^2 = P$.

Un'altra classe importante di operatori lineari è quella degli operatori *normali*. L è normale se e solo se $L^\dagger L = LL^\dagger$. Il teorema della *scomposizione spettrale* stabilisce che un operatore è normale se e solo se è *diagonalizzabile*, cioè può essere scritto nella forma $M = \sum_i \lambda_i |i\rangle \langle i|$ dove λ_i sono gli autovalori di M e $\{|i\rangle\}$ forma una base ortonormale per V.

Una matrice Hermitiana è ovviamente normale. Si dimostra che una matrice Hermitiana è normale se e solo se ha autovalori reali.

Una matrice (o equivalentemente un operatore) U si dice *unitaria* se $U^\dagger U = I$. Le matrici unitarie godono di un'importante proprietà, cioè preservano i prodotti interni. Infatti, se U è unitaria, allora dati due vettori $|v\rangle$ e $|w\rangle$

$$(U|v\rangle, U|w\rangle) = \langle v| U^\dagger U |w\rangle = \langle v| I |w\rangle = \langle v, w\rangle.$$

Da questo risultato segue che la rappresentazione di un operatore unitario U rispetto a due basi ortonormali di input e output $\{|v_i\rangle\}_i$ e $\{|w_i\rangle\}_i$ si può dare in forma di prodotti esterni come:

$$U = \sum_i |w_i\rangle \langle v_i| .$$

Esercizio 8.3 *Mostrare che gli autovalori di una matrice unitaria hanno tutti modulo* 1 *e quindi possono essere scritti nella forma* $e^{i\theta}$ *con* $\theta \in \mathbb{R}$.

Esercizio 8.4 *Mostrare che le matrici di Pauli sono Hermitiane e unitarie.*

Introduzione agli algoritmi quantistici

9 Computazioni classiche

Una differenza fondamentale tra i circuiti classici e quelli quantistici è che le porte logiche classiche potrebbero essere irreversibili (ad esempio `AND`, `XOR`, `NAND`), mentre le porte logiche quantistiche sono sempre unitarie e quindi reversibili. D'altra parte, sarebbe auspicabile che un modello di computazione alternativo fosse in grado di esprimere almeno tutte le computazioni esprimibili con il modello classico. Il nostro primo obiettivo è quindi quello di rappresentare le computazioni classiche come trasformazioni unitarie, cioè come computazioni quantistiche. Poichè le trasformazioni unitarie sono invertibili (cioè reversibili), il primo passo da fare è quello di trasformare ogni computazione classica irreversibile in una reversibile. Per poter operare in

IN			OUT		
a	b	c	a'	b'	c'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Table 1: Tabella di verità della porta di Toffoli

modo reversibile è necessario che la funzione da valutare sia una biezione (i.e. iniettiva e surgettiva). In tal caso possiamo infatti risalire in maniera univoca da ogni output al valore dell'input che l'aveva generato, cioè operare in modo inverso. Una qualsiasi computazione irreversibile può essere trasformata in una equivalente computazione reversibile rendendo biunivoca la corrispondente funzione da valutare. Ad esempio, data una qualsiasi funzione:

$$f : \{0,1\}^k \mapsto \{0,1\}^m,$$

possiamo costruire $\tilde{f} : \{0,1\}^{k+m} \mapsto \{0,1\}^{k+m}$, tale che $\tilde{f}$ è biunivoca e calcola $(x, f(x))$ agendo sull'input $(x, 0^m)$, dove 0^m denota m bits inizializzati a 0. Ogni funzione biunivoca $f : \{0,1\}^n \mapsto \{0,1\}^n$, si può in realtà vedere come una permutazione sugli n bits in input o, equivalentemente, sugli interi $0, 1, \ldots, 2^n - 1$. Di conseguenza, essa descrive una computazione classica reversibile.

Una qualsiasi computazione classica irreversibile si può trasformare in una computazione equivalente ma reversibile usando la *porta di Toffoli*. Questa è un'operazione classica reversibile, rappresentata dal circuito in Figura 15, che opera su tre bits in input: due sono bits di controllo e il terzo è il bit target che viene scambiato se i bits di controllo sono entrambi 1, come mostrato nella Tabella 1.

La reversibilità di questa operazione si verifica facilmente osservando che applicando per due volte consecutive la porta di Toffoli si ottiene lo stesso risultato di partenza:

$$(a, b, c) \to (a, b, c \oplus ab) \to (a, b, c).$$

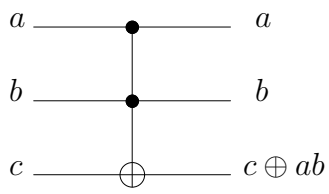

Figure 15: Rappresentazione della porta di Toffoli

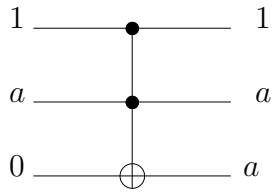

Figure 16: `FANOUT` realizzato mediante la porta di Toffoli

Quindi l'operazione stessa coincide con la sua inversa. Si verifica altrettanto facilmente che la porta di Toffoli rappresenta la permutazione $\pi = (67)$ sugli interi $0, 1, \ldots, 7$ (scambia la due sequenze 110 e 111).

La porta di Toffoli è universale per le computazioni classiche reversibili, cioè ogni computazione classica si può costruire in modo reversibile mediante la porta di Toffoli. Questo risultato segue dall'universalità delle operazioni di `NAND` e `FANOUT` (l'operazione di copia di un bit classico) per le computazioni classiche e dal fatto che entrambe queste operazioni si possono esprimere mediante il circuito di Toffoli. Infatti, applicando l'operazione con $c = 1$, otteniamo $a' = a$, $b' = b$ e $c' = 1 \oplus ab = \neg ab$, cioè la simulazione di `NAND` come operazione reversibile. Il `FANOUT` reversibile si ottiene invece come mostrato in Figura 16: applicando la porta di Toffoli con $a = 1$ e $c = 0$ si ottiene come risultato la copia del bit b (Ricordiamo che questa operazione di copia non è possibile per un qubit).

Cosi come per `NAND` e `FANOUT` la costruzione di un circuito reversibile per una qualsiasi operazione classica f mediante la porta di Toffoli comporta l'utilizzo di alcuni bits di servizio in input (o *ancilla bits*) e in output (o *garbage*). Dopo aver eliminato questi bits di servizio, il circuito risultante esegue la trasformazione:

$$(x, y) \mapsto (x, y \oplus f(x)),$$

(dove x è l'input di f e y è il registro destinato a contenere l'output) e può essere considerato come il *circuito reversibile standard* per la valutazione di f.

9.1 Computazioni classiche su circuiti quantistici

Come abbiamo già osservato, una computazione classica reversibile corrisponde ad una permutazione sulle sequenze dei bit in input. Questo garantisce la possibilità di costruire una matrice unitaria complessa che la rappresenta[3]. In particolare la porta di Toffoli può essere implementata come circuito quantistico. In questo caso l'input è dato da tre qubits e la trasformazione, analogamente al caso classico, consiste nello scambio del terzo qubit se i primi due sono 1. Ad esempio la porta di Toffoli quantistica applicata allo stato $|110\rangle$ produce lo stato $|111\rangle$. Un semplice esercizio è scrivere la matrice unitaria U corrispondente a questa permutazione.

La porta di Toffoli quantistica si può quindi usare per simulare su un computer quantistico tutte le computazioni classiche, assicurando che un computer quantistico è in grado di eseguire una qualsiasi computazione eseguibile su un computer classico.

9.2 Computazioni probabilistiche su circuiti quantistici

Gli algoritmi *randomizzati* sono algoritmi che vengono eseguiti usando un generatore di numeri casuali (il lancio di una moneta) per decidere uno dei possibili rami di esecuzione. Il primo algoritmo randomizzato venne introdotto da Solovay e Strassen negli anni '70 per determinare se un numero è primo oppure no. L'algoritmo produce una risposta corretta solo con una certa probabilità. Tale probabilità si può aumentare ripetendo l'esecuzione per un opportuno numero di volte.

Anche questi algoritmi possono essere simulati efficientemente mediante circuiti quantistici. Infatti, per simulare un bit random è sufficiente preparare un qubit nello stato $|0\rangle$ e applicare poi la porta di Hadamard. Si otterrà lo stato $(|0\rangle + |1\rangle)/\sqrt{2}$ che misurato darà 0 o 1 ciascuno con probabilità 1/2. Bisogna osservare inoltre che in questo modo si ottiene un numero "realmente" random, cosa che un computer classico non può fare.

[3]Un risultato della teoria dei gruppi e rappresentazioni assicura che esiste una rappresentazione, chiamata *rappresentazione unitaria standard*, del gruppo simmetrico delle permutazioni su 2^n simboli nel gruppo delle matrici unitarie complesse $2^n \times 2^n$. Tale rappresentazione associa ad una permutazione π la matrice U di elemento generico $U_{ij} = \delta_{i,\pi(j)}$, dove δ_{kl} denota il delta di Kronecker definito da $\delta_{kl} = 1$ se $k = l$ e $\delta_{kl} = 0$ altrimenti.

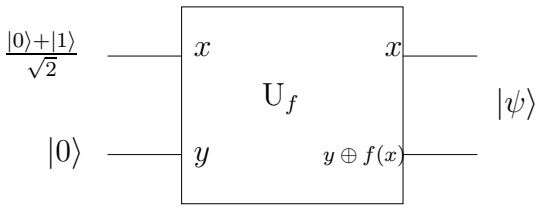

Figure 17: Circuito quantistico per valutare $f(0)$ e $f(1)$ simultaneamente

10 Parallelismo quantistico

Su un computer quantistico si può valutare una funzione $f(x)$ su valori differenti di x contemporaneamente. Questo è noto come *parallelismo quantistico* e rappresenta una caratteristica fondamentale dei circuiti quantistici.

Consideriamo una funzione booleana della forma:

$$f(x) : \{0,1\} \mapsto \{0,1\}.$$

Per calcolare $f(x)$ mediante una computazione quantistica si deve definire la trasformazione $f(x)$ come una trasformazione unitaria U_f. Come visto precedentemente, questo si può fare applicando sullo stato di input $|x,y\rangle$, detto registro dei dati, un'appropriata sequenza di porte logiche quantistiche (che indicheremo con una scatola nera chiamata U_f) che trasformano $|x,y\rangle$ nello stato $|x, y \oplus f(x)\rangle$, detto registro target. Se $y = 0$ allora lo stato finale del secondo qubit conterrà esattamente il valore di $f(x)$.

Nel circuito in Figura 17, l'input è $\frac{|0\rangle+|1\rangle}{\sqrt{2}} \otimes |0\rangle$, cioè il valore di x è una sovrapposizione di 0 e 1 che si può ottenere applicando Hadamard a $|0\rangle$. Applicando U_f a questo registro di dati si ottiene

$$\frac{|0, f(0)\rangle + |1, f(1)\rangle}{\sqrt{2}}.$$

Questo stato contiene informazioni sia sul valore $f(0)$ che sul valore $f(1)$. Abbiamo quindi valutato f simultaneamente su $x = 0$ e $x = 1$. Questo tipo di parallelismo è diverso da quello classico dove più circuiti (ognuno dei quali calcola $f(x)$ per un singolo valore di x) vengono eseguiti contemporaneamente.

Si può generalizzare questa procedura per calcolare funzioni su un numero arbitrario di bits usando una generalizzazione della porta di Hadamard nota come la *trasformata di Walsh-Hadamard*. Questa operazione consiste in n porte di Hadamard che agiscono in parallelo su n qubits. Per esempio,

per $n = 2$, la trasformata di Walsh-Hadamard si indica con $\mathrm{H}^{\otimes 2} = H \otimes H$ e applicata a due qubits preparati nello stato $|0\rangle$ dà come risultato

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2}.$$

In generale, il risultato di $\mathrm{H}^{\otimes n}$ applicato a n qubits nello stato $|0\rangle$ è:

$$\frac{1}{\sqrt{2^n}} \sum_x |x\rangle ,$$

dove x è la rappresentazione binaria dei numeri da 0 a $2^n - 1$ Quindi la trasformata di Walsh-Hadamard produce una sovrapposizione equiprobabile di tutti gli stati della base computazionale di n qubits. Osserviamo che per ottenere una sovrapposizione di 2^n stati servono soltanto n porte logiche H.

Esercizio 10.1 *Verifica che*

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

Esercizio 10.2 *Verifica che*

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

dove $x \in \{0,1\}^n$ *e* $x \cdot y$ *è* $\sum_{j=1}^n x_i y_i$ *modulo* 2.

La valutazione parallela di una funzione, $f(x)$ con input x di n bits e 1 bit come output, può quindi essere eseguita da un circuito simile a quello in Figura 17, con $n + 1$ qubits in input preparati nello stato $|0\rangle^{\otimes n} |0\rangle$. Si applica quindi Hadamard ai primi n qubits e successivamente il circuito U_f. Il risultato sarà

$$\frac{1}{\sqrt{2^n}} \sum_x |x\rangle |f(x)\rangle .$$

Il parallelismo quantistico non è direttamente utilizzabile nel senso che non è possibile ottenere tutti i valori calcolati con una sola misurazione: la misurazione dello stato $\frac{1}{\sqrt{2^n}} \sum_x |x\rangle |f(x)\rangle$ darà il valore di $f(x)$ per un *singolo* valore di x. Per sfruttare l'informazione nascosta in questo parallelismo

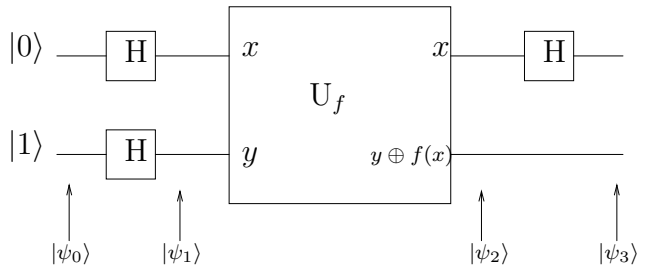

Figure 18: Circuito quantistico che implementa l'algoritmo di Deutsch

dobbiamo in qualche modo sfruttare meglio l'informazione contenuta nella sovrapposizione $\frac{1}{\sqrt{2^n}} \sum_x |x\rangle |f(x)\rangle$, ad esempio utilizzando in maniera opportuna l'*interferenza* tra gli stati nella sovrapposizione. Combinando il parallelismo quantistico con questa proprietà che viene dalla meccanica quantistica si possono ottenere risultati come quello esemplificato dall'algoritmo di Deutsch.

10.1 Algoritmo di Deutsch

L'algoritmo di Deutsch mostra come attraverso la valutazione parallela di una funzione su tutti i suoi inputs si possano determinare proprietà globali della funzione come, per esempio, quella di essere una funzione *costante* o *bilanciata*.

Considera il circuito in Figura 18.

L'input del circuito che calcola la funzione f sono ora i qubits risultanti dall'applicazione di Hadamard agli stati $|0\rangle$ e $|1\rangle$. Tale input è quindi

$$|\psi_1\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

Applicando U_f allo stato $|x\rangle (|0\rangle - |1\rangle)/\sqrt{2}$ si ottiene $(-1)^{f(x)} |x\rangle (|0\rangle - |1\rangle)/\sqrt{2}$. Infatti, poichè U_f non modifica $|x\rangle$, se $f(x) = 0$ allora il risultato è $|x\rangle (|0\rangle - |1\rangle)/\sqrt{2}$, mentre se $f(x) = 1$ il risultato è $|x\rangle (|1\rangle - |0\rangle)/\sqrt{2}$.

Quindi, applicando U_f a $|\psi_1\rangle$ otteniamo un risultato $|\psi_2\rangle$ che dipende da due possibilità:

$$\begin{array}{ll} \text{se } f(0) = f(1) & \pm \left[\frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] \\ \text{se } f(0) \neq f(1) & \pm \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right]. \end{array}$$

Nota che nella seconda alternativa, si ha che $(-1)^{f(1)} = -(-1)^{f(0)}$. Nota inoltre che $|\psi_1\rangle$ si può scrivere come

$$\frac{1}{\sqrt{2}}\left(|0\rangle\,\frac{|0\rangle - |1\rangle}{\sqrt{2}} + |1\rangle\,\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right).$$

Pertanto U_f applicato $|\psi_1\rangle$ si può scrivere come

$$\frac{1}{\sqrt{2}}\left((-1)^{f(0)}\,|0\rangle\,\frac{|0\rangle - |1\rangle}{\sqrt{2}} + (-1)^{f(1)}\,|1\rangle\,\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right),$$

o, equivalentemente

$$\frac{1}{\sqrt{2}}\left((-1)^{f(0)}\,|0\rangle\,\frac{|0\rangle - |1\rangle}{\sqrt{2}} - (-1)^{f(0)}\,|1\rangle\,\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right),$$

cioè

$$\pm\frac{1}{\sqrt{2}}\left(|0\rangle\,\frac{|0\rangle - |1\rangle}{\sqrt{2}} - |1\rangle\,\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right),$$

ovvero

$$\pm\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}\right].$$

Applichiamo ora Hadamard al primo qubit e otteniamo $|\psi_3\rangle$ che risulta

$$\begin{array}{ll} \text{se } f(0) = f(1) & \pm\,|0\rangle\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \\ \text{se } f(0) \neq f(1) & \pm\,|1\rangle\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]. \end{array}$$

A questo punto osserviamo che $f(0) \oplus f(1) = 0$ se $f(0) = f(1)$, altrimenti $f(0) \oplus f(1) = 1$. Possiamo quindi scrivere il risultato in maniera più coincisa come

$$|\psi_3\rangle = \pm\,|f(0) \oplus f(1)\rangle\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right].$$

Attraverso una misurazione del primo qubit possiamo quindi determinare con certezza (la probabilità associata al primo qubit è 1) il valore di $f(0) \oplus f(1)$ e quindi se la funzione f è *costante* oppure *bilanciata*. Per far questo abbiamo dovuto valutare f una volta sola.

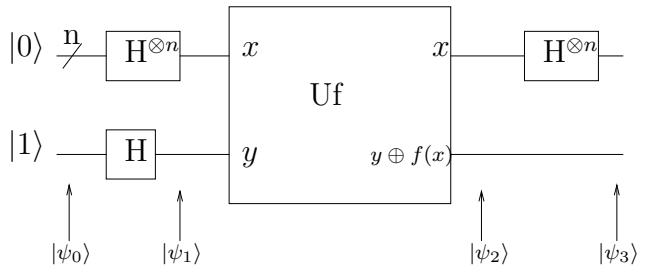

Figure 19: Circuito quantistico che implementa l'algoritmo di Deutsch-Jozsa

10.2 Algoritmo di Deutsch-Jozsa

L'algoritmo precedente si può estendere a funzioni booleane su n bits. Consideriamo una funzione $f : \{0,1\}^n \mapsto \{0,1\}$ e supponiamo di sapere che f può essere o costante oppure *bilanciata* (cioè assume valore 0 su esattamente metà degli inputs e valore 1 sulla restante metà). Usando un algoritmo classico, nel caso peggiore abbiamo bisogno di valutare la funzione su almeno $2^{n-1}+1$ valori per poter essere in grado di stabilire con certezza se f è costante o bilanciata. L'algoritmo quantistico di Deutsch-Jozsa ci permette di stabilirlo in un solo passo. Il circuito quantistico che implementa questo algoritmo è dato in Figura 19. A differenza del circuito in Figura 18, l'input x della funzione è dato da n qubits preparati nello stato $|0\rangle$, che chiameremo il registro dei dati. Il qubit target, destinato a contenere il risultato di $f(x)$, è invece preparato nello stato $|1\rangle$. Il registro iniziale è quindi

$$|\psi_0\rangle = |0\rangle^{\otimes n} |1\rangle ,$$

dove $|0\rangle^{\otimes n}$ indica il prodotto tensore di n qubits $|0\rangle$.

Al registro dei dati $|0\rangle^{\otimes n}$ viene applicata la trasformazione di Walsh-Hadamard, $\mathrm{H}^{\otimes n}$, per produrre una sovrapposizione equiprobabile di tutti i 2^n stati della base computazionale.

Indicando con $x, y \in \{0,1\}^n$ sequenze di bits di lunghezza n, si dimostra che

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

dove $x \cdot y$ è $\sum_{j=1}^{n} x_i y_i$ modulo 2 (cf. Esercizio 10.2). In particolare, risulta

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

Gli stati $|\psi_1\rangle, |\psi_2\rangle, |\psi_3\rangle$ nel circuito in figura risultano quindi:

$$\begin{aligned}
|\psi_1\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \\
|\psi_2\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} |x\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \\
|\psi_3\rangle &= \sum_{y\in\{0,1\}^n} \sum_{x\in\{0,1\}^n} \frac{(-1)^{x\cdot y + f(x)} |y\rangle}{2^n} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right].
\end{aligned}$$

L'ampiezza di $|y\rangle = |0\rangle^{\otimes n}$ in $|\psi_3\rangle$ è $\sum_x (-1)^{f(x)}/2^n$. Consideriamo i due casi. Se f è costante, tale valore risulta 1 o -1. Questo significa (per la condizione di normalizzazione) che le ampiezze associate a tutti gli altri stati sono 0. Una misurazione del registro dei dati produce lo stato $|0\rangle^{\otimes n}$ con probabilità 1. Se invece f è bilanciata, l'ampiezza di $|y\rangle = |0\rangle^{\otimes n}$ risulta 0. Ciò significa che il risultato di una misurazione deve essere diverso da zero su almeno un qubit del registro dei dati. Dopo la misurazione siamo quindi in grado di stabilire con certezza se la funzione è costante o bilanciata.

Abbiamo ottenuto quindi una riduzione esponenziale della complessità rispetto ad un qualsiasi algoritmo classico.

Il Modello dei Circuiti per la computazione quantistica

11 Modelli di computazione classica e universalità

L'esigenza di formalizzare l'idea intuitiva di funzione *calcolabile* tramite un algoritmo (o *funzione algoritmica*) diede origine nella prima metà del ventesimo secolo alla teoria della calcolabilità che nasce con le varie proposte di *modelli di calcolo* da parte di Kleene, Church e Turing (1936). Negli anni successivi furono proposte altre caratterizzazioni formali, tra cui ad esempio quelle di Post (1943) e Markov (1954). La definizione formale di algoritmo dipende ovviamente dal modello di calcolo adottato e può avere quindi formulazioni molto diverse l'una dall'altra.

Un risultato fondamentale della teoria della calcolabilità afferma che tutte queste diverse caratterizzazioni sono equivalenti: **ogni funzione calcolabile in un certo modello di calcolo formale, è calcolabile anche negli altri**, purché opportunamente codificata in accordo ai diversi formalismi. In particolare, la *Tesi di Church-Turing* afferma che ogni funzione calcolabile si può calcolare nel formalismo di Turing, basandosi sull'evidenza che non esiste un concetto di calcolabilità più ampio di quello di Turing.

Esamineremo successivamente più in dettaglio il formalismo di Turing e il concetto di universalità associato, cioè la Macchina di Turing Universale, mentre ci concentreremo ora su un altro modello di calcolo che risulta in alcuni casi più utile come modello di riferimento per lo studio della complessità computazionale. Questo modello computazionale alternativo per la computazione classica è basato sui *circuiti booleani*. In questo caso l'universalità del modello si esprime individuando un insieme di circuiti elementari mediante i quali è possibile esprimere qualsiasi altro circuito. Tale insieme si dice una *base completa*. Come è noto i circuiti elementari che calcolano le funzioni booleane NOT, OR e AND formano una base completa per la computazione classica [1].

L'obiettivo di questa lezione è di definire in analogia con il caso classico un modello astratto per la computazione quantistica e far vedere che, come per la computazione classica, esiste anche per la computazione quantistica una nozione di universalità.

12 Circuiti quantistici: realizzazione esatta

Il modello di computazione basato sui circuiti quantistici è stato introdotto da Deutsch (1989) e sviluppato ulteriormente da Yao (1993). Nel suo articolo, Yao dimostra anche l'equivalenza con un altro modello per la computazione quantistica: la Macchina di Turing quantistica. Quest'ultima è stata introdotta da Benioff (1980) e successivamente sviluppata da Deutsch (1985) e Yao (1993). L'articolo di Bernstein e Vazirani *Quantum Complexity Theory* (1997) contiene una descrizione completa e moderna della macchina di Turing quantistica. In questo articolo viene affrontato il tema della complessità computazionale dimostrando per la computazione quantistica gli analoghi dei risultati della complessità computazionale classica.

Questa lezione è dedicata al modello dei circuiti, mentre si consiglia fortemente la lettura dell'articolo di Bernstein e Vazirani [2] per una descrizione del modello basato sulla Macchina di Turing quantistica.

Dal momento che si possono definire infiniti operatori unitari (e quindi infinite operazioni quantistiche), la questione che si pone è come definire una base completa per la computazione quantistica. Ci sono due possibilità:

- considerare un insieme infinito di gates,
- considerare un insieme finito e sostituire la condizione di realizzazione esatta dalla condizione di realizzazione *approssimata*.

Esamineremo nel seguito la prima possibilità, mentre la seconda sarà discussa in 13.

12.1 Operazioni su un qubit

Le operazioni quantistiche più elementari sono quelle su un singolo qubit. Ricordiamo che un qubit è un vettore $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$ nello spazio di Hilbert $\mathbb{C}^2$, tale che $|a|^2 + |b|^2 = 1$. Le operazioni su un qubit devono preservare questa norma e sono pertanto descritte da matrici unitarie 2×2. Tra queste abbiamo visto le matrici di Pauli:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \text{ e } Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

spesso denotate anche con σ_x, σ_y e σ_z. Un ruolo molto importante nella definizione del modello dei circuiti è svolto anche dalle seguenti operazioni:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},\ S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \text{ e } T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}.$$

Abbiamo già visto la porta di Hadamard H. Il gate S è detto gate di fase, mentre il gate T è anche chiamato gate $\pi/8$ perché risulta

$$T = e^{i\pi/8} \begin{bmatrix} e^{-i\pi/8} & 0 \\ 0 & e^{i\pi/8} \end{bmatrix}.$$

Si verifica facilmente che $H = (X + Z)/\sqrt{2}$, $S = T^2$ e $S = \sqrt{Z}$.

Ricordiamo inoltre che lo stato di un qubit $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$ può essere visualizzato come un punto (θ, φ) sulla sfera unitaria in $\mathbb{R}^3$ (la sfera di Bloch) dato che esiste una corrispondenza biunivoca tra qubits e questi punti. Analogamente, le operazioni su un qubit (cioè le matrici unitarie 2×2) corrispondono alle *rotazioni* del punto che rappresenta il qubit sulla sfera di Bloch. In generale il teorema di Bloch (cf. Teorema 12.4) assicura che qualsiasi matrice unitaria 2×2 si può decomporre in un prodotto di rotazioni nel piano $(\vec{x}, \vec{y})$ e rotazioni intorno all'asse $\vec{z}$. Le prime sono descritte da matrici della forma

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix},$$

mentre, come vedremo, le seconde corrispondono a operazioni unitarie della forma

$$\begin{bmatrix} e^{-i\beta/2} & 0 \\ 0 & e^{i\beta/2} \end{bmatrix}.$$

Per poter enunciare questo teorema abbiamo bisogno di introdurre la nozione di *esponenziale di una matrice* e la definizione di *operatore di rotazione* intorno agli assi $\vec{x}$, $\vec{y}$ e $\vec{z}$.

12.1.1 Esponenziale di una matrice

In generale, l'esponenziale e^A di una matrice complessa $n \times n$, A, è definita mediante la serie di potenze

$$e^A = \sum_{m=0}^{\infty} \frac{A^m}{m!}.$$

Si dimostra che questa serie converge e che e^A è una funzione continua di A. Si dimostrano inoltre le seguenti proprietà:

1. $e^0 = I$
2. e^A è invertibile e $(e^A)^{-1} = e^{-A}$

3. $e^{(a+b)A} = e^{aA}e^{bA}$, per ogni $a, b \in \mathbb{C}$

4. se $AB = BA$, allora $e^{A+B} = e^A e^B = e^B e^A$

5. se C è invertibile, allora $e^{CAC^{-1}} = Ce^A C^{-1}$

6. $\|e^A\| \leq e^{\|A\|}$.

Se A è diagonalizzabile possiamo calcolare esplicitamente e^A nel seguente modo[4]. Poichè A è diagonalizzabile esiste una matrice invertibile C tale che $A = CDC^{-1}$, dove

$$D = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix}.$$

Osserviamo che e^A è la matrice diagonale con autovalori e^{λ_i}; quindi per la proprietà 5, otteniamo

$$e^A = C \begin{bmatrix} e^{\lambda_1} & & 0 \\ & \ddots & \\ 0 & & e^{\lambda_n} \end{bmatrix} C^{-1}.$$

Esempio 12.1 *Consideriamo la matrice*

$$A = \begin{bmatrix} 0 & -a \\ a & 0 \end{bmatrix}.$$

A ha autovettori $\begin{bmatrix} 1 \\ i \end{bmatrix}$ *e* $\begin{bmatrix} i \\ 1 \end{bmatrix}$ *con corrispondenti autovalori $-ia$ e ia. Quindi la matrice invertibile* $C = \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix}$ *trasforma i vettori della base* $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ *e* $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ *negli autovettori di A. Si verifica facilmente che la matrice $D = C^{-1}AC$ che rappresenta A nella nuova base è una matrice diagonale. Dalla relazione $A = CDC^{-1}$ e la proprietà (5) dell'esponenziale di una matrice si ottiene:*

$$e^A = \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix} \begin{bmatrix} e^{-ia} & 0 \\ 0 & e^{ia} \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{-i}{2} \\ \frac{-i}{2} & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \cos a & -\sin a \\ \sin a & \cos a \end{bmatrix}.$$

[4]Nota che le nostre operazioni su qubits sono matrici unitarie e quindi in particolare operatori normali e ammettono pertanto una scomposizione spettrale, cioè sono diagonalizzabili.

12.1.2 Operatori di rotazione

Gli esponenziali delle matrici di Pauli danno luogo a tre utili classi di matrici unitarie: gli *operatori di rotazione* intorno agli assi $\vec{x}$, $\vec{y}$ e $\vec{z}$ definiti come:

$$R_x(\theta) \equiv e^{-i\theta\sigma_x/2} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}\sigma_x = \begin{bmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{bmatrix} \quad (2)$$

$$R_y(\theta) \equiv e^{-i\theta\sigma_y/2} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}\sigma_y = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix} \quad (3)$$

$$R_z(\theta) \equiv e^{-i\theta\sigma_z/2} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}\sigma_z = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}. \quad (4)$$

Esercizio 12.2 *Verifica che $XYX = -Y$ e $XZX = -Z$. Quindi dimostra che $XR_y(\theta)X = R_y(-\theta)$ e $XR_z(\theta)X = R_z(-\theta)$.*

Possiamo usare il procedimento visto nell'Esempio 12.1 per calcolare esplicitamente l'esponenziale delle matrici di Pauli e verificare le equazioni (2), (3) e (4). Per esempio, per calcolare l'esponenziale $e^{-i\theta\sigma_z/2}$, ricordiamo che σ_z ha autovettori $|0\rangle$ e $|1\rangle$ con corrispondenti autovalori 1 e -1, quindi in questo caso $C = I$ e pertanto

$$e^{-i\theta\sigma_z/2} = I \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix} I.$$

Analogamente, si calcolano gli esponenziali per σ_x e σ_y ricordando che gli autovettori sono:

$$|x_1\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \text{ e } |x_2\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \text{ per } \sigma_x,$$

$$|y_1\rangle = \frac{|0\rangle + i\,|1\rangle}{\sqrt{2}} \text{ e } |y_2\rangle = \frac{|0\rangle - i\,|1\rangle}{\sqrt{2}} \text{ per } \sigma_y,$$

corrispondenti rispettivamente agli autovalori 1 e -1. Si noti che in questo caso la matrice del cambio di base C è $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ per σ_x, e $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix}$ per σ_y.

Un modo alternativo per verificare le equazioni (2), (3) e (4) è usare la seguente proprietà che è una diretta conseguenza della definizione di esponenziale come serie di potenze data precedentemente .

Proposition 12.3 *Dato $x \in \mathbb{R}$ e A una matrice tale che $A^2 = I$, allora*

$$e^{iAx} = \cos x I + i \sin x A.$$

Dimostrazione Per definizione

$$e^{iAx} = \sum_{m=0}^{\infty} \frac{(ix)^m}{m!} A^m.$$

Dall'ipotesi $A^2 = I$, si ha quindi, separando le somme sugli indici pari e dispari:

$$\begin{aligned} e^{iAx} &= \sum_{m=0}^{\infty} \frac{(ix)^{2m}}{(2m)!} I + \sum_{m=0}^{\infty} \frac{(ix)^{2m+1}}{(2m+1)!} A \\ &= \sum_{m=0}^{\infty} \frac{(-1)^m x^{2m}}{(2m)!} I + i \sum_{m=0}^{\infty} \frac{(-1)^m x^{2m+1}}{(2m+1)!} A \end{aligned}$$

Osserviamo ora che le due somme sono lo sviluppo in serie di potenze per $\cos x$ e $\sin x$ rispettivamente. □

Le equazioni (2), (3) e (4) si ottengono direttamente dalla Proposizione 12.3 notando che per le matrici di Pauli vale la condizione della proposizione, cioè $\sigma_x^2 = \sigma_y^2 = \sigma_z^2 = I$.

Se $\hat{n} = (n_x, n_y, n_z)$ è un vettore unitario nello spazio $\mathbb{R}^3$, allora possiamo generalizzare le definizioni di R_x, R_y e R_z per definire una rotazione di θ intorno all'asse $\hat{n}$:

$$R_{\hat{n}}(\theta) \equiv e^{-i\theta \hat{n} \cdot \vec{\sigma}/2} = \cos \frac{\theta}{2} I - i \sin \frac{\theta}{2} (n_x \sigma_x + n_y \sigma_y + n_z \sigma_z),$$

dove $\vec{\sigma} = \sigma_x + \sigma_y + \sigma_z$.

Possiamo ora enunciare il teorema di Bloch:

Teorema 12.4 *Per ogni matrice unitaria 2×2, U esistono numeri reali α, β, γ e δ tali che*

$$U = e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta).$$

Dimostrazione

Come visto nell'esercizio 1.1 dell'Esercitazione 1 (vedi Esercizi di Riepilogo), esistono α, β, δ e γ tali che

$$U = \begin{bmatrix} e^{i(\alpha - \beta/2 - \delta/2)} \cos(\frac{\gamma}{2}) & -e^{i(\alpha - \beta/2 + \delta/2)} \sin(\frac{\gamma}{2}) \\ e^{i(\alpha + \beta/2 - \delta/2)} \sin(\frac{\gamma}{2}) & e^{i(\alpha + \beta/2 + \delta/2)} \cos(\frac{\gamma}{2}) \end{bmatrix}.$$

Si verifica che questa matrice è esattamente il risultato della moltiplicazione $e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta)$. □

Il seguente corollario sarà utile nel seguito per dimostrare il risultato di universalità in 12.3.

Corollary 12.5 *Supponiamo che U sia un'operazione unitaria su un singolo qubit. Allora esistono operatori unitari su un singolo qubit A, B e C tali che $ABC = I$ e $U = e^{i\alpha}AXBXC$, per qualche $\alpha \in \mathbb{R}$.*

Dimostrazione Considera i valori α, β, δ e γ del Teorema 12.4 e definisci

$$A = R_z(\beta)R_y(\frac{\gamma}{2}),\ B = R_y(-\frac{\gamma}{2})R_z(-\frac{\delta+\beta}{2}),$$
$$\text{e } C = R_z(\frac{\delta-\beta}{2}).$$

Si verifica facilmente che $ABC = I$. Dall'esercizio 12.2 e dal fatto che $X^2 = I$ si ottiene:

$$XBX = XR_y(-\frac{\gamma}{2})X^2R_z(-\frac{\delta+\beta}{2})X = R_y(\frac{\gamma}{2})R_z(\frac{\delta+\beta}{2}).$$

Pertanto si ha che

$$\begin{aligned} AXBXC &= R_z(\beta)R_y(\tfrac{\gamma}{2})R_y(\tfrac{\gamma}{2})R_z(\tfrac{\delta+\beta}{2})R_z(\tfrac{\delta-\beta}{2}) \\ &= R_z(\beta)R_y(\gamma)R_z(\delta), \end{aligned}$$

e quindi dal Teorema 12.4

$$U = e^{i\alpha}AXBXC.$$

□

Le seguenti identità sono utili e facili da verificare:

$$HXH = Z;\ HYH = -Y;\ HZH = X.$$

La terza identità ci permette di dimostrare la seguente proposizione

Proposition 12.6 *$HTH = R_x(\pi/4)$ a meno di un fattore di fase globale.*

Dimostrazione
Per definizione di operatore di rotazione e dalla identità $HZH = X$, si ha che $R_x(\pi/4) = e^{-i\frac{\pi}{8}\sigma_x} = e^{-i\frac{\pi}{8}H\sigma_zH}$. Poichè H è invertibile, per la proprietà (5) dell'esponeziale di una matrice (cf. 12.1.1) otteniamo

$$e^{-i\frac{\pi}{8}H\sigma_zH} = H\begin{bmatrix} e^{-i\frac{\pi}{8}} & 0 \\ 0 & e^{i\frac{\pi}{8}} \end{bmatrix}H \approx He^{i\frac{\pi}{8}}\begin{bmatrix} e^{-i\frac{\pi}{8}} & 0 \\ 0 & e^{i\frac{\pi}{8}} \end{bmatrix}H = HTH.$$

□

Esercizio 12.7 *Dimostrare che un arbitrario operatore unitario su singolo qubit può essere scritto nella forma*

$$U = e^{i\gamma}R_{\hat{n}}(\alpha),$$

dove $\gamma, \alpha \in \mathbb{R}$ e $\hat{n}$ è un vettore unitario in $\mathbb{R}^3$.

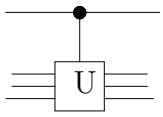

Figure 20: Operazione controllata su singolo qubit

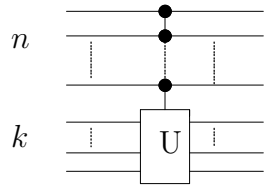

Figure 21: Controlled-U generale con n qubit di controllo e operazione U su k qubit

12.2 Operazioni controllate

Il prototipo di queste operazioni è il `CNOT` (controlled-`NOT`). Come abbiamo già visto, `CNOT` agisce su due qubits chiamati rispettivamente *controllo* e *target* e che il suo effetto nella base computazionale è dato da $|c\rangle\,|t\rangle \rightarrow |c\rangle\,|t \oplus c\rangle$.

In generale, si può definire un'operazione controllata a partire da una qualsiasi operazione unitaria U mediante il circuito in Figura 20. Se U agisce su un singolo qubit, il circuito trasforma i qubits in input $|c\rangle\,|t\rangle$ in $|c\rangle\,U^c\,|t\rangle$, cioè U è applicata al qubit target soltanto se il qubit di controllo è 1.

Il circuito in Figura 21 rappresenta l'operazione controllata più generale $C^n(U)$, dove n è il numero dei qubits di controllo e U è una operazione unitaria su k qubits per qualche intero k.

Le operazioni controllate possono essere implementate usando solo `CNOT` e operazioni unitarie su un singolo qubit che, come vedremo in 12.3, costituisce una *base completa* di cardinalità *infinita* per la computazione quantistica. Un aspetto fondamentale della computazione quantistica è l'implementazione fault-tolerant (cioè resistente agli errori) delle varie operazioni. Esiste un insieme discreto di operazioni elementari che è universale e può essere implementato in modo robusto usando dei codici di correzione degli errori. Come vedremo in 13, questo insieme universale discreto è formato da Hadmard, `CNOT`, S e T. Poichè l'insieme delle operazioni unitarie è continuo, l'universalità di questo insieme discreto deve essere intesa nel senso che le quattro operazioni che lo compongono possono essere usate per *approssimare*

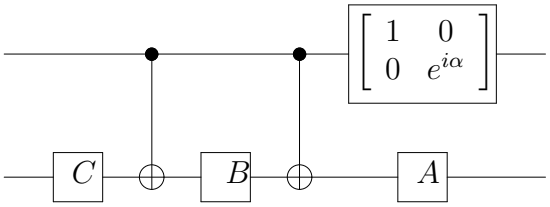

Figure 22: Circuito che implementa controlled-U per operazioni U su singolo qubit.

una qualsiasi operazione unitaria, piuttosto che per implementarla esattamente. Il significato di tale approssimazione verrà chiarito in 13.

12.2.1 Implementazione di $C^n(U)$ mediante `CNOT` e operazioni su singolo qubit

Consideriamo prima il caso più semplice dove $n = 1$ e U è un'operazione unitaria su un singolo qubit, cioè circuiti come quello in Figura 20 dove U ha un solo qubit di input. Per il Corollario 12.5, esistono matrici unitarie su un singolo qubit A, B, C tali che $U = e^{i\alpha} AXBXC$. Allora possiamo costruire un circuito equivalente usando solo `CNOT`, le operazioni A, B, C e l'operazione che realizza la moltiplicazione controllata del qubit target per $e^{i\alpha}$ come mostrato in Figura 22.

L'operazione

$$\alpha = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{bmatrix}$$

applicata al primo qubit effettivamente realizza la moltiplicazione per $e^{i\alpha}$ se il qubit di controllo è 1. Infatti, il suo effetto sulla base computazionale è:

$$|0\rangle \mapsto |0\rangle\,, \quad |1\rangle \mapsto e^{i\alpha}\,|1\rangle\,.$$

Quindi il circuito in Figura 22 opera nel seguente modo[5]:

$$|0\rangle \otimes |x\rangle \mapsto |0\rangle \otimes ABC\,|x\rangle = |0\rangle \otimes |x\rangle\,,$$
$$|1\rangle \otimes |x\rangle \mapsto e^{i\alpha}\,|1\rangle \otimes AXBXC\,|x\rangle = e^{i\alpha}\,|1\rangle \otimes e^{-i\alpha} U\,|x\rangle = |1\rangle \otimes U\,|x\rangle\,.$$

Consideriamo ora il caso in cui $n = 2$ e U è ancora un operatore unitario su un singolo qubit. Allora il circuito in Figura 23 implementa $C^2(U)$ nell'ipotesi che esista un operatore unitario V tale che $V^2 = U$.

[5]Per le trasformazioni lineari basta verificarne l'effetto sulla base computazionale

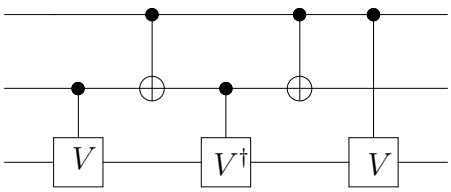

Figure 23: Circuito che realizza $C^2(U)$ per $U = V^2$

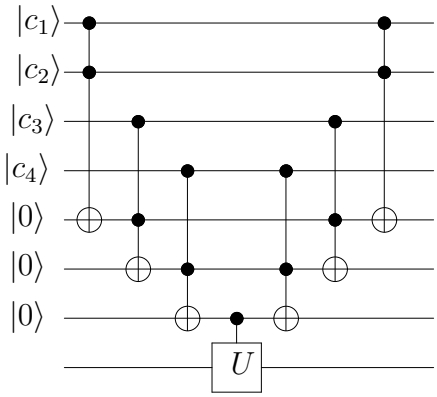

Figure 24: Circuito che realizza $C^n(U)$ per $n = 4$ e $k = 1$

Per $U = X$ si ottiene l'operazione di Toffoli che, vista come operazione classica, è universale per le computazioni classiche reversibili. Ricordiamo che questo risultato di universalità non si poteva ottenere con un numero di input inferiore a tre. Al contrario, l'operazione di Toffoli si può implementare come operazione quantistica usando solo operazioni su uno o due qubit. Tale implementazione si ottiene mediante il circuito in Figura 23 con $V = (1-i)(I+iX)/2$. Infatti si verifica che $V^2 = X$ (esercizio).

Possiamo ora considerare il caso di n qubits di controllo e ancora un solo qubit target, cioè l'operazione $C^n(U)$ definita dall'equazione

$$C^n(U)\,|c_1c_2\ldots c_n\rangle\,|\psi\rangle = |c_1c_2\ldots c_n\rangle\,U^{c_1c_2\ldots c_n}\,|\psi\rangle\,.$$

L'operatore U è applicato al qubit target $|\psi\rangle$ se i primi n qubits sono tutti 1 (l'esponente di U nell'equazione rappresenta un `AND` dei bit c_i). Un circuito che implementa $C^n(U)$ si può ottenere usando l'operazione di Toffoli per realizzare l'`AND` reversibile dei bit di controllo come mostrato in Figura 24.

Esercizio 12.8 *Dimostra che ogni operazione $C^n(U)$, dove U è una matrice unitaria su un singolo qubit, può essere implementata usando $O(n)$ operazioni su singolo qubit e* `CNOT`*.*

12.3 Universalità di CNOT e operazioni su singolo qubit

Qualsiasi operatore unitario si può esprimere in maniera *esatta* mediante CNOT e operazioni su singolo qubit. Per dimostrarlo dobbiamo completare i risultati visti precedentemente per le operazioni controllate, considerando operazioni unitarie U su un numero arbitrario $k \geq 2$ di qubits. L'idea fondamentale è quella di scomporre una U arbitraria nel prodotto di matrici unitarie che operano in maniera non banale su al più due componenti del vettore a cui sono applicate. Poichè tali matrici, dette *matrici unitarie a due livelli*, si possono implementare mediante circuiti costruiti usando CNOT e operazioni su singolo qubit, potremo quindi concludere il risultato di universalità sopra citato.

Dato un operatore U su uno spazio d-dimensionale, è sempre possibile mediante una procedura puramente algebrica costruire matrici unitarie a due livelli $V_1, V_2 \dots V_{d'}$ tali che $U = V_1 V_2 \dots V_{d'}$ e $d' \leq d(d-1)/2$.

Quindi non ci rimane altro che far vedere come costruire un circuito che implementa una matrice unitaria a due livelli ed è formato solo da CNOT e operazioni su singolo qubit.

Supponiamo che U sia una matrice unitaria a due livelli che opera su n qubits e supponiamo che gli stati su cui agisce effettivamente siano tutti quelli generati dai due stati $|s\rangle$ e $|t\rangle$ della base computazionale, dove s e t sono le rappresentazioni binarie dei numeri $s, t \in [0, n-1]$. Allora la sottomatrice non banale $\tilde{U}$ di U è una matrice unitaria 2×2 e si può vedere come un operatore unitario su un singolo qubit. Per costruire il nostro circuito consideriamo la sequenza $g_1, g_2, \dots, g_m$ di numeri binari tali che $g_1 = s$, $g_m = t$ e per ogni $1 \leq i < m$, g_i e g_{i+1} differiscono per esattamente un bit. Tali sequenze sono dette *codici di Gray*. Ad esempio 1011, 1001, 0001, 0000 è un codice di Gray che connette 1011 e 0000.

Usando operazioni controllate si realizza il passaggio da g_i a g_{i+1} per $1 \leq i < m-1$. A questo punto si esegue un controlled-$\tilde{U}$ con qubit target corrispondente al singolo bit su cui g_{m-1} e g_m differiscono e con i qubits di controllo corrispondenti ai valori dei bits che sono identici in g_{m-1} e g_m. Le operazioni di passaggio da g_i a g_{i+1} devono essere realizzate in modo reversibile e quindi in generale saranno necessarie $2(n-1)$ operazioni controllate per implementarle.

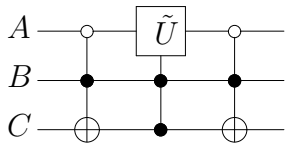

Figure 25: Circuito che implementa la matrice a due livelli U dell'esempio 12.9

Esempio 12.9 *Considera la matrice su tre qubits definita da*

$$U = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a & 0 & 0 & 0 & 0 & c \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & b & 0 & 0 & 0 & 0 & d \end{bmatrix},$$

dove $\tilde{U} = \begin{bmatrix} a & c \\ b & d \end{bmatrix}$ *è una arbitraria matrice unitaria* 2×2*. Notiamo che* U *agisce effettivamente solo sugli stati (generati da)* $|s\rangle = |010\rangle$ *e* $|t\rangle = |111\rangle$*. Il circuito che implementa* U *è dato in Figura 25. Gli input* A, B, C *sono inizialmente nella configurazione corrispondente a* s*, cioè* 010*. Lo stato successivo della sequenza* $\{g_i\}$ *è* 011 *e il passaggio è effettuato da un* `NOT` *su* C *condizionato dai valori dei bit* $A = 0$ *e* $B = 1$ *(ricordiamo che il pallino bianco indica che il target viene modificato se il controllo è* 0*, in maniera duale al pallino nero). Lo stato successivo è lo stato finale* $|t\rangle = |111\rangle$*. Quindi applichiamo* $\tilde{U}$ *ad* A *a condizione che* $B = 1$ *e* $C = 1$*. La parte destra del circuito riproduce simmetricamente la parte sinistra per ottenere la reversibilità.*

Nota 12.10 *La procedura descritta corrisponde essenzialmente a trovare un cambio nella base computazionale tale che i due livelli non banali corrispondano a un singolo qubit. Ricordiamo che la matrice di rappresentazione nella nuova base è* $V = BUB^{-1}$*, dove* B *è la matrice del cambio di base, da cui si ottiene* $U = B^{-1}VB$*. Individuare un codice di Gray e implementare le trasformazioni descritte dall'algoritmo corrisponde essenzialmente a costruire* B *e la sua inversa.*

Dall'esercizio 12.8 sappiamo che ciascuna delle operazioni controllate necessarie per effettuare il passaggio da g_1 a g_m in modo reversibile richiede $O(n)$ operazioni su singolo qubit e `CNOT`. Abbiamo visto inoltre che tale passaggio richiede almeno $2(n-1)$ operazioni controllate. Quindi per implementare una matrice unitaria a due livelli sono necessarie $O(n^2)$ operazioni su singolo qubit e `CNOT`.

In conclusione, possiamo implementare una generica operazione unitaria su n qubits come composizione di al più $2^n(2^n-1)/2$, cioè $O(2^{2n})$, operazioni unitarie a due livelli, e quindi con $O(n^2 2^{2n})$ operazioni su singolo qubit e `CNOT`.

Naturalmente questa implementazione è estremamente inefficiente. In pratica si usa un'altra tecnica che permette di utilizzare un insieme discreto di gates per ottenere in maniera efficiente un'implementazione non più esatta ma *approssimata* di una qualsiasi trasformazione unitaria su due o più qubits.

13 Circuiti quantistici: realizzazione approssimata

13.1 Universalità di Hadamard, `CNOT`, S e T

Le operazioni Hadamard, `CNOT`, S e T formano un insieme universale *finito* per la computazione quantistica nel senso che data un'operazione unitaria arbitraria, è possibile simularla con una "buona approssimazione" mediante un circuito che contiene un numero finito di queste operazioni.

Per definire la nozione di approssimazione abbiamo bisogno di introdurre la definizione di norma di un operatore.

Definition 13.1 *Dato un vettore $|\psi\rangle \in \mathbb{C}^n$, consideriamo la norma euclidea definita da $\sqrt{\langle\psi|\psi\rangle}$. La norma di un operatore $L : \mathbb{C}^n \to \mathbb{C}^n$ è definita da:*

$$\|L\| = sup_{|\psi\rangle \neq 0} \frac{\|L|\psi\rangle\|}{\||\psi\rangle\|}.$$

Per definizione di norma, la norma di un operatore gode delle seguenti proprietà:

$$\begin{aligned}
&\|L\| \geq 0 \\
&\|L\| = 0 \text{ sse } L = 0 \\
&\|L + V\| \leq \|L\| + \|V\| \\
&\|cL\| = c\|L\|.
\end{aligned}$$

Si verificano inoltre le seguenti importanti proprietà:

(*i*) $\|LV\| \leq \|L\| \cdot \|V\|$,

(*ii*) $\|L^\dagger\| = \|L\|$,

(*iii*) $\|L \otimes V\| = \|L\| \cdot \|V\|$,

(*iv*) $\|U\| = 1$ se U è unitaria.

Possiamo ora dare la definizione di realizzazione approssimata di un operatore U, che denoteremo con $\tilde{U}$.

Definition 13.2 *L'operatore* $\tilde{U}$ approssima l'operatore U con precisione δ *se*

$$\|\tilde{U} - U\| \leq \delta.$$

Dalle proprietà (*i*) e (*iv*) segue che se $\tilde{U}$ approssima U con precisione δ, allora $\tilde{U}^{-1}$ approssima U^{-1} con la stessa precisione δ.

Un'altra importante conseguenza è che gli errori si accumulano linearmente: se $U = U_m \cdots U_2 U_1$ e ogni U_k ha un'approssimazione $\tilde{U_k}$ con precisione δ_k, allora il prodotto di queste approssimazioni $\tilde{U} = \tilde{U_m} \cdots \tilde{U_2}\tilde{U_1}$ approssima U con precisione $\sum_k \delta_k$.

Il risultato di universalità sopra citato è conseguenza del fatto che le due operazioni T e HTH possono essere usate per approssimare una qualsiasi operazione unitaria su singolo qubit con un errore arbitrariamente piccolo.

Ricordiamo che, a meno di un fattore di fase globale, T corrisponde a una rotazione di $\pi/4$ intorno all'asse $\hat{z}$ e HTH a una rotazione di $\pi/4$ intorno all'asse $\hat{x}$ (cf. Proposizione 12.6). Componendo queste due operazioni si ottiene:

$$\begin{aligned} e^{-i\frac{\pi}{8}\sigma_x} e^{-i\frac{\pi}{8}\sigma_z} &= (\cos\frac{\pi}{8} I - i \sin\frac{\pi}{8}\sigma_x)(\cos\frac{\pi}{8} I - i\sin\frac{\pi}{8}\sigma_z) \\ &= \cos^2\frac{\pi}{8} I - i(\cos\frac{\pi}{8}(\sigma_x + \sigma_z) + i\sigma_x\sigma_z \sin\frac{\pi}{8})\sin\frac{\pi}{8} \\ &= \cos^2\frac{\pi}{8} I - i(\cos\frac{\pi}{8}(\sigma_x + \sigma_z) + \sin\frac{\pi}{8}\sigma_y)\sin\frac{\pi}{8} \end{aligned}$$

Ricordando che $R_{\hat{n}}(\theta) = \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2}(n_x\sigma_x + n_y\sigma_y + n_z\sigma_z)$, l'espressione ottenuta è quindi una rotazione intorno all'asse $\vec{n} = (\cos\frac{\pi}{8}, \sin\frac{\pi}{8}, \cos\frac{\pi}{8})$, con corrispondente vettore unitario $\hat{n}$, di un angolo θ definito da $\cos\theta/2 = \cos^2\frac{\pi}{8}$. Iterando l'applicazione di $R_{\hat{n}}(\theta)$ per un numero sufficiente di volte si può approssimare una qualsiasi rotazione $R_{\hat{n}}(\alpha)$ in modo da ottenere un arbitrario livello di precisione δ. Questo ci è garantito dal seguente teorema:

Teorema 13.3 *Date due rotazioni $R_{\hat{n}}(\theta_1)$ e $R_{\hat{n}}(\theta_2)$ di un qubit intorno allo stesso asse $\hat{n}$, se $\beta \in [0, 2\pi)$ è tale che $\frac{\beta}{\pi}$ non è razionale, allora per qualsiasi $\delta > 0$ e $\alpha \in [0, 2\pi)$, esiste $m \in \mathbb{Z}$ tale che*

$$\|R_{\hat{n}}^m(\beta) - R_{\hat{n}}(\alpha)\| \leq \delta.$$

Quindi, ricordando il risultato dimostrato nell'esercizio 12.7, dato un operatore unitario U su singolo qubit e un $\delta > 0$, è possibile approssimare U con precisione δ usando un circuito composto soltanto da operazioni T e H (il gate S non è strettamente necessario per questo risultato di universalità, ma viene usato per effettuare approssimazioni fault-tolerant).

In conclusione, dato un circuito che contiene solo `CNOT` e operazioni unitarie su un singolo qubit (che formano una base completa, come visto in 12.3), possiamo simulare questo circuito con un'approssimazione fissata ma arbitraria, usando solo Hadamard, S, T e `CNOT`. Inoltre tale simulazione è efficiente nel senso che può essere eseguita in $O(\log^c s/\delta)$, dove c è una costante, s è il numero di gates nel circuito originario e δ è l'accuratezza desiderata (cf. Teorema di Solovay-Kitaev, Appendice 3 in [4]).

Un altro insieme finito di gates per il quale si dimostra un analogo risultato di universalità è formato da Hadamard, `CNOT`, S e l'operazione $C^2(X)$ (gate di Toffoli). Una dimostrazione si trova in [3].

Algoritmi e complessità computazionale quantistica

14 Cenni di complessità computazionale classica

La teoria della calcolabilità fornisce criteri per analizzare i problemi computazionali dal punto di vista della loro risolvibilità per via algoritmica. La domanda centrale della teoria della calcolabilità riguarda quindi l'*esistenza di algoritmi*. La teoria della complessità va oltre la semplice esistenza di un algoritmo concentrandosi sulla possibilità di realizzare una soluzione utilizzando una quantità ragionevole di risorse. Il suo obiettivo è quindi quello di indagare sull'*esistenza di algoritmi efficienti* cercando una controparte formale per il concetto di algoritmo efficiente.

14.1 Classi di complessità

L'obiettivo della teoria della complessità è quello di classificare i problemi computazionali in base alla quantità di risorse — tipicamente il tempo o il numero di passi — richieste per risolverli su un computer. Questa quantità

rappresenta una misura di complessità ben definita sulla base della *tesi di Church-Turing* secondo cui ogni modello "ragionevole" di calcolo può essere simulato *efficientemente* (cioè in tempo polinomiale rispetto al tempo di esecuzione della macchina simulata) su una Macchina di Turing probabilistica. Un'interpretazione informale di questa tesi è la seguente: Tutte le implementazioni fisiche di dispositivi di computazione possono essere simulate in tempo polinomiale da una macchina di Turing probabilistica[6].

Questa tesi permette di parlare di algoritmi efficienti (o inefficienti) in generale, senza fare cioè riferimento ad un particolare modello, e assicura la validità generale di definizioni e dimostrazioni di trattabilità e intrattabilità condotte su un particolare modello. Nelle seguenti definizioni useremo come modello di riferimento la Macchina di Turing (MdT)[1, 5].

L'oggetto principale di studio in teoria della complessità sono problemi formulati in modo *decisionale*, cioè in modo che la soluzione equivalga ad una scelta tra i due valori alternativi "si" o "no". Secondo questa formulazione un problema corrisponde ad un insieme di istanze e le sue soluzioni dal sottinsieme delle istanze positive (cioè il cui valore è "si"). Se si usa un alfabeto di simboli Σ per codificare le istanze di un problema, allora si può associare ad ogni problema decisionale l'insieme delle stringhe sull'alfabeto Σ (o linguaggio) formato dalle istanze positive del problema. Di conseguenza un problema decisionale è equivalente al problema di decidere l'appartenenza di una stringa al linguaggio che ne descrive le istanze positive. Questo sarà un sottinsieme di Σ^*, cioè l'insieme di tutte le possibili stringhe su Σ. Le definizioni delle varie classi di complessità sono date classicamente utilizzando tale formulazione.

Definition 14.1 *Un linguaggio $L \subseteq \Sigma^*$ è nella classe* **P** *(tempo polinomiale) se esiste una Macchina di Turing deterministica che dato un input $x \in \Sigma^*$ decide in tempo polinomiale nella lunghezza dell'input se x appartiene a L oppure no.*

Dopo l'avvento degli algoritmi randomizzati[7], è stata introdotta un'altra classe computazionale che comprende tutti i problemi per i quali esistono algoritmi probabilistici che li risolvono in tempo polinomiale e con probabilità

[6]Come vedremo questa versione moderna della Tesi di Church-Turing è stata recentemente messa in discussione da alcuni risultati che evidenziano come questa accezione "fisica" della tesi non sia più valida a livello della meccanica quantistica.

[7]Nel 1970 Solavay e Strassen costruirono un algoritmo randomizzato per determinare se un numero è primo oppure no. L'algoritmo fornisce una risposta corretta in tempo polinomiale ma solo con una certa probabilità che può essere tuttavia aumentata ripetendo l'esecuzione dell'agoritmo.

di errore piccola. Questa classe è nota come **BPP** (bounded error probabilistic polynomial time), e ha sostituito **P** come classe rappresentativa dei problemi effettivamente trattabili su un computer classico.

Definition 14.2 *Un linguaggio $L \subseteq \Sigma^*$ è nella classe* **BPP** *se esiste una Macchina di Turing probabilistica che dato un input $x \in \Sigma^*$ in tempo polinomiale nella lunghezza dell'input lo accetta con probabilità almeno $2/3$ quando x appartiene a L e lo rifiuta con probabilità almeno $2/3$ quando x non appartiene a L.*

La classe **BPP** comprende quindi tutti i problemi *trattabili*, cioè tutti quei problemi che possono essere risolti efficientemente.

La classe di complessità che rappresenta i problemi (ritenuti fino ad oggi) classicamente *intrattabili* è la classe **NP**. Informalmente si possono vedere i problemi in questa classe come problemi decisionali per i quali ipotetiche soluzioni possono essere verificate in tempo polinomiale, pur essendo il calcolo di tali soluzioni presumibilmente "difficile". Il problema della fattorizzazione è un ben noto rappresentante di questa classe: mentre calcolare i fattori di un numero intero positivo n richiede un tempo che cresce esponenzialmente in n, la verifica che due numeri x e y siano fattori di n si può fare efficientemente (basta moltiplicare x e y).

Definition 14.3 *Un linguaggio $L \subseteq \Sigma^*$ è nella classe* **NP** *se esiste una Macchina di Turing non-deterministica M che per ogni input $x \in \Sigma^*$ decide in tempo polinomiale nella lunghezza dell'input se x appartiene a L oppure no.*

Lo studio delle relazioni tra le classi **P** e **NP** è un argomento fondamentale in complessità che ha come scopo quello di quantificare i vantaggi computazionali offerti dal non determinismo rispetto al determinismo nell'ambito di computazioni di lunghezza polinomiale. Mentre l'nclusione $\mathbf{P} \subseteq \mathbf{NP}$ è facilmente dimostrabile, lo studio della validità dell'inclusione inversa è ancora oggi il più importante problema aperto della complessità computazionale. Ci sono molte ragioni per credere che $\mathbf{P} \neq \mathbf{NP}$. Fino ad oggi questa ipotesi non è stata smentita neppure in riferimento ad una classificazione dei problemi computazionali che tiene conto del paradigma di computazione quantistico.

15 Complessità computazionale quantistica

Con gli sviluppi che si sono avuti nel campo della computazione quantistica (vedi l'algoritmo di Deutsch-Jozsa e l'algoritmo di Shor che vedremo più avanti), si è sentita l'esigenza di rivedere la teoria della compessità facendo riferimento alla meccanica quantistica. I risultati ottenuti si basano su modelli formali di calcolo per la computazione quantistica analoghi a quelli classici: la Macchina di Turing Quantistica e i circuiti quantistici introdotti da Deutsch e Yao. L'equivalenza di questi due modelli e la loro universalità è stata dimostrata da Yao [6]. Pertanto le definizioni e i risultati che vedremo nel seguito hanno validità in entrambi i modelli sebbene formulati in termini di Macchina di Turing quantistica (QTM).

Il corrispondente quantistico della classe di complessità classica **BPP** è la classe dei problemi che possono essere risolti in tempo polinomiale su una Macchina di Turing quantistica. Tale classe si indica con **BQP** (bounded error quantum polynomial time).

Definition 15.1 *Un linguaggio $L \subseteq \Sigma^*$ è nella classe* **BQP** *se esiste una Macchina di Turing quantistica che dato un input $x \in \Sigma^*$, in tempo polinomiale nella lunghezza dell'input lo accetta con probabilità almeno $2/3$ quando x appartiene a L e lo rifiuta con probabilità almeno $2/3$ quando x non appartiene a L.*

Questa definizione si può equivalentemente riformulare in termini di circuiti quantistici. Dal fatto che il circuito (quantistico) di Toffoli è universale per le computazioni classiche reversibili, segue in modo abbastanza immediato il risultato che $\mathbf{P} \subseteq \mathbf{BQP}$. Inoltre, poichè altrettanto facilmente si dimostra che una Macchina di Turing probabilistica si può simulare con una macchina di Turing quantistica, si ottiene che

$$\mathbf{P} \subseteq \mathbf{BPP} \subseteq \mathbf{BQP}.$$

Non esistono al momento risultati che dimostrano che $\mathbf{BPP} \neq \mathbf{BQP}$ e quindi la superiorità del modello quantistico rispetto a quello classico. D'altra parte, dimostrare che **BPP** è strettamente contenuto in **BQP** ($\mathbf{BPP} \subset \mathbf{BQP}$) porterebbe a risolvere il problema centrale (ancora aperto) della teoria della complessità computazionale, se $\mathbf{P} \neq \mathbf{PSPACE}$[8].

La scoperta di una tecnica chiamata *Fourier Sampling* ha permesso di definire alcuni algoritmi quantistici con prestazioni esponenzialmente migliori

[8] Questa classe contiene problemi che possono essere calcolati da una MdT in *spazio* polinomiale nella lunghezza dell'input.

dei loro omologhi classici. Questo ha portato naturalmente a chiedersi se un computer quantistico sia in grado di risolvere in tempo polinomiale problemi **NP**-completi, cioè se $\mathbf{NP} \subseteq \mathbf{BQP}$. Ricordiamo che i problemi **NP**-completi sono problemi **NP** che rappresentano tutti quei problemi cosiddetti "difficili" che si ritengono classicamente intrattabili. Una risposta affermativa alla questione posta precedentemente avrebbe quindi delle conseguenze di enorme importanza.

La *Trasformata di Fourier Quantistica* rappresenta fino ad oggi l'unico strumento il cui uso permette di costruire algoritmi quantistici che sono esponenzialmente più efficienti dei corrispettivi classici.

16 La Trasformata di Fourier Quantistica

La scoperta finora più importante in computazione quantistica consiste nella dimostrazione che fattorizzare un mumero di n bits su un computer quantistico richiede un numero di operazioni dell'ordine di $O(n^2 \log n \log\log n)$, mentre il migliore algoritmo classico che si conosce richiede tempo esponenziale. Il problema di stabilire se e quali altri problemi che risultano intrattabili su un computer classico possono essere risolti efficientemente con un computer quantistico è di fondamentale importanza e rappresenta un punto cruciale della ricerca corrente.

Un ingrediente fondamentale dell'algoritmo quantistico per la fattorizzazione è la Trasformata di Fourier Quantistica (QFT) che ha applicazioni anche nella soluzione di altri problemi classicamente noti come problemi **NP**.

Un processo fisico, in generale, può essere espresso sia nel dominio dei tempi, con una funzione $h(t)$ che descrive la variazione di una generica grandezza, sia nel dominio delle frequenze[9]. La trasformata di Fourier è un metodo matematico per passare dal dominio dei tempi a quello delle frequenze. Questo metodo permette di trasformare una funzione di periodo r in una funzione che assume valori diversi da zero solo in corrispondenza dei multipli della frequenza $\frac{2\pi}{r}$. La *Trasformata di Fourier Discreta* (DFT) opera su N punti scelti a uguale distanza nell'intervallo $[0, 2\pi)$, per qualche intero N e restituisce una funzione il cui dominio sono i numeri tra 0 e $N-1$. La DFT di una funzione di periodo r è una funzione concentrata sui multipli di $\frac{N}{r}$: se r divide N allora la funzione risultante avrà valori diversi da zero solo sui multipli di $\frac{N}{r}$; altrimenti la funzione assumerà valori non nulli anche su interi vicini ai multipli di $\frac{N}{r}$.

[9] La frequenza è l'inverso del periodo.

Nella notazione matematica usuale, la DFT prende in input un vettore di numeri complessi $x_0, x_1, \ldots, x_{N-1}$ e restituisce il vettore trasformato $y_0, y_1, \ldots, y_{N-1}$ definito da

$$y_k \equiv \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi i j k/N}.$$

La *Trasformata di Fourier Quantistica* (QFT) è una variante della versione della DFT dove N è una potenza di 2 (la Fast Fourier Transform o FFT). La QFT opera sulle ampiezze degli stati quantistici in modo simile alla DFT. In particolare, se $|0\rangle, \ldots, |N-1\rangle$ è una base ortonormale dello spazio degli stati, allora un generico stato $\sum_{j=0}^{N-1} x_j |j\rangle$ viene trasformato in $\sum_{k=0}^{N-1} y_k |k\rangle$, dove le ampiezze y_k sono le DFT delle ampiezze x_j.

Esercizio 16.1 *Dimostrare che l'operatore*

$$F : \sum_{j=0}^{N-1} x_j |j\rangle \mapsto \sum_{k=0}^{N-1} y_k |k\rangle,$$

è unitario.

La QFT è quindi definita come l'operatore lineare F che trasforma gli stati della base nel seguente modo:

$$|j\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k/N} |k\rangle.$$

Equivalentemente, la QFT si può vedere come una matrice F, $N \times N$ il cui elemento generico F_{jk} è $(e^{2\pi i/N})^{jk}$. Un modo per dimostrare che la trasformazione definita dalla QFT è unitaria è mediante la costruzione di un circuito quantistico che la implementa.

Assumiamo che $N = 2^n$ e che la base $|0\rangle, \ldots, |2^n - 1\rangle$ sia la base computazionale per gli stati su n qubits. Inoltre, è conveniente esprimere il generico stato $|j\rangle$ della base computazionale come il prodotto tensore $|j_1, \ldots, j_n\rangle$ dei bits della rappresentazione binaria di j, $j_1 2^{n-1} + j_2 2^{n-2} + \ldots, +j_n 2^0$. Possiamo quindi esprimere la QFT di $|j\rangle$ mediante il prodotto

$$\frac{(|0\rangle + e^{2\pi i 0.j_n} |1\rangle)(|0\rangle + e^{2\pi i 0.j_{n-1} j_n} |1\rangle) \ldots (|0\rangle + e^{2\pi i 0.j_1 j_2 \ldots j_n} |1\rangle)}{\sqrt{2^n}}, \qquad (5)$$

dove la notazione $0.j_l j_{l+1} \ldots j_m$ rappresenta la frazione binaria $j_l/2 + j_{l+1}/2^2 + \ldots + j_m/2^{m-l+1}$. Infatti, osservando che

$$\begin{aligned} \frac{k}{2^n} &= \frac{k_1 2^{n-1}}{2^n} + \ldots + \frac{k_n 2^0}{2^n} \\ &= \frac{k_1}{2} + \ldots + \frac{k_n}{2^n}, \end{aligned}$$

otteniamo

$$\begin{aligned} |j\rangle &\rightarrow \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i j k/2^n} |k\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} e^{2\pi i j \left(\sum_{l=1}^{n} k_l 2^{-l}\right)} |k_1 \ldots k_n\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \bigotimes_{l=1}^{n} e^{2\pi i j k_l 2^{-l}} |k_l\rangle \\ &= \frac{1}{\sqrt{2^n}} \bigotimes_{l=1}^{n} \left[\sum_{k_l=0}^{1} e^{2\pi i j k_l 2^{-l}} |k_l\rangle \right] \\ &= \frac{1}{\sqrt{2^n}} \bigotimes_{l=1}^{n} \left[|0\rangle + e^{2\pi i j 2^{-l}} |1\rangle \right], \end{aligned}$$

che sviluppato risulta nel prodotto (5).

Da questa rappresentazione della QFT si può derivare il circuito che la implementa e descritto in Figura 26.

L'input è un registro di n qubits ognuno dei quali rappresenta una cifra binaria della rappresentazione di $j = j_1 \ldots j_n$. Applicando Hadamard al primo qubit $|j_1\rangle$ si ottiene

$$\frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i 0.j_1} |1\rangle \right) |j_2 \ldots j_n\rangle,$$

in quanto se $j_1 = 1$, $e^{2\pi i 0.j_1} = e^{\pi i} = -1$, mentre se $j_1 = 0$, $e^{2\pi i 0.j_1} = 1$. Le altre operazioni impiegate nel circuito sono trasformazioni unitarie della forma

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix},$$

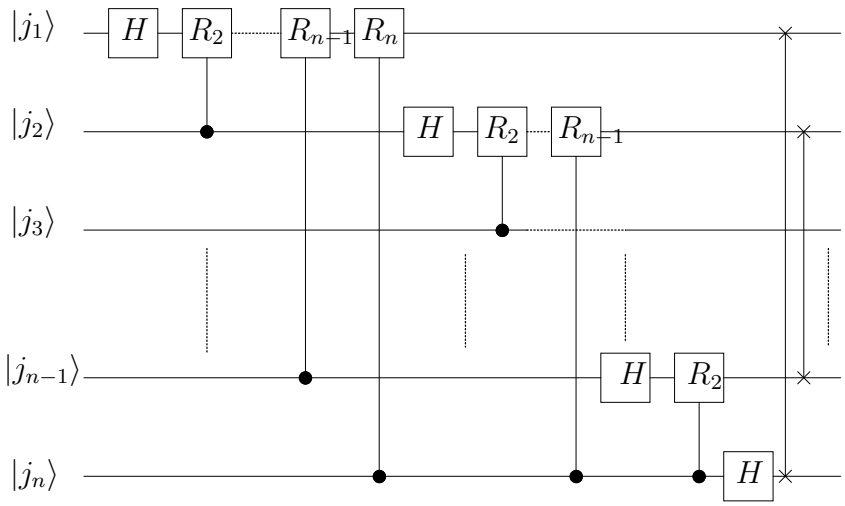

Figure 26: Circuito per la trasformata di Fourier quantistica

che usate come operazioni controllate permettono di aggiungere il bit di indice k alla fase del coefficiente di $|1\rangle$. Dopo l'applicazione dei controlled-R_i, $i = 2, \ldots, n$ al primo qubit si ottiene quindi

$$\frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i 0.j_1 \ldots j_n} |1\rangle\right) |j_2 \ldots j_n\rangle \,.$$

A questo punto si esegue un procedimento analogo al secondo qubit ottenendo

$$\frac{1}{\sqrt{2^2}} \left(|0\rangle + e^{2\pi i 0.j_1 \ldots j_n} |1\rangle\right) \left(|0\rangle + e^{2\pi i 0.j_2 \ldots j_n} |1\rangle\right) |j_3 \ldots j_n\rangle \,,$$

e si continua in questo modo per ogni qubit, fino ad ottenere lo stato

$$\frac{1}{\sqrt{2^n}} \left(|0\rangle + e^{2\pi i 0.j_1 \ldots j_n} |1\rangle\right) \left(|0\rangle + e^{2\pi i 0.j_2 \ldots j_n} |1\rangle\right) \ldots \left(|0\rangle + e^{2\pi i 0.j_n} |1\rangle\right) .$$

Per ritrovare la QFT di $|j\rangle$ basta ora applicare $n/2$ operazioni di scambio per riportare i fattori nell'ordine giusto. Ricordiamo che è possibile scambiare due qubit utilizzando tre `CNOT`. Poichè tutte le operazioni utilizzate sono unitarie, possiamo concludere che QFT è una trasformazione unitaria. In totale il numero delle operazioni richieste per calcolare la QFT è $O(n^2)$ (sono utilizzati $(n+1) + n + (n-1) + \ldots + 1 = n(n+1)/2$ Hadamard e controlled-R_k; inoltre sono necessari $\lceil n/2 \rceil$ circuiti per lo scambio di due

qubits). Notiamo che contrariamente al caso classico della DFT o della FFT, il risultato dell'applicazione della QFT ad un generico stato ottenuto come sovrapposizione degli stati $|j\rangle$ della base computazionale risiede nelle ampiezze della sovrapposizione quantistica e non è quindi direttamente accessibile mediante misurazione. Gli usi della QFT in computazione quantistica sono tutti di tipo indiretto, ma fondamentali per la costruzione di tutti quegli algoritmi quantistici che esibiscono un comportamento esponenzialmente più veloce rispetto alle loro controparti classiche.

Esercizio 16.2 *Verificare che l'inversa della trasformata di Fourier quantistica è*

$$QFT^{\dagger} : |j\rangle \rightarrow \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^{n-1}} e^{-2\pi i j k/2^n} |k\rangle ,$$

e costruire un circuito che la implementa.

17 Algoritmi quantistici basati sulla Trasformata di Fourier Quantistica

La Trasformata di Fourier è alla base di una procedura generale nota come *stima delle fasi* che permette di ottenere una stima degli autovalori di una matrice unitaria. Questa procedura è a sua volta parte essenziale di molti algoritmi quantistici.

17.1 Stima dell'autovalore di una matrice unitaria

Dato un operatore unitario U su m qubits con un autovalore λ corrispondente ad un autovettore $|u\rangle$, il problema è stabilire con una certa precisione il valore di λ disponendo di oracoli che eseguono controlled-U^{2^j}, $j = 0, \ldots, t$ per un dato intero positivo t, e di una preparazione dello stato $|u\rangle$. Nota che poichè U è unitaria, $|\lambda| = 1$ cioè $\lambda = e^{2\pi i \varphi}$ per qualche φ tale che $0 \leq \varphi \leq 1$. Il problema consiste quindi nel trovare una stima per φ.

Questo problema si può risolvere efficientemente mediante l'algoritmo quantistico che descriviamo nel seguito. L'uso di questo algoritmo non è limitato a risolvere il problema descritto, ma è parte integrante di altri algoritmi, dal momento che molti problemi si possono ridurre al problema della stima delle fasi, tra cui quelli descritti in 17.2 e l'algoritmo per la fattorizzazione in 17.5.

Il circuito schematizzato in Figura 27 implementa la procedura che si svolge in tre passi a partire da un input formato da due registri.

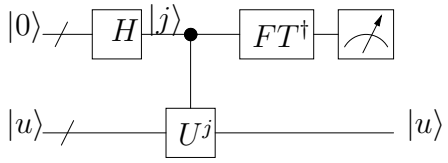

Figure 27: Circuito schematico per l'algoritmo di stima delle fasi

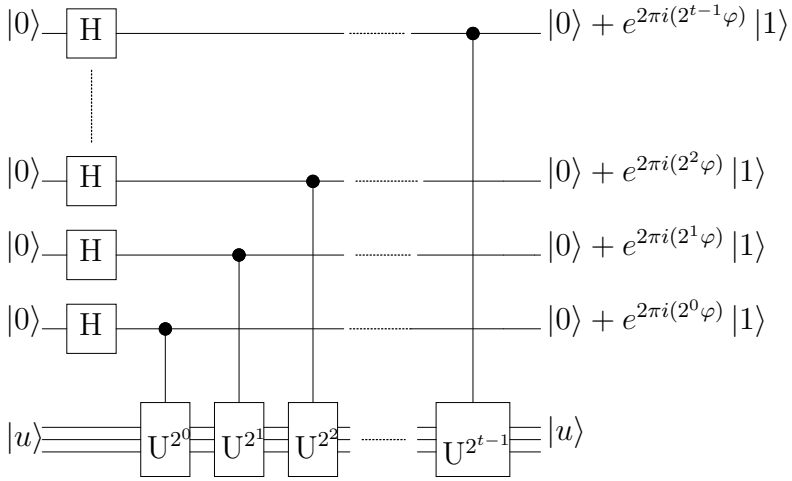

Figure 28: Descrizione dettagliata del primo stadio del circuito in Figura 27

Il primo registro contiene t qubits inizialmente nello stato $|0\rangle$. Il secondo registro contiene l'autovettore $|u\rangle$ che (come vedremo) non verrà modificato dall'applicazione di U^j. Nella fase iniziale si applica $H^{\otimes t}$ al primo registro $|0\rangle^{\otimes t}$ che diventa quindi

$$|j\rangle = \left[(|0\rangle + |1\rangle)/\sqrt{2}\right]^{\otimes t} = \frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} |k\rangle \,.$$

Ciascun qubit nel primo registro viene usato come qubit di controllo dell'operazione controlled-U^{2^k} come illustrato in Figura 28.

L'effetto della sequenza di controlled-U^{2^k} è di trasformare $|j\rangle\,|u\rangle$ in $|j\rangle\,U^j\,|u\rangle$. Infatti, notando che la rappresentazione binaria di j è $j_1 2^{t-1} + j_2 2^{t-2} + \ldots j_t 2^0$,

$$U^j = U^{j_t 2^0} U^{j_{t-1} 2^1} \cdots U^{j_2 2^{t-2}} U^{j_1 2^{t-1}},$$

che viene applicata solo quando $j_k = 1$ per ogni $k = 1, \ldots, t$, cioè solo se $j = 1$.

Osserviamo che se $|u\rangle$ è una autovettore di U con autovalore λ, allora $|u\rangle$ è anche un autovettore di U^{2^k} con autovalore λ^{2^k}. Pertanto

$$\begin{aligned} |j\rangle U^j |u\rangle &= |j\rangle e^{2\pi i j\varphi} |u\rangle \\ &= e^{2\pi i j\varphi} |j\rangle |u\rangle \\ &= \left[(e^{2\pi i j_1 2^{t-1}\varphi} |j_1\rangle) \cdots (e^{2\pi i j_{t-1} 2^1 \varphi} |j_{t-1}\rangle)(e^{2\pi i j_t 2^0 \varphi} |j_t\rangle) \right] |u\rangle . \end{aligned}$$

Poichè ogni j_k è della forma $(|0\rangle + |1\rangle)/\sqrt{2}$, lo stato del primo registro dopo l'applicazione del circuito in Figura 28 risulta

$$\bigotimes_{k=0}^{t-1} \frac{|0\rangle + e^{2\pi i 2^k \varphi} |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2^t}} \sum_{y=0}^{2^t-1} e^{2\pi i \varphi y/2^t} |y\rangle ,$$

che corrisponde all'espressione ottenuta dall'applicazione della QFT al vettore $|\varphi\rangle$. Il secondo passo dell'algoritmo consiste quindi nell'applicazione dell'inversa della trasformata di Fourier, $QFT^\dagger$. Supponiamo che φ si possa rappresentare esattamente con t bits. Allora il secondo passo produce $|\varphi_1 \varphi_2 \ldots \varphi_t\rangle$. L'ultimo passo è una misurazione nella base computazionale che permette quindi di ottenere il valore esatto di φ.

Supponiamo ora che φ sia un qualsiasi numero reale (razionale o irrazionale) in $[0, 1]$ la cui rappresentazione su t bit è approssimata a meno di un errore δ. Più precisamente, supponiamo che $\varphi = a/2^t + \delta$, dove $a = a_0 a_1 \ldots a_{t-1}$, $a_k \in \{0, 1\}$ per ogni $k \in [0, t-1]$. Quindi $a/2^t$ è la migliore approssimazione di φ su t bit, cioè $0 \leq \delta \leq 1/2^{t+1}$.Vogliamo considerare il caso in cui l'errore δ è strettamente positivo. In questo caso si dimostra che dopo l'applicazione della $QFT^\dagger$, una misurazione dello stato finale produce questa approssimazione su t bits di φ con probabilità almeno $4/\pi^2 \approx 0.405$.

Il valore di t si può fissare in modo da ottenere una stima di φ precisa quanto si vuole. Si può cioè fissare un ϵ tale che l'algoritmo produca un'approssimazione di φ precisa fino all'n-simo bit con probabilità almeno $1 - \epsilon$. Per ottenere questa stima si deve fissare t al valore

$$t = n + \left\lceil \log \left(2 + \frac{1}{2\epsilon} \right) \right\rceil , \tag{6}$$

dove $\lceil \cdot \rceil$ indica la parte intera alta.

In generale, dati gli autovettori $|u\rangle , u \in T$ di un operatore unitario U con corrispondenti autovalori ϕ_u, l'algoritmo di stima delle fasi trasforma lo stato normalizzato

$$|0\rangle \, (\sum_{u \in T} d_u |u\rangle)$$

nello stato

$$\sum_{u \in T} d_u \left| \tilde{\phi_u} \right\rangle |u\rangle),$$

dove $\left| \tilde{\phi_u} \right\rangle$ è una buona stima di $|\phi_u\rangle$. In questo caso la probabilità che questa stima sia precisa fino all'n-simo bit è almeno $|d_u|^2(1-\epsilon)$ se t è scelto come in (6).

17.2 Trovare l'ordine di un numero

Dati due interi positivi x ed N che non hanno fattori comuni e tali che $x < N$, si definisce l'*ordine* di x modulo N come il più piccolo intero r tale che $x^r = 1(modN)$.

Non si conoscono algoritmi classici che dati x ed N permettono di determinare r in tempo polinomiale in $L = \lceil \log N \rceil$. Nel seguito descriviamo un algoritmo quantistico che risolve questo problema con un numero di operazioni $O(L^3)$.

Richiamiamo prima le nozioni di aritmetica modulare necessarie per capire il problema e la tecnica utilizzata nell'algoritmo quantistico.

17.2.1 Aritmetica modulo n

Dati due interi positivi x ed n, si indica con $x(mod\ n)$ il resto della divisione di x per n. Più precisamente, x si può scrivere in modo univoco come $x = kn + r$ dove $k \geq 0$ e r è il resto, $0 \leq r \leq n-1$. Le operazioni nell'aritmetica modulo n sono le operazioni aritmetiche di somma, sottrazione, moltiplicazione e divisione dove si prende il risultato modulo n. Una differenza sostanziale è la definizione dell'inverso di un numero modulo n. Infatti, mentre in aritmetica gli unici interi ad avere un inverso sono 1 e -1, in aritmetica modulare ci possono essere altri interi che hanno un inverso. La seguente proposizione stabilisce una condizione necessaria e sufficiente perché un intero abbia un inverso modulo n.

Proposition 17.1 *Un intero x ha un inverso modulo n se e solo se $MCD(x,n) = 1$, dove $MCD(x,n)$ è il massimo comun divisore di x ed n.*

17.2.2 L'algoritmo quantistico

L'algoritmo quantistico per trovare l'ordine di un numero x modulo N consiste essenzialmente nell'applicare il metodo di stima delle fasi all'operatore

unitario U definito da:

$$U|y\rangle = |xy(mod\ N)\rangle \qquad \text{se } 0 \leq y \leq N-1$$
$$U|y\rangle = |y\rangle \quad \text{se } N \leq y \leq 2^L - 1.$$

Si verifica facilmente che, se r è l'ordine di x modulo N e $0 \leq s \leq r-1$, ogni vettore della forma

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi isk/r} \left| x^k (mod\ N) \right\rangle,$$

è un autovettore di U con autovalore $e^{2\pi is/r}$. Infatti, usando l'ipotesi $x^r = 1(mod\ N)$, si ha che:

$$\begin{aligned} U|u_s\rangle &= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi isk/r} \left| x^{k+1} (mod\ N) \right\rangle \\ &= \frac{1}{\sqrt{r}} \sum_{k=1}^{r} e^{-2\pi is(k-1)/r} \left| x^k (mod\ N) \right\rangle \\ &= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi is(k-1)/r} \left| x^k (mod\ N) \right\rangle \\ &= e^{2\pi is/r} |u_s\rangle. \end{aligned}$$

Il penultimo passaggio si spiega osservando che nella sommatoria

$$\sum_{k=0}^{r-1} e^{-2\pi is(k-1)/r} \left| x^k (mod\ N) \right\rangle$$

il valore dell'elemento che si ottiene per $k = 0$ (cioè $e^{2\pi is/r} |1(mod\ N)\rangle$) è esattamente quello che si ottiene nella sommatoria

$$\sum_{k=1}^{r} e^{-2\pi is(k-1)/r} \left| x^k (mod\ N) \right\rangle$$

per $k = r$. Quest'ultimo infatti risulta:

$$e^{-2\pi is(r-1)/r} |x^r (mod\ N)\rangle = e^{-2\pi is} e^{2\pi is/r} |1(mod\ N)\rangle = e^{2\pi is/r} |1(mod\ N)\rangle.$$

L'idea base dell'algoritmo è di ottenere un'approssimazione accurata della fase s/r da cui poter ricavare il valore di r.

Il primo problema da risolvere è preparare lo stato $|u_s\rangle$. Questo problema non è banale dal momento che $|u_s\rangle$ è definito in termini del valore r che vogliamo trovare. Invece di $|u_s\rangle$, prepariamo il secondo registro in input al circuito per la stima delle fasi nello stato $|1\rangle$. Pochi calcoli dimostrano infatti che questo stato corrisponde ad una combinazione lineare di tutti gli autovettori $|u_s\rangle$ di U per $0 \leq s \leq r-1$:

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = \sum_{k=0}^{r-1} \left(\frac{1}{r} \sum_{s=0}^{r-1} e^{-2\pi i s k/r} \right) \left| x^k (mod\ N) \right\rangle ,$$

dove l'espressione in parentesi risulta uguale a 1 se $k = 0$, mentre è 0 per tutti gli altri valori di k. Per vedere questo basta osservare che per $k > 0$ la sommatoria in parentesi è la serie geometrica di ragione $e^{-2\pi i k/r}$ e quindi converge a $\frac{1-(e^{-2\pi i k/r})^r}{1-e^{-2\pi i k/r}} = 0$. Pertanto, la combinazione degli autovettori $|u_s\rangle$ coincide con lo stato $|1\rangle$.

Per ragioni che saranno chiarite in 17.4, prendiamo $n = 2L + 1$. Di conseguenza, il numero di qubits del primo registro dell'algoritmo di stima delle fasi sarà $t = 2L + 1 + \left\lceil \log\left(2 + \frac{1}{2\epsilon}\right) \right\rceil$. Inoltre una misurazione del primo registro produrrà per ogni s, $0 \leq s \leq r-1$, un'approssimazione φ_s di s/r precisa fino al bit $2L+1$ con una probabilità di almeno $(1-\epsilon)/r$ (cf. 17.1).

Dobbiamo ora risolvere ancora due problemi:

Come eseguire la sequenza di controlled-U^{2^k},
Come ricavare r dalla stima φ_s trovata.

17.3 Complessità delle operazioni controllate $C(U^{2^k})$

Nell'algoritmo descritto in 17.1 le operazioni controllate U^{2^k} sono state trattate come "scatole nere". Nell'istanza particolare dell'algoritmo per il nostro problema di trovare l'ordine di un numero x modulo N, la sequenza di controlled-U^{2^k} corrisponde esattamente alla moltiplicazione modulo N del secondo registro per x elevato ad una potenza pari al contenuto del primo registro. Infatti, se $|k\rangle$ è lo stato del primo registro e $|u\rangle$ lo stato del secondo registro, allora dopo l'applicazione della sequenza di controlled-U^{2^k} si ottiene

$$\begin{aligned} |k\rangle |u\rangle &\mapsto |k\rangle U^{k_t 2^{t-1}} U^{k_{t-1} 2^{t-2}} \dots U^{k_1 2^0} |u\rangle \\ &= |k\rangle \left| x^{k_t 2^{t-1}} x^{k_{t-1} 2^{t-2}} \dots x^{k_1 2^0} u (mod\ N) \right\rangle \\ &= |k\rangle \left| x^{k_t 2^{t-1} + k_{t-1} 2^{t-2} \dots k_1 2^0} u (mod\ N) \right\rangle \end{aligned}$$

$$= \quad |k\rangle \left| x^k u (mod\ N) \right\rangle .$$

Questa operazione può essere realizzata usando lo schema di computazione reversibile con un numero di operazioni elementari dell'ordine di L^3. L'idea è di eseguire $t-1 = O(L)$ operazioni di moltiplicazione modulare per calcolare $x^2 (mod\ N)$, $x^4 (mod\ N)$ e così via fino a $x^{2^{t-1}} (mod\ N)$. Ciascuna di queste operazioni costa $O(L^2)$ (se si usano le regole ordinarie per la moltiplicazione binaria). Quindi il calcolo costa in totale $O(L^3)$. A questo punto si usa l'identità

$$x^k u (mod\ N) = x^{k_t 2^{t-1} + k_{t-1} 2^{t-2} \ldots + k_1 2^0} u (mod\ N),$$

per ottenere $x^k u (mod\ N)$ con altre $t-1$ moltiplicazioni reversibili con complessità $O(L^3)$.

17.4 Frazioni continue

Per ricavare r dalla stima φ_s si usa la tecnica delle frazioni continue che illustriamo brevemente nel seguito.

Le frazioni continue permettono di approssimare un qualsiasi numero reale con una sequenza di numeri razionali della forma

$$[a_0, a_1, a_2, \ldots, a_p] = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_{2+\cfrac{1}{\ldots + \cfrac{1}{a_p}}}}},$$

dove a_j sono interi positivi per $j \geq 1$.

Per un numero reale c, si definisce l'espansione in frazioni continue nel seguente modo. A partire da c si costruiscono ricorsivamente le sequenze $\{a_j\}_{j \geq 0}$ e $\{r_j\}_{j \geq 0}$ prendendo $a_0 = \lfloor c \rfloor$, $r_0 = c - a_0$ e per ogni $j \geq 1$, $a_j = \lfloor \frac{1}{r_{j-1}} \rfloor$ e $r_j = \frac{1}{r_{j-1}} - \lfloor \frac{1}{r_{j-1}} \rfloor$. Allora per ogni $j \geq 0$ con $r_j > 0$,

$$c = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_{2+\cfrac{1}{\ldots + \cfrac{1}{a_j + r_j}}}}}.$$

Il numero razionale $[a_0, a_1, a_2, \ldots, a_j]$ è detto il j-esimo convergente di c. Se $r_j = 0$ allora le frazioni continue terminano con a_j e si ottiene l'uguaglianza $c = [a_0, a_1, a_2, \ldots, a_j]$.

Esempio 17.2 *L'espansione in frazioni continue di* $47/13$ *è data da:*

$$\begin{aligned}
\frac{47}{13} &= 3+\frac{8}{13} = 3+\frac{1}{\frac{13}{8}} \\
&= 3+\frac{1}{1+\frac{5}{8}} = 3+\frac{1}{1+\frac{1}{\frac{8}{5}}} \\
&= 3+\frac{1}{1+\frac{1}{1+\frac{3}{5}}} = 3+\frac{1}{1+\frac{1}{1+\frac{1}{\frac{5}{3}}}} \\
&= 3+\frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{2}{3}}}} = 3+\frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{1}{\frac{3}{2}}}}} \\
&= 3+\frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{1}{2}}}}}.
\end{aligned}$$

Pertanto $c = [3,1,1,1,1,2]$.

Elenchiamo alcuni importanti risultati della teoria dei numeri che ci saranno utili per il nostro problema. In particolare, siamo interessati al caso in cui c è un numero razionale (come appunto s/r). In questo caso si dimostra la seguente proposizione:

Proposition 17.3 *L'espansione in frazioni continue di un numero reale* c *termina se e solo se* c *è un numero razionale.*

La seguente proposizione ci permette di ricavare numeratore e denominatore di un numero razionale dalla sua espansione in frazioni continue.

Proposition 17.4 *Il numero razionale* $[a_0, a_1, a_2, \ldots, a_j]$ *è il numero* $\frac{p_j}{q_j}$ *dove* $p_0 = a_0$, $q_0 = 1$, $p_1 = 1 + a_0 a_1$, $q_1 = a_1$ *e per ogni* $j \geq 2$*:*

$$p_j = a_j p_{j-1} + p_{j-2} \text{ e } q_j = a_j q_{j-1} + q_{j-2}.$$

Corollary 17.5 *L'espansione in frazioni continue di un numero razionale positivo* p/q *si può ottenere in* $O(m^3)$ *operazioni, se* p *e* q *sono interi su* m *bits.*

Teorema 17.6 *Supponiamo che* x *e* p/q *siano numeri razionali tali che* $|x - p/q| \leq 1/2q^2$*. Allora* p/q *è un convergente della frazione continua per* x*.*

Questo teorema ci permette di concludere che s/r è un convergente del numero razionale φ_s ottenuto dall'algoritmo di stima delle fasi, purché si scelga una precisione $n = 2L+1$. Infatti, in questo caso, poichè $r \leq N \leq 2^L$, si ha che

$$|\varphi_s - s/r| < 1/2^{2L+1} \leq 1/2r^2.$$

Inoltre, questo convergente s/r si può calcolare con $O(L^3)$ operazioni elementari.

Avendo a disposizione questi risultati possiamo ora completare la descrizione dell'algoritmo quantistico per trovare l'ordine r di un numero positivo x modulo N.

Calcoliamo con il metodo delle frazioni continue un convergente p/q di φ_s, tale che $q \leq 2^L$ e $|\varphi_s - p/q| < 1/2^{2L+1}$ (si dimostra che esiste sempre almeno una frazione con queste proprietà). Il denominatore q è un candidato per r. Si controlla se $x^q = 1(mod\ N)$. Se il test ha successo, allora q è l'ordine di x modulo N. Altrimenti l'algoritmo fallisce e bisogna eseguirlo nuovamente.

Una condizione importante per il successo dell'algoritmo è che r ed s non abbiano fattori comuni, altrimenti il numero restituito dall'algoritmo delle frazioni continue potrebbe essere un fattore di r piuttosto che r stesso. Per N molto grandi, la probabilità che r ed s siano co-primi è almeno $1/\log N$. Basta quindi ripetere l'algoritmo un numero adeguato di volte per ottenere con alta probabilità un istanza di s/r con r ed s co-primi. In particolare, ripetendo l'algoritmo $O(L)$ volte l'algoritmo ha successo con probabilità $(1-\epsilon)(1-1/N)$, con un costo totale di $O(L^4)$ operazioni.

17.5 Fattorizzazione

Dato un numero intero positivo dispari e composto N, il problema della fattorizzazione consiste nel trovare i fattori primi di N.

Determinare se un numero N è primo o composto è un problema computazionalmente "facile": l'algoritmo probabilistico di Miller-Rabin per il test di primalità impiega $O(s \log N)$ operazioni aritmetiche con una probabilità di errore $P \leq 2^{-s}$. Recentemente (nell'estate del 2002) tre ricercatori indiani hanno scoperto un algoritmo deterministico in grado di stabilire in tempo polinomiale (e con certezza) se un numero è primo oppure no, dimostrando che il problema è effettivamente in **P**.

Una volta stabilito che un numero è composto, non sembra altrettanto facile determinare i suoi fattori primi.

I sistemi crittografici oggi più diffusi (come il sistema RSA[10]) sono basati sulla seguente congettura:

Conjecture 17.7 *La fattorizzazione di interi è un problema computazionalmente più difficile della moltiplicazione di interi: mentre esistono molti algoritmi polinomiali per la moltiplicazione di interi, non esistono algoritmi polinomiali per la fattorizzazione di interi.*

Questa assunzione è basata sul fatto che gli sforzi che per molte centinaia di anni hanno impegnato i più grandi scenziati non sono stati sufficienti a produrre un algoritmo polinomiale per la fattorizzazione. Il più efficiente algoritmo classico che si conosca al momento è il cosiddetto "number field sieve" la cui complessità è stata euristicamente dimostrata essere $O(exp\left[c(\log N)^{1/3}(loglogN)^{2/3}\right])$, cioè l'algoritmo richiede un tempo superpolinomiale nel numero di cifre $O(\log N)$ in N.

Verso la metà degli anni '90 Peter Shor ideò (partendo da risultati precedenti dovuti a Benioff, Bennet, Deutsch, Feynmann, Simon ed altri) un algoritmo quantistico in grado di fattorizzare un intero in tempo polinomiale. La complessità di questo algoritmo risulta infatti

$$O\left((\log N)^2(\log\log N)(\log\log\log N)\right),$$

cioè polinomiale nel numero di cifre $O(\log N)$ in N.

L'algoritmo consiste dei cinque passi descritti alla fine di questo capitolo, di cui solo il **Passo 3** richiede l'uso di un computer quantistico (in particolare questo passo invoca la procedura quantistica per trovare l'ordine di un numero descritta in 17.2). Tutti gli altri passi dell'algoritmo possono essere eseguiti su un computer classico.

Richiamiamo brevemente i principali risultati di teoria dei numeri necessari per capire come funziona l'algoritmo e qual è la sua complessità.

Descriviamo per prima cosa l'algoritmo di Euclide per trovare il massimo comun divisore, $MCD(p,q)$, di due interi positivi p e q che si basa sul seguente risultato:

Proposition 17.8 *Dati due interi p e q, supponiamo che r sia il resto della divisione di p per q e che sia $r \neq 0$. Allora*

$$MCD(p,q) = MCD(q,r).$$

[10]RSA è un sistema crittografico a chiavi pubbliche inventato da Rivest, Shamir, Adleman da cui ha preso il nome.

L'algoritmo di Euclide è il seguente: supponendo che $p > q$, si divide p per q e si trova il resto $r_1 < q$. Per la proposizione 17.8, $MCD(p,q) = MCD(q,r_1)$. Si divide quindi q per r_1 e si trova il resto $r_2 < r_1$ con $MCD(q,r_1) = MCD(r_1,r_2)$. Induttivamente, si divide r_{n-2} per r_{n-1} per trovare il resto $r_n < r_{n-1}$ con

$$MCD(p,q) = MCD(q,r_1) = \ldots = MCD(r_n, r_{n+1}).$$

Poichè la sequenza r_j è una sequenza di interi positivi strettamente decrescente, esiste n tale che $r_{n+1} = 0$, cioè r_{n-1} è un multiplo di r_n. Quindi l'algoritmo termina con

$$MCD(p,q) = MCD(r_{n-1}, r_n) = r_n.$$

La complessità dell'algoritmo di Euclide è $O(L^3)$, dove L è il numero di bits necessari per rappresentare p e q. Infatti, l'algoritmo richiede al più $O(L)$ divisioni binarie, ciascuna delle quali richiede $O(L^2)$ operazioni su bits.

Il seguente teorema ci fa vedere come il problema della fattorizzazione si riduce a quello di trovare l'ordine di un numero.

Teorema 17.9 *Supponiamo che N sia un numero composto e che $x \in \{1,\ldots,N\}$ sia una soluzione non banale dell'equazione $x^2 = 1(mod\ N)$, cioè tale che $x \neq 1(mod\ N)$ e $x \neq N-1 = -1(mod\ N)$. Supponiamo inoltre che L sia il numero di bits necessari per rappresentare N ($L = \lceil \log N \rceil$). Allora almeno uno tra $MCD(x-1,N)$ e $MCD(x+1,N)$ è un fattore non banale di N che può essere calcolato in $O(L^3)$ passi.*

Questo risultato combinato con il seguente teorema permette di determinare con alta probabilità un fattore non banale di un qualsiasi numero composto N.

Teorema 17.10 *Supponiamo che la scomposizione in fattori primi del numero intero dispari composto N sia $N = p_1^{n_1} \cdots p_m^{n_m}$. Se x è un intero scelto casualmente tra 1 e N tale che $MCD(x,N) = 1$ e r è l'ordine di x modulo N, allora*

$$P(r \text{ è pari e } x^{r/2} \neq -1(mod\ N)) \geq 1 - 1/2^m.$$

L'algoritmo Per calcolare un fattore non banale di un numero intero N di L bits, dispari e composto si procede come segue:

Passo 1 Scegliamo un numero casuale x tra 1 e $N-1$.

Passo 2 Con l'algoritmo di Euclide calcoliamo il $MCD(x, N)$. Se $MCD(x, N) > 1$ allora abbiamo trovato un fattore non banale di N. Altrimenti si procede con il passo 3.

Passo 3 Usiamo l'algoritmo quantistico descritto in 17.2 per trovare l'ordine r di x modulo N.

Passo 4 Se r è dispari, oppure r è pari e $x^{r/2} = -1(mod\ N)$, allora ritorniamo al passo 1.

Passo 5 Con l'algoritmo di Euclide calcoliamo $MCD(x^{r/2}-1, N)$ e $MCD(x^{r/2}+1, N)$ e se uno dei due interi calcolati risulta essere un fattore non banale di N, allora l'algoritmo termina con successo. Altrimenti fallisce e bisogna ripeterlo a partire dal passo 1.

Algoritmi quantistici di ricerca

Nel 1996 Lov Grover ha introdotto un metodo quantistico che permette di risolvere problemi di ricerca non strutturati fornendo un miglioramento quadratico rispetto alle prestazioni degli algoritmi di ricerca classici esistenti.

18 Problemi di ricerca

I problemi di ricerca costituiscono una numerosa classe di problemi che vanno dalla ricerca in una base di dati all'ordinamento e alla colorazione di un grafo. La forma generale di un problema di ricerca è: "trovare un elemento x in un insieme di possibili soluzioni tale che una certa condizione $P(x)$ è vera". Per esempio, il problema della colorazione di un grafo si può vedere come la ricerca di un assegnamento di colori ai vertici tale che la condizione "vertici adiacenti hanno colori diversi" è soddisfatta. Analogamente un problema di ordinamento equivale alla ricerca di una permutazione che soddisfa la particolare relazione d'ordine desiderata.

Un problema di ricerca non strutturato è un problema di ricerca dove non si conosce la struttura dello spazio delle soluzioni. Per un problema di ricerca strutturato queste informazioni possono invece essere utilizzate permettendo di costruire algoritmi efficienti (e.g. ricerca binaria per problemi di ricerca dove lo spazio delle soluzioni ha una struttura ad albero binario).

Nel caso generale di un problema di ricerca non strutturato, il migliore algoritmo classico che si possa applicare è quello che controlla la condizione

$P(x)$ su ciascuno degli elementi x scelti casualmente nell'insieme delle possibili soluzioni. Se quest'ultimo ha dimensione N, allora l'algoritmo richiede $O(N)$ valutazioni di P. Su un computer quantistico, questi problemi si possono risolvere con una probabilità di errore limitata, con $O(\sqrt{N})$ valutazioni di P, utilizzando il metodo di Grover che descriviamo nel seguito.

Questo metodo è stato dimostrato ottimale per problemi di ricerca completamente non strutturati e la sue applicazioni più importanti e utili sono per ottenere soluzioni più rapide per problemi **NP**-completi.

19 L'algoritmo di Grover

Assumiamo per convenienza che il numero delle soluzioni candidate del problema di ricerca sia $N = 2^n$, con $n \geq 1$ (nei problemi concreti n è in genere molto grande) e che ogni elemento sia una sequenza di n bits. Assumiamo inoltre che il numero delle soluzioni reali sia esattamente M e che esista un oracolo che determina se una data sequenza di n bits è una soluzione oppure no. Tale oracolo corrisponde ad una trasformazione unitaria che implementa una particolare funzione booleana $f : \{0,1\}^n \mapsto \{0,1\}$ tale che $f(x) = 1$ se x è una soluzione e $f(x) = 0$ altrimenti, e ha la forma:

$$O : |x\rangle\,|q\rangle \mapsto |x\rangle\,|q \oplus f(x)\rangle\,,$$

dove $x \in \{0,1\}^n$ e $|q\rangle$ è un singolo qubit. Se inizializzato a 0, q diventa 1 quando $f(x) = 1$. Se $|q\rangle$ è preparato nello stato $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, allora l'azione dell'oracolo O è quella di invertire le ampiezze degli stati $|x\rangle$ che sono soluzioni, lasciando invariati gli altri stati:

$$O : |x\rangle\,\frac{|0\rangle - |1\rangle}{\sqrt{2}} \mapsto (-1)^{f(x)}\,|x\rangle\,\frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

Poichè il qubit $|q\rangle$ non viene modificato da O possiamo ignorarlo. L'azione di O su un generico stato quantistico è quindi:

$$O : \sum_{x\in\{0,1\}^n} \alpha_x\,|x\rangle \mapsto \sum_{x\in\{0,1\}^n} (-1)^{f(x)}\alpha_x\,|x\rangle\,.$$

L'algoritmo di Grover è definito su un input preparato nello stato $|0\rangle^{\otimes n}$. Viene quindi applicata la trasformata di Hadamard per ottenere la sovrapposizione equiprobabile di stati

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x\in\{0,1\}^n} |x\rangle\,.$$

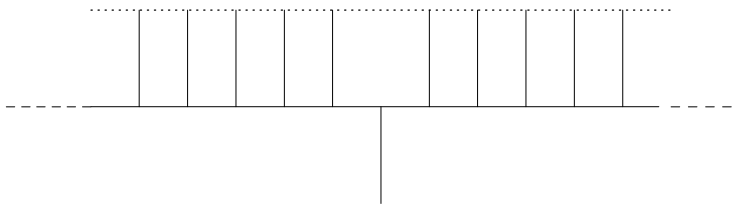

Figure 29: Effetto dell'oracolo O sulle ampiezze

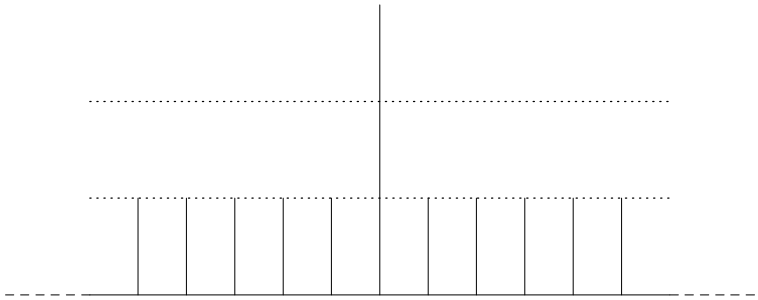

Figure 30: Inversione intorno alla media delle ampiezze

A questo punto si ripete l'applicazione per un numero appropriato di volte del seguente operatore, detto operatore di Grover,

$$G = H^{\otimes n} P_0 H^{\otimes n} O,$$

dove P_0 effettua uno shift di fase di -1 su tutti gli stati computazionali diversi da $|0\rangle$, cioè

$$P_0 : |x\rangle \mapsto \begin{cases} |x\rangle & \text{se } x = 0 \\ -|x\rangle & \text{se } x > 0 \end{cases},$$

per ogni $0 \leq x \leq N-1$. Si verifica facilmente che $P_0 = 2|0\rangle\langle 0| - I$.

L'operazione $H^{\otimes n} P_0 H^{\otimes n}$ è detta "inversione intorno alla media" perché il suo effetto è quello di amplificare le ampiezze degli stati soluzione (che erano state moltiplicate per -1 dall'oracolo, cf. Figura 29) innalzandole del doppio al di sopra della media di tutte le ampiezze (cf. Figura 30).

Formalmente tale operazione corrisponde a

$$H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n} = 2|\psi\rangle\langle\psi| - I,$$

e la sua azione su un generico stato $\sum_x \alpha_x |x\rangle$ è

$$(2|\psi\rangle\langle\psi| - I)(\sum_x \alpha_x |x\rangle) = \sum_x (2A - \alpha_x)|x\rangle,$$

dove $A = \sum_x \alpha_x/N$. La dimostrazione segue dall'osservazione che la matrice $N \times N$ definita da:

$$D = \begin{bmatrix} \frac{2}{N} - 1 & \frac{2}{N} & \dots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} - 1 & \dots & \frac{2}{N} \\ \dots & \dots & \dots & \dots \\ \frac{2}{N} & \frac{2}{N} & \dots & \frac{2}{N} - 1 \end{bmatrix}$$

è unitaria ed effettua la trasformazione $\sum_x \alpha_x |x\rangle \mapsto \sum_x (2A - \alpha_x)|x\rangle$. Questa matrice corrisponde esattamente a $H^{\otimes n}(2|0\rangle\langle 0|)H^{\otimes n} - I$. Infatti

$$H^{\otimes n}(2|0\rangle\langle 0|)H^{\otimes n} = \frac{2}{N}\sum_{x,y}|x\rangle\langle y| = \begin{bmatrix} \frac{2}{N} & \dots & \frac{2}{N} \\ \dots & \frac{2}{N} & \dots \\ \frac{2}{N} & \dots & \frac{2}{N} \end{bmatrix}.$$

Possiamo quindi scrivere l'operatore di Grover come

$$G = (2|\psi\rangle\langle\psi| - I)O.$$

19.1 Interpretazione geometrica di G

L'operatore di Grover corrisponde ad una rotazione dello stato iniziale $|\psi\rangle$ nel piano reale generato dai due stati normalizzati

$$|\sigma\rangle = \frac{1}{\sqrt{N-M}}\sum_{x\in S}|x\rangle \text{ e } |\tau\rangle = \frac{1}{\sqrt{M}}\sum_{x\in T}|x\rangle,$$

dove $T = \{x \in \{0,1\}^n \mid f(x) = 1\}$ e $S = \{0,1\}^n \setminus T$. Questi due stati corrispondono a sovrapposizioni uniformi rispettivamente degli stati che non sono soluzioni e di quelli che sono soluzioni. Si verifica facilmente che $|\psi\rangle$ appartiene al piano generato da $|\sigma\rangle$ e $|\tau\rangle$, cioè che

$$|\psi\rangle = \sqrt{\frac{N-M}{N}}|\sigma\rangle + \sqrt{\frac{M}{N}}|\tau\rangle.$$

Per vedere che G è una rotazione, osserviamo che l'azione dell'oracolo O in questo piano è una riflessione intorno all'asse $|\sigma\rangle$. Infatti, per ogni stato nel piano, $a|\sigma\rangle + b|\tau\rangle$ (con $a^2 + b^2 = 1$), $O(a|\sigma\rangle + b|\tau\rangle) = a|\sigma\rangle - b|\tau\rangle$.

Esercizio 19.1 *Dimostra che l'azione di* $2|\psi\rangle\langle\psi| - I$ *nel piano* $|\sigma\rangle$, $|\tau\rangle$ *è una riflessione intorno a* $|\psi\rangle$.

Pertanto G è la composizione di due riflessioni e il seguente teorema della geometria piana ci assicura che questa composizione corrisponde ad una rotazione.

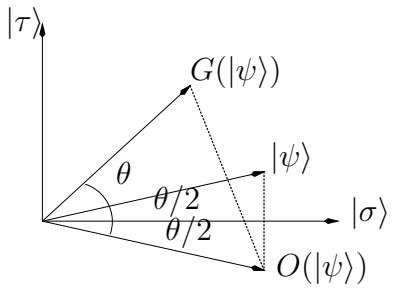

Figure 31: Singola iterazione dell'operatore G

Teorema 19.2 *Se L_1 e L_2 sono due rette nel piano euclideo $\mathbb{R}^2$ che si intersecano in un punto P, e se β è l'angolo tra L_1 e L_2, allora una riflessione in L_1 seguita da una riflessione in L_2 è una rotazione di un angolo 2β intorno al punto P.*

Allora, se $\theta/2$ è l'angolo tra $|\psi\rangle$ e $|\sigma\rangle$ (cf. Figura 31), l'operatore G ruota i vettori nel piano $|\sigma\rangle$, $|\tau\rangle$ di un angolo θ verso $|\tau\rangle$. Poichè $|\psi\rangle = \cos\theta/2\,|\sigma\rangle + \sin\theta/2\,|\tau\rangle$, l'angolo $\theta/2$ è tale che $\cos\theta/2 = \sqrt{(N-M)/N}$ e $\sin\theta/2 = \sqrt{M/N}$.

Esercizio 19.3 *Dimostrare che nella base $|\sigma\rangle$, $|\tau\rangle$ l'operatore G ha la matrice di rappresentazione*

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix},$$

dove θ è un numero reale in $[0, \pi/2]$ tale che $\sin\theta/2 = \sqrt{M/N}$.

Dopo m iterazioni abbiamo:

$$G^m\,|\psi\rangle = \cos\left(\frac{2m+1}{2}\theta\right)|\sigma\rangle + \sin\left(\frac{2m+1}{2}\theta\right)|\tau\rangle\,.$$

Se $\frac{2m+1}{2}\theta \approx \pi/2$, cioè dopo

$$m = \lfloor \pi/2\theta - 1/2 \rceil.$$

iterazioni (dove $\lfloor x \rceil$ denota l'intero più vicino al numero reale x), l'angolo tra $|\psi\rangle$ e $|\sigma\rangle$ è $\pi/4$, che è quindi uguale all'angolo tra $|\psi\rangle$ e $|\tau\rangle$, poichè i due angoli sono complementari. Quindi una misurazione di $|\psi\rangle$ ci darebbe una soluzione con probabilità $\sin^2\pi/4 = 1/2$.

In generale, dall'equazione precedente segue che $m \leq \lceil \pi/2\theta \rceil$ (dove $\lceil x \rceil$ denota la parte intera alta di x). Poichè per ogni $\theta \in \mathbb{R}$

$$\frac{\theta}{2} \geq \sin \frac{\theta}{2} = \sqrt{\frac{M}{N}},$$

si ottiene il seguente limite superiore per il numero di iterazioni di G necessarie per trovare una soluzione:

$$m \leq \left\lceil \frac{\pi}{4} \sqrt{\frac{N}{M}} \right\rceil .$$

19.2 Complessità

La complessità dell'algoritmo di Grover è data essenzialmente dal numero m di iterazioni dell'operatore G che, come abbiamo visto precedentemente, è $O\left(\sqrt{\frac{N}{M}}\right)$. In ogni iterazione viene applicata due volte la trasformata di Hadamard che richiede $O(n) = O(\log N)$ operazioni su un singolo qubit. Quindi complessivamente l'algoritmo di Grover ha un costo computazionale di $O\left(\sqrt{\frac{N}{M}} \log N\right)$.

References

[1] A. Bernasconi and B. Codenotti. *Introduzione alla Complessitá Computazionale*. Springer-Verlag, Milano, Italy, 1998.

[2] Ethan Bernstein and Umesh V. Vazirani. Quantum complexity theory. *SIAM J. Comput.*, 26(5):1411–1473, 1997.

[3] A.Yu. Kitaev, A.H. Shen, and M.N. Vyalyi. *Classical and Quantum Computation*. Cambridge University Press, Cambridge, UK, 2000.

[4] M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, UK, 2000.

[5] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[6] A. Yao. Quantum circuit complexity. In *Proceedings of the 34th Annual Symposium on Foundations of Computer Science*, pages 352–361, Los Alamitos, CA, 1993. IEEE Computer Society Press.