Corso di

Metodi della Fisica Quantistica per l'Ingegneria

Dispensa

Prof. Vincenzo Berardi

Corso di Laurea in Ingegneria Informatica e dell'Automazione

Politecnico di Bari

Anno accademico: 2026/27

Ultima modifica: 18 Agosto 2026

Nota metodologica

Questa dispensa è stata sviluppata attraverso un processo di lavoro nel quale **ChatGPT (OpenAI)** è stato impiegato come strumento di assistenza alla progettazione didattica, alla rielaborazione e revisione dei testi, al controllo della coerenza tra argomenti, esempi e lezioni, nonché alla preparazione tecnica dei materiali in LaTeX.
L'intelligenza artificiale è stata utilizzata come supporto al processo di elaborazione, non come fonte autonoma né come sostituto della valutazione scientifica. La scelta, la verifica e l'approvazione finale dei contenuti restano responsabilità dell'autore.

Indice

Capitolo 1

Calcolo classico e calcolo quantistico

Questo capitolo introduce il rapporto tra calcolo classico e calcolo quantistico da una prospettiva storica, concettuale e computazionale. L'obiettivo non è presentare il computer quantistico come sostituto generale del computer classico, ma chiarire perché i due paradigmi descrivano modalità diverse di rappresentare, trasformare e leggere l'informazione.

Un calcolo è una procedura finita che trasforma dati iniziali in risultati. Questa definizione, apparentemente astratta, contiene due aspetti distinti.

Il primo è *logico*: occorre specificare quali operazioni siano ammesse, in quale ordine debbano essere applicate e quando la procedura debba terminare. Il secondo è *fisico*: ogni algoritmo deve essere eseguito da un dispositivo reale, costituito da materia, energia e segnali che evolvono secondo determinate leggi fisiche. La separazione fra algoritmo e supporto materiale è estremamente utile, ma non è assoluta. Un modello di calcolo non descrive soltanto simboli e regole; incorpora, almeno implicitamente, una certa idea di quali trasformazioni fisiche siano realizzabili.

Nel calcolo classico l'informazione elementare è rappresentata mediante il *bit*, che assume uno dei due valori 0 e 1. Nel calcolo quantistico l'unità elementare è il quantum-bit (*qubit*), il cui stato può essere una sovrapposizione coerente degli stati di base $|0\rangle$ e $|1\rangle$. Questa differenza non riguarda soltanto il supporto usato per memorizzare i dati. Cambia l'insieme delle trasformazioni ammesse, cambia il modo in cui sistemi composti rappresentano informazione e cambia il ruolo della misurazione.

È quindi opportuno evitare due errori opposti. Il primo consiste nel pensare che un computer quantistico sia semplicemente un computer classico molto più veloce. Il secondo consiste nel ritenere che i due paradigmi non abbiano alcun rapporto. Un computer quantistico deve essere controllato, programmato e interrogato da apparati classici; inoltre può simulare computazioni classiche reversibili. Al tempo stesso, la sua dinamica sfrutta strutture fisiche e matematiche che non hanno un equivalente diretto nella logica booleana ordinaria[1].

1.1 Dalle procedure algoritmiche alla macchina universale

Le procedure algoritmiche precedono di molti secoli la costruzione dei computer. Metodi sistematici per l'aritmetica, la geometria, la soluzione di equazioni e il calcolo astronomico furono sviluppati

[1] La logica booleana è un sistema logico in cui le variabili possono assumere soltanto due valori, convenzionalmente indicati come *vero* e *falso* (oppure 1 e 0). Le operazioni fondamentali sono la negazione (NOT), la congiunzione (AND) e la disgiunzione (OR), dalle quali possono essere costruite espressioni logiche più complesse. Essa costituisce la base logica del funzionamento dei circuiti digitali e del calcolo classico.

in civiltà molto diverse. Tuttavia, fino al XX secolo mancava una definizione matematica generale di *procedura effettiva*: si sapeva eseguire un algoritmo, ma non era chiaro come caratterizzare formalmente tutto ciò che potesse essere calcolato.

Negli anni Trenta del Novecento questo problema fu affrontato con formalismi diversi. Tra essi vi furono il calcolo lambda di Alonzo Church[2], le funzioni ricorsive e la macchina astratta proposta da Alan Turing nel 1936. Sebbene nati da impostazioni differenti, questi modelli risultarono equivalenti per quanto riguarda la classe delle funzioni calcolabili.

La macchina di Turing

Una macchina di Turing è costituita idealmente da:

- un nastro suddiviso in celle, potenzialmente illimitato;
- un alfabeto finito di simboli;
- una testina capace di leggere e scrivere un simbolo;
- un insieme finito di stati interni;
- una funzione di transizione che determina, a ogni passo, il nuovo simbolo, lo spostamento della testina e il nuovo stato interno.

La semplicità del modello è intenzionale. Turing non cercava di descrivere l'architettura di un elaboratore concreto, ma di isolare le operazioni minime necessarie a rappresentare una procedura algoritmica. Il risultato decisivo è la possibilità di costruire una *macchina universale*: una singola macchina capace di simulare qualunque altra macchina di Turing, purché riceva come parte dell'input la descrizione della macchina da simulare.

Questa idea contiene il principio del computer programmabile. Il programma non deve essere incorporato una volta per tutte nella struttura meccanica della macchina; può essere codificato come informazione e caricato insieme ai dati.

La tesi di Church–Turing afferma, in forma informale, che ogni funzione effettivamente calcolabile mediante una procedura algoritmica può essere calcolata da una macchina di Turing. Non è un teorema matematico ordinario, perché mette in relazione un concetto intuitivo — "procedura effettiva" — con un formalismo preciso. La sua forza deriva dall'equivalenza fra molti modelli indipendenti e dall'assenza di procedure algoritmiche convincenti che eccedano il modello di Turing.

Occorre distinguere la tesi originaria da una versione più forte, relativa non soltanto alla calcolabilità, ma anche all'efficienza:

> Ogni modello fisicamente realistico di calcolo può essere simulato efficientemente da una macchina di Turing probabilistica.

L'avverbio *efficientemente* è cruciale. Una simulazione può esistere in linea di principio, ma richiedere risorse così grandi da essere impraticabile. Il calcolo quantistico non è normalmente presentato come una confutazione della tesi originaria: un computer classico può simulare un computer quantistico, almeno con precisione finita. La difficoltà è che, per sistemi generali, il

[2] R. Rojas, *A Tutorial Introduction to the Lambda Calculus*, 2015, arXiv:1503.09060.

costo della simulazione classica cresce molto rapidamente, tipicamente in modo esponenziale con il numero di qubit.

1.2 Dal modello astratto al computer classico

Negli anni Quaranta, il lavoro di John von Neumann e di altri pionieri rese operativa l'idea del calcolatore programmabile. Nell'architettura a programma memorizzato, istruzioni e dati sono rappresentati nella memoria della macchina. Un'unità di controllo preleva le istruzioni, le decodifica e coordina le operazioni dell'unità aritmetico-logica.

Il modello moderno di calcolo classico può essere descritto a livelli diversi:

1. **livello logico**: bit, porte booleane e circuiti;
2. **livello architetturale**: registri, memoria, processore, gerarchia di cache, dispositivi di ingresso e uscita;
3. **livello fisico**: tensioni, correnti, transistor e materiali semiconduttori;
4. **livello software**: linguaggi, compilatori, sistemi operativi, librerie e applicazioni.

La possibilità di separare questi livelli è uno dei maggiori successi del calcolo classico. Il programmatore può ragionare su variabili e strutture dati senza seguire il moto di ogni carica nel circuito; il progettista di hardware può ottimizzare transistor e interconnessioni senza conoscere l'applicazione finale.

L'invenzione del transistor nel 1947 rese possibile sostituire dispositivi ingombranti e fragili con componenti più piccoli, veloci e affidabili. La successiva integrazione di molti transistor sullo stesso circuito diede origine alla microelettronica moderna.

La cosiddetta legge di Moore descrisse per decenni la crescita, approssimativamente esponenziale, del numero di transistor integrabili in un circuito. Non si tratta di una legge fondamentale della natura, ma di una regolarità tecnologica sostenuta da investimenti, miglioramenti nei processi produttivi e innovazioni architetturali. La miniaturizzazione ha aumentato la potenza di calcolo disponibile e ridotto il costo per operazione.

La progressiva riduzione delle dimensioni rende però sempre più rilevanti effetti dissipativi, fluttuazioni, tunnelling e altri fenomeni microscopici. Ciò non implica che l'informatica classica sia destinata a scomparire. Indica piuttosto che la crescita delle prestazioni non può essere affidata indefinitamente allo stesso meccanismo di miniaturizzazione e che diventano necessarie architetture parallele, acceleratori specializzati e nuovi materiali.

Un circuito classico trasforma stringhe di bit mediante porte logiche. Operazioni come NOT, AND, OR e NAND implementano funzioni booleane. Un insieme di porte è detto *universale* se permette di costruire qualunque funzione booleana finita. Per esempio, NAND da sola è universale.

Questa universalità non significa che tutte le funzioni siano calcolate con lo stesso costo. Due circuiti possono produrre lo stesso risultato ma usare un numero molto diverso di porte o una profondità molto diversa. Il problema centrale passa quindi dalla possibilità del calcolo alla quantità di risorse necessarie.

1.3 Computazione reversibile

Molte porte classiche ordinarie sono irreversibili. Dall'uscita di una porta AND non è possibile ricostruire in modo univoco i due bit di ingresso. La porta NOT, invece, è reversibile: applicandola due volte si recupera il bit iniziale.

Per il calcolo quantistico la reversibilità è importante perché l'evoluzione di un sistema quantistico chiuso è descritta da trasformazioni unitarie, quindi invertibili (vedi appendice A). Tuttavia, ogni computazione classica può essere incorporata in una computazione reversibile introducendo registri ausiliari. Una funzione $f : \{0,1\}^n \to \{0,1\}^m$ può essere implementata nella forma

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

dove $\oplus$ indica la somma bit a bit modulo due. Questa trasformazione è invertibile perché conserva l'ingresso x e combina reversibilmente il risultato con il registro y.

La relazione fra calcolo classico reversibile e calcolo quantistico mostra che il paradigma quantistico non elimina la logica classica: la contiene come caso particolare, ma opera in uno spazio degli stati più ricco.

Per un input di lunghezza n, si studia come crescono il tempo di esecuzione, la memoria e altre risorse. Un algoritmo è normalmente considerato efficiente se il numero di passi cresce al più come un polinomio:

$$T(n) = O(n^k)$$

per qualche costante k. Una crescita esponenziale, come $O(2^n)$, diventa rapidamente proibitiva.

La classe P contiene, in forma semplificata, i problemi decisionali risolvibili in tempo polinomiale da una macchina deterministica. La classe BPP contiene quelli risolvibili in tempo polinomiale da un algoritmo probabilistico con probabilità di errore limitata. Nel paradigma quantistico la classe analoga è BQP: problemi risolvibili in tempo polinomiale da un algoritmo quantistico con errore limitato.

Sono note le inclusioni

$$\mathrm{P} \subseteq \mathrm{BPP} \subseteq \mathrm{BQP},$$

ma non è dimostrato in generale che le inclusioni siano strette. I vantaggi quantistici noti derivano da algoritmi specifici e da risultati in modelli particolari; non equivalgono a una dimostrazione completa che $\mathrm{BPP} \neq \mathrm{BQP}$.

1.4 Perché il calcolo classico è così efficace

Prima di discutere i vantaggi quantistici è essenziale riconoscere la forza del paradigma classico. La tecnologia classica non è un'approssimazione obsoleta del calcolo quantistico. È il metodo più efficace per una vastissima classe di compiti.

Stabilità dell'informazione

Un bit classico può essere rappresentato da due intervalli macroscopicamente distinguibili di tensione, carica o magnetizzazione. Non è necessario controllare con precisione assoluta ogni variabile fisica: basta che il segnale rimanga entro la regione associata a 0 o a 1. La rigenerazione digitale permette di correggere piccole perturbazioni prima che si accumulino.

Questa robustezza è alla base dell'affidabilità dei circuiti digitali. Copiare un bit non pone alcun vincolo fondamentale; lo stesso valore può essere replicato, amplificato e distribuito. Nel calcolo quantistico, al contrario, il *no-cloning theorem*[3] impedisce la copia perfetta di uno stato quantistico arbitrario sconosciuto.

L'informazione classica può normalmente essere letta senza alterare in modo essenziale il valore logico memorizzato. Ciò rende naturale ispezionare stati intermedi, eseguire diagnostica, creare registri di controllo e interrompere un programma sulla base di risultati parziali.

La misurazione quantistica ha un ruolo differente. In generale restituisce un risultato probabilistico e modifica lo stato. Per questo un algoritmo quantistico deve orchestrare le ampiezze prima della misura finale; non è possibile leggere liberamente tutte le componenti di una sovrapposizione.

Maturità tecnologica e software

Il calcolo classico dispone di:

- processi di fabbricazione industriale altamente ottimizzati;
- sistemi di memoria ad alta capacità;
- protocolli affidabili di comunicazione;
- compilatori, sistemi operativi e linguaggi maturi;
- algoritmi efficienti sviluppati in decenni di ricerca;
- infrastrutture distribuite e acceleratori specializzati.

Questa stratificazione riduce drasticamente il costo pratico di trasformare un algoritmo in un'applicazione. Anche qualora un sottoproblema ammetta un vantaggio quantistico, preparazione dei dati, controllo, post-elaborazione e interazione con l'utente rimangono prevalentemente classici.

Molti compiti quotidiani — ordinamento, gestione di basi di dati, elaborazione di testi, controllo di dispositivi, grafica, comunicazione, calcolo numerico standard — sono già trattati molto efficacemente da algoritmi classici. Per tali compiti un computer quantistico non offre automaticamente un beneficio. L'overhead necessario a codificare i dati, proteggere i qubit e ottenere un risultato può superare qualunque eventuale riduzione nel numero astratto di operazioni.

Il vantaggio del calcolo classico è quindi sia teorico sia ingegneristico: opera con variabili direttamente osservabili, tollera bene il rumore, dispone di memoria persistente e implementa in modo economico la maggior parte delle trasformazioni richieste dall'informatica ordinaria.

[3] Il *no-cloning theorem* afferma che non esiste alcuna trasformazione unitaria capace di produrre una copia perfetta di uno stato quantistico arbitrario e sconosciuto. In particolare, non è possibile realizzare universalmente una trasformazione del tipo

$$|\psi\rangle|0\rangle \longrightarrow |\psi\rangle|\psi\rangle$$

per ogni stato $|\psi\rangle$. Il risultato deriva dalla linearità dell'evoluzione quantistica ed è una delle differenze fondamentali tra informazione classica e informazione quantistica.

1.5 Il computer quantistico: qubit e sovrapposizione

La meccanica quantistica descrive sistemi microscopici mediante vettori in uno spazio di Hilbert[4], trasformazioni unitarie e regole probabilistiche di misurazione. Se un computer è un sistema fisico, è naturale chiedersi se l'informazione possa essere elaborata direttamente secondo queste leggi, anziché costringere il dispositivo a comportarsi come una macchina classica.

La domanda divenne concreta tra la fine degli anni Settanta e l'inizio degli anni Ottanta, quando furono chiariti i rapporti fra informazione, reversibilità e dissipazione. Paul Benioff descrisse una macchina di Turing in termini quantomeccanici; Richard Feynman osservò che la simulazione di sistemi quantistici generali sembra richiedere a una macchina classica una quantità di risorse che cresce esponenzialmente. Propose quindi di usare un sistema quantistico controllabile per simulare un altro sistema quantistico.

Nel 1985 David Deutsch formulò un modello di computer quantistico universale. La questione non era più soltanto costruire un simulatore speciale, ma identificare un modello generale di computazione coerente con le leggi quantistiche.

I primi esempi di Deutsch e Deutsch–Jozsa mostrarono che l'interferenza quantistica può ridurre il numero di interrogazioni a una funzione in problemi costruiti appositamente. Il passaggio decisivo avvenne nel 1994 con l'algoritmo di Peter Shor per la fattorizzazione e il logaritmo discreto. Questi problemi non sono noti come risolvibili in tempo polinomiale con algoritmi classici, mentre l'algoritmo quantistico li tratta efficientemente nel modello ideale.

Nel 1996 Lov Grover mostrò che la ricerca non strutturata in uno spazio di N possibilità può essere eseguita con ordine $\sqrt{N}$ interrogazioni, invece delle N richieste nel caso classico non strutturato. Il vantaggio è quadratico, non esponenziale, ma si applica a un modello molto generale.

Questi risultati cambiarono la prospettiva. Il computer quantistico non era più soltanto un modo naturale per simulare la fisica quantistica; diventava un modello capace, almeno per alcuni problemi, di ridurre radicalmente le risorse computazionali.

Lo stato puro di un qubit può essere scritto come

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \qquad \alpha, \beta \in \mathbb{C}, \qquad |\alpha|^2 + |\beta|^2 = 1.$$

I coefficienti α e β sono ampiezze, non probabilità. Le probabilità dei risultati 0 e 1 in una misura nella base computazionale sono rispettivamente $|\alpha|^2$ e $|\beta|^2$.

La sovrapposizione non deve essere interpretata come una memoria classica che contiene contemporaneamente due valori leggibili. Una misura produce un solo risultato. L'utilità computazionale nasce dal fatto che le ampiezze possono essere trasformate coerentemente e possono interferire.

I vantaggi specifici del calcolo quantistico

Il vantaggio più direttamente connesso alla fisica del dispositivo riguarda la simulazione di sistemi quantistici. Un sistema di molte particelle può richiedere uno spazio degli stati di dimensione esponenziale. Una macchina classica deve rappresentare esplicitamente o approssimare tale struttura; un computer quantistico usa invece un sistema quantistico controllato la cui dinamica vive naturalmente in uno spazio di Hilbert.

[4]Uno spazio di Hilbert è uno spazio vettoriale, generalmente complesso, dotato di un prodotto scalare e completo rispetto alla norma da esso indotta. In meccanica quantistica rappresenta lo spazio matematico degli stati del sistema.

Ciò non elimina tutte le difficoltà. Occorre preparare lo stato iniziale, implementare l'evoluzione con sufficiente precisione e misurare osservabili significative. Tuttavia, per classi importanti di Hamiltoniane, la simulazione quantistica offre una via polinomiale dove la descrizione classica diretta richiede risorse esponenziali.

Le applicazioni potenziali includono fisica della materia condensata, chimica quantistica, scienza dei materiali e dinamica di sistemi a molti corpi. In queste aree il vantaggio non deriva dall'adattare un algoritmo classico a un processore più veloce, ma dalla corrispondenza fra il sistema da simulare e il dispositivo che esegue la simulazione.

Fattorizzazione e logaritmo discreto

L'algoritmo di Shor riduce la fattorizzazione alla ricerca dell'ordine di un elemento modulo N e usa la trasformata di Fourier quantistica per estrarre la periodicità. Nel modello ideale, il numero di operazioni cresce polinomialmente con il numero di bit di N.

Il confronto corretto è con i migliori algoritmi classici conosciuti, non con la divisione per tentativi. Il vantaggio atteso è comunque superpolinomiale. La rilevanza pratica deriva dal fatto che la difficoltà della fattorizzazione e del logaritmo discreto sostiene sistemi crittografici ampiamente studiati.

È importante formulare la conclusione con precisione: Shor non dimostra che ogni problema difficile per il calcolo classico diventi facile su un computer quantistico. Dimostra che due problemi strutturati, fondamentali in teoria dei numeri e crittografia, ammettono algoritmi quantistici drasticamente più efficienti dei migliori metodi classici noti.

Ricerca non strutturata

Dato un insieme di N elementi e un *oracle* che riconosce la soluzione, Grover trova un elemento marcato con $O(\sqrt{N})$ *query*. Un metodo classico, senza struttura aggiuntiva, richiede $O(N)$ *query* nel caso peggiore.

Il guadagno quadratico è meno spettacolare di quello di Shor, ma è robusto: Grover stabilisce un miglioramento generale per la ricerca non strutturata e può essere incorporato in procedure più ampie. Al tempo stesso, il limite quadratico mostra che il computer quantistico non può trasformare ogni ricerca esaustiva in un algoritmo polinomiale.

Stima di proprietà globali

Molti algoritmi quantistici non cercano di produrre l'intera descrizione di un oggetto, ma di stimarne una proprietà globale: una fase, un periodo, un valore medio, un'ampiezza o una caratteristica spettrale. La stima di fase è un esempio : data una trasformazione unitaria U e un suo autostato $|u\rangle$,

$$U|u\rangle = e^{2\pi i\varphi}|u\rangle,$$

un circuito quantistico può stimare φ mediante interferenza e trasformata di Fourier inversa.

Questa impostazione è congeniale al formalismo quantistico, dove l'informazione può essere codificata in fasi relative che non sono direttamente osservabili ma diventano accessibili attraverso opportuni circuiti.

1.6 Quando si perde il vantaggio (non tutti i problemi sono accelerati)

Un computer quantistico universale può simulare un circuito classico reversibile, ma ciò non implica che esista un algoritmo quantistico più veloce per ogni problema. Per molte attività non è noto alcun vantaggio; per altre esistono limiti che escludono accelerazioni esponenziali.

La domanda corretta non è "il quantistico è più veloce del classico?", ma:

> Per quale problema, con quale modello di accesso ai dati, con quale precisione e includendo quali costi, un algoritmo quantistico usa meno risorse del migliore algoritmo classico disponibile?

Il costo di ingresso e uscita

Se un algoritmo quantistico richiede di caricare una grande base di dati classica in uno stato quantistico, il costo della preparazione può annullare il vantaggio teorico. Analogamente, l'uscita di un circuito è ottenuta tramite misure: non si possono estrarre gratuitamente tutte le 2^n ampiezze di uno stato a n qubit.

Un vantaggio significativo richiede quindi che:

- l'input sia già quantistico, oppure sia preparabile efficientemente;
- l'algoritmo estragga una quantità limitata ma rilevante di informazione;
- il numero di ripetizioni necessarie per la stima sia controllato;
- il costo totale, non soltanto il nucleo quantistico, sia favorevole.

Rumore e correzione degli errori

Gli stati quantistici sono sensibili all'interazione con l'ambiente. Errori di fase, rilassamento, imperfezioni delle porte e letture inaccurate degradano la computazione. La ridondanza classica non può essere applicata copiando direttamente un qubit arbitrario, perché ciò violerebbe il teorema di non clonazione.

La correzione quantistica degli errori aggira il problema codificando l'informazione logica in stati entangled di molti qubit fisici e misurando *sindromi* che rivelano l'errore senza leggere direttamente l'informazione protetta. Il teorema della soglia stabilisce, in modelli appropriati, che una computazione arbitrariamente lunga è possibile se il tasso di errore elementare è inferiore a una certa soglia e se si accetta un overhead di risorse.

Questa possibilità rende il calcolo quantistico concettualmente diverso da un calcolatore analogico ideale la cui precisione scompare in presenza di rumore. Tuttavia, l'overhead della tolleranza ai guasti è uno dei principali costi ingegneristici del paradigma quantistico.

Confronto con i migliori algoritmi classici

Un presunto vantaggio quantistico deve essere confrontato con lo stato dell'arte classico. Algoritmi di approssimazione, metodi randomizzati, parallelismo massivo, rappresentazioni tensoriali e

acceleratori specializzati possono ridurre drasticamente il costo di problemi che, in una formulazione ingenua, sembrano intrattabili.

Il vantaggio quantistico non è una proprietà assoluta del problema isolato; dipende dal confronto fra famiglie di algoritmi, dal modello di precisione e dalle risorse contabilizzate.

Aspetto	**Calcolo classico**	**Calcolo quantistico**
Unità di informazione	Bit in uno stato logico 0 o 1	Qubit in una sovrapposizione $\alpha\|0\rangle + \beta\|1\rangle$
Trasformazioni	Porte booleane, spesso irreversibili; circuiti reversibili possibili	Trasformazioni unitarie reversibili, seguite da misurazioni
Lettura	Il valore logico può essere copiato e letto ripetutamente	La misura è probabilistica e in generale modifica lo stato
Copia	La duplicazione dei bit è un'operazione fondamentale	Uno stato quantistico sconosciuto arbitrario non può essere clonato
Rumore	Elevata robustezza digitale e correzione consolidata	Decoerenza e errori richiedono codici quantistici e forte overhead
Spazio degli stati	Una configurazione di n bit alla volta, o distribuzioni classiche	Sovrapposizioni in uno spazio di dimensione 2^n
Punti di forza	Affidabilità, maturità, memoria, controllo, costo e generalità pratica	Simulazione quantistica, interferenza, periodicità, ricerca quadratica, protocolli quantistici
Limite principale	Crescita proibitiva delle risorse per alcuni problemi e simulazioni	Preparazione, controllo, misura, rumore e assenza di vantaggio universale

Tabella 1.1: Confronto concettuale fra i due paradigmi.

La tabella non deve essere letta come una graduatoria. Le caratteristiche che rendono il calcolo classico robusto — copia, osservabilità e dissipazione controllata — non sono semplicemente difetti da superare. Sono proprietà che permettono l'astrazione digitale e l'ingegneria su larga scala. Le caratteristiche quantistiche — sovrapposizione, entanglement e interferenza — aprono nuove possibilità, ma impongono un controllo molto più delicato.

1.7 Architetture ibride

Il modello più realistico non è quello di una sostituzione completa, ma di una cooperazione. Un sistema quantistico operativo richiede componenti classiche per:

- compilare e ottimizzare i circuiti;
- generare segnali di controllo;
- sincronizzare le operazioni;
- decodificare le sindromi di errore;
- raccogliere e analizzare i risultati delle misure;
- eseguire le parti dell'algoritmo che non beneficiano del paradigma quantistico.

Anche molti algoritmi sono naturalmente ibridi. Un processore classico aggiorna parametri o gestisce una procedura iterativa, mentre un processore quantistico prepara stati e stima quantità difficili da ottenere classicamente.

La distinzione fra "computer classico" e "computer quantistico" resta utile a livello teorico, ma le macchine concrete sono sistemi stratificati. Il processore quantistico può essere visto come un acceleratore specializzato, analogo per certi aspetti a una GPU, ma basato su principi fisici diversi e destinato a sottoproblemi selezionati.

Bilancio dei vantaggi

Il calcolo classico offre:

1. **robustezza**: gli stati logici sono separati e rigenerabili;
2. **copiabilità**: i dati possono essere duplicati, distribuiti e verificati;
3. **accessibilità**: gli stati intermedi possono essere letti e usati per il controllo;
4. **maturità**: hardware, software e protocolli sono industrialmente consolidati;
5. **efficienza generale**: la maggior parte delle applicazioni ordinarie è trattata con costi contenuti;
6. **memoria e comunicazione**: grandi quantità di informazione persistente possono essere archiviate e trasmesse;
7. **prevedibilità ingegneristica**: prestazioni, errori e consumo possono essere caratterizzati con modelli ben sviluppati.

Il calcolo quantistico offre, per problemi appropriati:

1. **rappresentazione naturale di stati quantistici**: utile per simulare sistemi a molti corpi;
2. **interferenza controllata**: permette di amplificare proprietà globali e cancellare contributi indesiderati;
3. **accesso a strutture algebriche**: periodicità e fasi possono essere estratte con trasformazioni quantistiche;
4. **accelerazioni dimostrate in modelli ideali**: esponenziali o superpolinomiali per alcuni problemi strutturati, quadratiche per la ricerca non strutturata;
5. **nuovi protocolli informativi**: teletrasporto, codifica superdensa e crittografia quantistica non hanno equivalenti classici diretti;
6. **nuova prospettiva sulla complessità**: il modello costringe a distinguere la calcolabilità dall'efficienza fisica della simulazione.

1.8 Ricerca non strutturata

Esempio 1.8.1 (Esempio motivazionale)**.** Per rendere concreto il significato di un vantaggio quantistico, consideriamo il problema della ricerca non strutturata. Supponiamo di avere N possibili posizioni e una sola soluzione, riconoscibile mediante una funzione

$$f(x) = \begin{cases} 1, & \text{se } x \text{ è la soluzione,} \\ 0, & \text{altrimenti.} \end{cases}$$

Un algoritmo classico che non dispone di ulteriore struttura deve, nel caso peggiore, controllare un numero di elementi proporzionale a N.
L'algoritmo di Grover riduce il numero di *query* all'*oracle* a

$$O(\sqrt{N}).$$

L'idea non consiste nel "leggere tutte le risposte contemporaneamente". Il registro viene preparato in una sovrapposizione uniforme, l'*oracle* modifica la fase dello stato marcato e una successiva trasformazione di interferenza redistribuisce le ampiezze in modo da aumentare progressivamente la probabilità della soluzione. Solo alla fine viene effettuata la misura.
Per $N = 4$, la sovrapposizione uniforme è

$$|s\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle).$$

Se lo stato marcato è $|10\rangle$, l'*oracle* ne cambia il segno:

$$\frac{1}{2}(|00\rangle + |01\rangle - |10\rangle + |11\rangle).$$

Nel caso particolare $N = 4$, una singola iterazione completa di Grover porta idealmente tutta l'ampiezza sullo stato marcato, che viene quindi ottenuto con probabilità uno. Per N generale servono circa

$$k \simeq \frac{\pi}{4}\sqrt{N}$$

iterazioni.
Questo esempio anticipa tre idee che saranno sviluppate nei capitoli successivi: le ampiezze possono essere modificate senza essere direttamente lette; le fasi relative controllano l'interferenza; la misura finale estrae soltanto l'informazione che l'algoritmo è riuscito a concentrare nelle probabilità degli esiti. Il dettaglio circuitale dell'algoritmo di Grover sarà più naturale dopo avere introdotto formalmente porte, registri e misure quantistiche.

1.9 Conclusioni

La storia del calcolo moderno può essere letta come una progressiva astrazione. Turing separò il concetto di algoritmo dalla macchina concreta e mostrò la possibilità di una macchina universale. L'architettura a programma memorizzato, il transistor e la microelettronica trasformarono tale astrazione in una tecnologia generale, affidabile e scalabile.

Il calcolo quantistico nasce quando si applica al concetto di computazione una domanda fisica più radicale: quali trasformazioni dell'informazione sono possibili se il dispositivo segue direttamente

le leggi della meccanica quantistica? La risposta introduce qubit, sovrapposizione, entanglement, interferenza e misurazione. Queste risorse modificano il costo di alcuni problemi, ma non rendono irrilevante il calcolo classico.

Il punto centrale può essere riassunto in tre affermazioni:

1. il computer quantistico non calcola funzioni non calcolabili da una macchina classica, ma può ridurre drasticamente le risorse necessarie per alcune di esse;
2. il vantaggio quantistico è selettivo e deve essere dimostrato problema per problema, includendo preparazione, errori e misura;
3. il futuro del calcolo è verosimilmente ibrido: l'infrastruttura classica rimane essenziale, mentre processori quantistici possono fungere da acceleratori per compiti nei quali la struttura del problema coincide con le risorse della meccanica quantistica.

La contrapposizione non è quindi fra "vecchio" e "nuovo", ma fra differenti modi fisicamente realizzabili di elaborare l'informazione. Il calcolo classico eccelle quando sono decisive robustezza, memoria, controllo e generalità. Il calcolo quantistico diventa promettente quando la struttura del problema può essere riformulata in modo che la risposta si possa ottenere con una riduzione verificabile delle risorse.

Capitolo 2

Richiami matematici: calcolo matriciale e prodotto tensoriale

Questa sezione riassume gli strumenti matematici fondamentali del calcolo indispensabili per comprendere il calcolo quantistico. Il calcolo matriciale si fonda su un insieme ristretto di operazioni che devono essere manipolate con attenzione, soprattutto perché il prodotto matriciale non è commutativo. Trasposta, aggiunta, inversa, determinante, traccia, rango e diagonalizzazione costituiscono gli strumenti essenziali per analizzare matrici reali e complesse.

Il prodotto tensoriale estende il calcolo ordinario costruendo matrici a blocchi. La formula

$$A \otimes B = (a_{ij}B)$$

è la definizione operativa fondamentale. Da essa discendono le proprietà di bilinearità, associatività e compatibilità con prodotto ordinario, aggiunta, inversa, traccia, rango e struttura spettrale.

2.1 Calcolo matriciale e tensoriale

Una matrice A di dimensione $m \times n$ a coefficienti complessi è un insieme ordinato di numeri

$$A = (a_{ij}) = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \qquad a_{ij} \in \mathbb{C}.$$

Scriveremo $A \in M_{m\times n}(\mathbb{C})$. Se $m = n$, la matrice è quadrata.

Una matrice colonna è una matrice $n \times 1$:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix},$$

mentre una matrice riga ha dimensione $1 \times n$.

La matrice identità di ordine n è

$$I_n = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}.$$

Essa soddisfa

$$I_n A = A, \qquad A I_n = A$$

quando le dimensioni sono compatibili.

Somma e moltiplicazione per scalare

Per matrici della stessa dimensione,

$$A + B = (a_{ij} + b_{ij}).$$

Esempio 2.1.1. Siano

$$A = \begin{pmatrix} 1 & 2 \\ -1 & 3 \end{pmatrix}, \qquad B = \begin{pmatrix} 4 & -2 \\ 5 & 1 \end{pmatrix}.$$

Allora

$$A + B = \begin{pmatrix} 1+4 & 2-2 \\ -1+5 & 3+1 \end{pmatrix} = \begin{pmatrix} 5 & 0 \\ 4 & 4 \end{pmatrix}.$$

Per $\lambda \in \mathbb{C}$,

$$\lambda A = (\lambda a_{ij}).$$

Le proprietà essenziali sono

$$A + B = B + A, \qquad (A + B) + C = A + (B + C),$$

$$\lambda(A + B) = \lambda A + \lambda B, \qquad (\lambda + \mu)A = \lambda A + \mu A.$$

Prodotto matriciale

Siano

$$A \in M_{m \times n}(\mathbb{C}), \qquad B \in M_{n \times p}(\mathbb{C}).$$

Il prodotto $C = AB$ è la matrice $m \times p$ definita da

$$\boxed{c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}.}$$

Questa è la regola “riga per colonna”.

Esempio 2.1.2. Consideriamo

$$A = \begin{pmatrix} 1 & 2 & -1 \\ 0 & 3 & 4 \end{pmatrix}, \qquad B = \begin{pmatrix} 2 & 1 \\ -1 & 0 \\ 3 & 2 \end{pmatrix}.$$

Le dimensioni sono

$$A : 2 \times 3, \qquad B : 3 \times 2,$$

quindi

$$AB : 2 \times 2.$$

Gli elementi sono

$$(AB)_{11} = 1 \cdot 2 + 2(-1) + (-1)3 = -3,$$
$$(AB)_{12} = 1 \cdot 1 + 2 \cdot 0 + (-1)2 = -1,$$
$$(AB)_{21} = 0 \cdot 2 + 3(-1) + 4 \cdot 3 = 9,$$
$$(AB)_{22} = 0 \cdot 1 + 3 \cdot 0 + 4 \cdot 2 = 8.$$

Pertanto

$$\boxed{AB = \begin{pmatrix} -3 & -1 \\ 9 & 8 \end{pmatrix}.}$$

In generale il prodotto non è commutativo:

$$AB \neq BA.$$

Per esempio,

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix},$$

danno

$$AB = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}, \qquad BA = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}.$$

Valgono invece associatività e distributività:

$$(AB)C = A(BC),$$
$$A(B + C) = AB + AC, \qquad (A + B)C = AC + BC.$$

Potenze

Per una matrice quadrata A:

$$A^0 = I, \qquad A^1 = A, \qquad A^n = \underbrace{AA \cdots A}_{n \text{ fattori}}.$$

Esempio 2.1.3. Consideriamo

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

Calcoliamo dapprima $A^2 = A\,A$:

$$A^2 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

Applicando la regola riga-per-colonna,

$$A^2 = \begin{pmatrix} 1\cdot 1 + 1\cdot 0 & 1\cdot 1 + 1\cdot 1 \\ 0\cdot 1 + 1\cdot 0 & 0\cdot 1 + 1\cdot 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}.$$

Ora calcoliamo

$$A^3 = A^2 A.$$

Pertanto

$$A^3 = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix},$$

e quindi

$$A^3 = \begin{pmatrix} 1\cdot 1 + 2\cdot 0 & 1\cdot 1 + 2\cdot 1 \\ 0\cdot 1 + 1\cdot 0 & 0\cdot 1 + 1\cdot 1 \end{pmatrix} = \begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix}.$$

Questi primi calcoli suggeriscono la forma generale

$$A^n = \begin{pmatrix} 1 & n \\ 0 & 1 \end{pmatrix}.$$

La dimostriamo per induzione. Per $n = 1$ la formula è immediatamente vera:

$$A^1 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

Supponiamo ora che, per un certo $n \geq 1$,

$$A^n = \begin{pmatrix} 1 & n \\ 0 & 1 \end{pmatrix}.$$

Allora

$$A^{n+1} = A^n A,$$

per cui

$$A^{n+1} = \begin{pmatrix} 1 & n \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

Eseguendo esplicitamente il prodotto,

$$A^{n+1} = \begin{pmatrix} 1\cdot 1 + n\cdot 0 & 1\cdot 1 + n\cdot 1 \\ 0\cdot 1 + 1\cdot 0 & 0\cdot 1 + 1\cdot 1 \end{pmatrix} = \begin{pmatrix} 1 & n+1 \\ 0 & 1 \end{pmatrix}.$$

La formula è dunque valida per ogni intero $n \geq 1$:

$$\boxed{A^n = \begin{pmatrix} 1 & n \\ 0 & 1 \end{pmatrix}.}$$

Trasposta, coniugata e aggiunta

La trasposta di $A = (a_{ij})$ è

$$A^T = (a_{ji}).$$

Per esempio,

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \quad \Longrightarrow \quad A^T = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}.$$

La coniugata complessa è

$$A^* = (a_{ij}^*).$$

L'aggiunta è

$$\boxed{A^\dagger = (A^*)^T.}$$

Se

$$A = \begin{pmatrix} 1+\mathrm{i} & 2 \\ -\mathrm{i} & 3-2\mathrm{i} \end{pmatrix},$$

allora

$$A^* = \begin{pmatrix} 1-\mathrm{i} & 2 \\ \mathrm{i} & 3+2\mathrm{i} \end{pmatrix},$$

e quindi

$$A^\dagger = \begin{pmatrix} 1-\mathrm{i} & \mathrm{i} \\ 2 & 3+2\mathrm{i} \end{pmatrix}.$$

Le proprietà principali sono

$$(A^T)^T = A,$$
$$(AB)^T = B^T A^T,$$
$$(A^\dagger)^\dagger = A,$$
$$(AB)^\dagger = B^\dagger A^\dagger.$$

Matrice inversa

Una matrice quadrata A è invertibile se esiste A^{-1} tale che

$$AA^{-1} = A^{-1}A = I.$$

Per

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix},$$

se

$$ad - bc \neq 0,$$

si ha

$$\boxed{A^{-1} = \frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.}$$

Esempio 2.1.4. Per

$$A = \begin{pmatrix} 2 & 1 \\ 3 & 2 \end{pmatrix}$$

si ha

$$\det A = 4 - 3 = 1,$$

quindi

$$A^{-1} = \begin{pmatrix} 2 & -1 \\ -3 & 2 \end{pmatrix}.$$

La verifica è

$$AA^{-1} = \begin{pmatrix} 2 & 1 \\ 3 & 2 \end{pmatrix}\begin{pmatrix} 2 & -1 \\ -3 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Per matrici invertibili,

$$\boxed{(AB)^{-1} = B^{-1}A^{-1}.}$$

Infatti

$$(AB)(B^{-1}A^{-1}) = A(BB^{-1})A^{-1} = AA^{-1} = I.$$

Determinante

Per una matrice 2×2,

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix},$$

si definisce

$$\boxed{\det A = ad - bc.}$$

Per una matrice 3×3,

$$A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix},$$

lo sviluppo lungo la prima riga dà

$$\boxed{\det A = a(ei - fh) - b(di - fg) + c(dh - eg).}$$

Esempio 2.1.5. Consideriamo

$$A = \begin{pmatrix} 1 & 2 & 0 \\ -1 & 3 & 1 \\ 2 & 0 & 4 \end{pmatrix}.$$

Si ha

$$\begin{aligned} \det A &= 1 \begin{vmatrix} 3 & 1 \\ 0 & 4 \end{vmatrix} - 2 \begin{vmatrix} -1 & 1 \\ 2 & 4 \end{vmatrix} \\ &= 1(12) - 2(-4-2) \\ &= 12 + 12 \\ &= 24. \end{aligned}$$

Le proprietà fondamentali sono

$$\det(AB) = \det A \, \det B,$$

$$\det(A^T) = \det A,$$

$$\det(A^{-1}) = \frac{1}{\det A}.$$

Inoltre

$$A \text{ invertibile} \quad \Longleftrightarrow \quad \det A \neq 0.$$

Traccia e rango

Per una matrice quadrata $A = (a_{ij})$,

$$\boxed{\mathrm{Tr}(A) = \sum_{i=1}^{n} a_{ii}.}$$

Per

$$A = \begin{pmatrix} 2 & 1 & 0 \\ 3 & -1 & 4 \\ 0 & 2 & 5 \end{pmatrix}$$

si ha

$$\mathrm{Tr}(A) = 2 - 1 + 5 = 6.$$

La traccia è lineare e soddisfa

$$\boxed{\mathrm{Tr}(AB) = \mathrm{Tr}(BA).}$$

Dimostrazione:

$$\begin{aligned}\mathrm{Tr}(AB) &= \sum_i (AB)_{ii} \\ &= \sum_i \sum_j a_{ij} b_{ji} \\ &= \sum_j \sum_i b_{ji} a_{ij} \\ &= \sum_j (BA)_{jj} \\ &= \mathrm{Tr}(BA).\end{aligned}$$

Il rango di una matrice è il massimo numero di righe, o equivalentemente di colonne, linearmente indipendenti.

Per una matrice quadrata $n \times n$,

$$A \text{ invertibile} \iff \mathrm{rank}(A) = n \iff \det A \neq 0.$$

Autovalori e diagonalizzazione

Un numero $\lambda \in \mathbb{C}$ è autovalore di una matrice quadrata A se esiste una matrice colonna non nulla x tale che

$$Ax = \lambda x.$$

Equivalentemente,

$$(A - \lambda I)x = 0.$$

Perché esista una soluzione non nulla occorre

$$\boxed{\det(A - \lambda I) = 0.}$$

Esempio 2.1.6. Sia

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}.$$

Allora

$$A - \lambda I = \begin{pmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{pmatrix},$$

e

$$\begin{aligned}\det(A - \lambda I) &= (2-\lambda)^2 - 1 \\ &= \lambda^2 - 4\lambda + 3 \\ &= (\lambda - 1)(\lambda - 3).\end{aligned}$$

Gli autovalori sono quindi

$$\lambda_1 = 1, \qquad \lambda_2 = 3.$$

Per $\lambda_1 = 1$:

$$(A - I)x = 0$$

diventa

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

cioè

$$x_1 + x_2 = 0.$$

Una scelta è

$$x^{(1)} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}.$$

Per $\lambda_2 = 3$:

$$(A - 3I)x = 0$$

porta a

$$-x_1 + x_2 = 0,$$

e quindi si può scegliere

$$x^{(2)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Una matrice A è diagonalizzabile se esiste una matrice invertibile S tale che

$$\boxed{S^{-1}AS = D,}$$

con D diagonale. Equivalentemente,

$$A = SDS^{-1}.$$

Se

$$A = SDS^{-1},$$

allora

$$A^n = SD^nS^{-1}.$$

Se

$$D = \mathrm{diag}(\lambda_1, \ldots, \lambda_n),$$

si ha

$$D^n = \mathrm{diag}(\lambda_1^n, \ldots, \lambda_n^n).$$

Matrici hermitiane e unitarie

Una matrice H è hermitiana se

$$\boxed{H^\dagger = H.}$$

Elemento per elemento,

$$H_{ij} = H_{ji}^*.$$

Una matrice hermitiana 3×3 ha quindi forma

$$H = \begin{pmatrix} a & b + \mathrm{i}c & d + \mathrm{i}e \\ b - \mathrm{i}c & f & g + \mathrm{i}h \\ d - \mathrm{i}e & g - \mathrm{i}h & \ell \end{pmatrix},$$

con tutti i parametri

$$a, b, c, d, e, f, g, h, \ell \in \mathbb{R}.$$

Le matrici hermitiane hanno autovalori reali e sono diagonalizzabili mediante matrici unitarie.

Una matrice U è unitaria se

$$\boxed{U^\dagger U = U U^\dagger = I.}$$

Pertanto

$$U^{-1} = U^\dagger.$$

Esempio 2.1.7. Sia

$$U = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Poiché U è reale e simmetrica,

$$U^\dagger = U.$$

Calcolando:

$$\begin{aligned} U^\dagger U &= \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\ &= \frac{1}{2} \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \\ &= I. \end{aligned}$$

Quindi U è unitaria.

2.2 Prodotto tensoriale di matrici

Nel calcolo matriciale il prodotto tensoriale tra matrici è rappresentato dal *prodotto di Kronecker*.

Siano

$$A = (a_{ij}) \in M_{m \times n}(\mathbb{C}), \qquad B \in M_{p \times q}(\mathbb{C}).$$

Si definisce

$$\boxed{A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{pmatrix}.}$$

La dimensione è

$$\boxed{(mp) \times (nq).}$$

Esempio 2.2.1. Siano

$$A = \begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} 0 & 1 \\ -1 & 2 \end{pmatrix}.$$

Per definizione,

$$A \otimes B = \begin{pmatrix} 1B & 3B \\ 0B & 1B \end{pmatrix}.$$

Calcoliamo i blocchi:

$$1B = \begin{pmatrix} 0 & 1 \\ -1 & 2 \end{pmatrix},$$

$$3B = \begin{pmatrix} 0 & 3 \\ -3 & 6 \end{pmatrix},$$

$$0B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

Quindi

$$\boxed{A \otimes B = \begin{pmatrix} 0 & 1 & 0 & 3 \\ -1 & 2 & -3 & 6 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 2 \end{pmatrix}.}$$

Esempio 2.2.2. Siano

$$A = \begin{pmatrix} 1 & 2 & -1 \end{pmatrix}, \qquad B = \begin{pmatrix} 1 & 0 \\ 3 & 4 \end{pmatrix}.$$

Poiché

$$A : 1 \times 3, \qquad B : 2 \times 2,$$

si ha

$$A \otimes B : 2 \times 6.$$

Calcoliamo

$$A \otimes B = \begin{pmatrix} 1B & 2B & -B \end{pmatrix}$$

e quindi

$$\boxed{A \otimes B = \begin{pmatrix} 1 & 0 & 2 & 0 & -1 & 0 \\ 3 & 4 & 6 & 8 & -3 & -4 \end{pmatrix}.}$$

Proprietà del prodotto tensoriale

Il prodotto tensoriale è bilineare:

$$(A + B) \otimes C = A \otimes C + B \otimes C,$$

$$A \otimes (B + C) = A \otimes B + A \otimes C.$$

Per uno scalare λ:

$$(\lambda A) \otimes B = \lambda(A \otimes B) = A \otimes (\lambda B).$$

È associativo, a meno della naturale identificazione delle dimensioni:

$$\boxed{(A \otimes B) \otimes C = A \otimes (B \otimes C).}$$

In generale non è commutativo:

$$A \otimes B \neq B \otimes A.$$

Esempio 2.2.3. Per

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, \qquad B = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

si ottiene

$$A \otimes B = \begin{pmatrix} 0 & 1 & 0 & 2 \\ 1 & 0 & 2 & 0 \\ 0 & 3 & 0 & 4 \\ 3 & 0 & 4 & 0 \end{pmatrix},$$

mentre

$$B \otimes A = \begin{pmatrix} 0 & 0 & 1 & 2 \\ 0 & 0 & 3 & 4 \\ 1 & 2 & 0 & 0 \\ 3 & 4 & 0 & 0 \end{pmatrix}.$$

Prodotto misto

Una proprietà fondamentale è

$$\boxed{(A \otimes B)(C \otimes D) = (AC) \otimes (BD),}$$

quando le dimensioni sono compatibili.

Esempio 2.2.4. Siano

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

$$C = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}, \quad D = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

Si ha

$$AC = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix},$$

e

$$BD = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}.$$

Quindi

$$(AC) \otimes (BD) = \begin{pmatrix} 0 & 2 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 2 & 2 \end{pmatrix}.$$

Per la proprietà del prodotto misto, questa è anche la matrice

$$(A \otimes B)(C \otimes D).$$

Trasposta, aggiunta e inversa del prodotto tensoriale

Valgono le formule

$$\boxed{(A \otimes B)^T = A^T \otimes B^T,}$$

$$\boxed{(A \otimes B)^* = A^* \otimes B^*,}$$

e quindi

$$\boxed{(A \otimes B)^\dagger = A^\dagger \otimes B^\dagger.}$$

Se A e B sono invertibili,

$$\boxed{(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}.}$$

Infatti

$$\begin{aligned}(A \otimes B)(A^{-1} \otimes B^{-1}) &= (AA^{-1}) \otimes (BB^{-1}) \\ &= I \otimes I \\ &= I.\end{aligned}$$

Traccia, determinante e rango del prodotto tensoriale

Per matrici quadrate

$$A \in M_m(\mathbb{C}), \qquad B \in M_n(\mathbb{C}),$$

si ha

$$\boxed{\mathrm{Tr}(A \otimes B) = \mathrm{Tr}(A)\,\mathrm{Tr}(B).}$$

Inoltre,

$$\boxed{\det(A \otimes B) = (\det A)^n (\det B)^m.}$$

Per matrici anche rettangolari:

$$\boxed{\mathrm{rank}(A \otimes B) = \mathrm{rank}(A)\,\mathrm{rank}(B).}$$

Esempio 2.2.5. Siano

$$A = \begin{pmatrix} 1 & 2 \\ 0 & 3 \end{pmatrix}, \qquad B = \begin{pmatrix} 4 & 0 \\ 1 & -1 \end{pmatrix}.$$

Si ha

$$\mathrm{Tr}(A) = 4, \qquad \mathrm{Tr}(B) = 3.$$

Quindi

$$\mathrm{Tr}(A \otimes B) = 12.$$

Autovalori del prodotto tensoriale

Supponiamo

$$Ax = \lambda x, \qquad By = \mu y.$$

Allora

$$\begin{aligned}(A \otimes B)(x \otimes y) &= (Ax) \otimes (By) \\ &= (\lambda x) \otimes (\mu y) \\ &= \lambda\mu (x \otimes y).\end{aligned}$$

Quindi

$$\boxed{\lambda\mu}$$

è un autovalore di $A \otimes B$.

Se gli autovalori di A sono

$$\lambda_1, \ldots, \lambda_m$$

e quelli di B sono

$$\mu_1, \ldots, \mu_n,$$

gli autovalori di $A \otimes B$ sono

$$\boxed{\lambda_i \mu_j.}$$

Esempio 2.2.6. Per

$$A = \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix}, \qquad B = \begin{pmatrix} -1 & 0 \\ 0 & 4 \end{pmatrix},$$

gli autovalori sono rispettivamente

$$2, 3$$

e

$$-1, 4.$$

I prodotti sono

$$-2,\ 8,\ -3,\ 12.$$

Infatti

$$A \otimes B = \begin{pmatrix} -2 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 \\ 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & 12 \end{pmatrix}.$$

Prodotti tensoriali iterati

Se

$$A : m_1 \times n_1, \quad B : m_2 \times n_2, \quad C : m_3 \times n_3,$$

allora

$$A \otimes B \otimes C$$

ha dimensione

$$\boxed{(m_1 m_2 m_3) \times (n_1 n_2 n_3).}$$

Più in generale, se

$$A_k \in M_{d_k}(\mathbb{C}),$$

allora

$$A_1 \otimes A_2 \otimes \cdots \otimes A_n$$

ha dimensione

$$\boxed{\left(\prod_{k=1}^{n} d_k\right) \times \left(\prod_{k=1}^{n} d_k\right).}$$

Nel caso particolare di n matrici 2×2:

$$\boxed{A_1 \otimes \cdots \otimes A_n \in M_{2^n}(\mathbb{C}).}$$

Commutatore e anticommutatore

Per due matrici quadrate A e B della stessa dimensione:

$$\boxed{[A, B] = AB - BA}$$

è il commutatore, mentre

$$\boxed{\{A, B\} = AB + BA}$$

è l'anticommutatore.

Vale

$$[A, B] = 0 \quad \Longleftrightarrow \quad AB = BA.$$

Esempio 2.2.7. Per

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix},$$

abbiamo

$$AB = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix},$$

$$BA = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}.$$

Pertanto

$$[A, B] = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Esponenziale di matrice

Per una matrice quadrata A:

$$\boxed{e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!} = I + A + \frac{A^2}{2!} + \cdots .}$$

Se

$$A = SDS^{-1},$$

allora

$$A^k = SD^kS^{-1},$$

e quindi

$$\begin{aligned} e^A &= \sum_{k=0}^{\infty} \frac{SD^kS^{-1}}{k!} \\ &= S\left(\sum_{k=0}^{\infty} \frac{D^k}{k!}\right) S^{-1} \\ &= \boxed{Se^DS^{-1}.} \end{aligned}$$

Se

$$D = \operatorname{diag}(\lambda_1, \ldots, \lambda_n),$$

allora

$$e^D = \operatorname{diag}(e^{\lambda_1}, \ldots, e^{\lambda_n}).$$

Quando

$$AB = BA,$$

vale

$$\boxed{e^{A+B} = e^Ae^B.}$$

Se A e B non commutano, questa relazione non è in generale valida.

2.3 Tabella riassuntiva

Operazione	**Formula**
Prodotto ordinario	$(AB)_{ij} = \sum_k a_{ik} b_{kj}$
Trasposta	$(AB)^T = B^T A^T$
Aggiunta	$(AB)^\dagger = B^\dagger A^\dagger$
Inversa	$(AB)^{-1} = B^{-1} A^{-1}$
Determinante	$\det(AB) = \det A \det B$
Traccia	$\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$
Prodotto tensoriale	$A \otimes B = (a_{ij} B)$
Dimensione tensoriale	$(m \times n) \otimes (p \times q) = (mp) \times (nq)$
Prodotto misto	$(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$
Aggiunta tensoriale	$(A \otimes B)^\dagger = A^\dagger \otimes B^\dagger$
Inversa tensoriale	$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$
Traccia tensoriale	$\mathrm{Tr}(A \otimes B) = \mathrm{Tr}(A)\,\mathrm{Tr}(B)$
Rango tensoriale	$\mathrm{rank}(A \otimes B) = \mathrm{rank}(A)\,\mathrm{rank}(B)$
Determinante tensoriale	$\det(A \otimes B) = (\det A)^n (\det B)^m$
Autovalori tensoriali	$\mathrm{spec}(A \otimes B) = \{\lambda_r(A)\lambda_s(B)\}_{r,s}$

2.4 Errori frequenti

1. Il prodotto AB non è il prodotto elemento per elemento.
2. In generale $AB \neq BA$.
3. L'ordine si inverte in
$$(AB)^\dagger = B^\dagger A^\dagger$$
e
$$(AB)^{-1} = B^{-1} A^{-1}.$$
4. A^T, A^* e $A^\dagger$ sono operazioni diverse.
5. Il prodotto tensoriale non coincide con il prodotto ordinario.
6. In generale
$$A \otimes B \neq B \otimes A.$$
7. Se
$$A : m \times n, \qquad B : p \times q,$$
allora
$$A \otimes B : (mp) \times (nq).$$

8. La formula

$$e^{A+B} = e^A e^B$$

richiede, in generale, la condizione $AB = BA$.

Capitolo 3

La notazione di Dirac

Questo capitolo introduce la notazione di Dirac come linguaggio compatto dell'algebra lineare complessa e ne esplicita in modo sistematico il rapporto con la rappresentazione matriciale. L'obiettivo è mostrare che ket, bra, prodotti interni, prodotti esterni e operatori non costituiscono una matematica diversa dal calcolo matriciale: una volta fissata una base, corrispondono rispettivamente a matrici colonna, matrici riga coniugate, prodotti riga-per-colonna, prodotti colonna-per-riga e matrici quadrate. L'esposizione insiste sui passaggi algebrici espliciti e sugli errori di notazione più frequenti.

3.1 Perché introdurre una nuova notazione?

La notazione di Dirac è un modo compatto di scrivere oggetti che, in una base fissata, possono essere rappresentati mediante vettori colonna, vettori riga e matrici. Il vantaggio principale è che essa separa l'oggetto astratto dalla particolare base utilizzata per rappresentarlo.

Un vettore può essere indicato con

$$|\psi\rangle,$$

senza specificare immediatamente le sue coordinate. Solo dopo aver scelto una base si può scrivere, per esempio,

$$|\psi\rangle \longleftrightarrow \begin{pmatrix} \alpha \\ \beta \end{pmatrix}.$$

Il punto fondamentale è quindi:

la notazione di Dirac è una notazione per l'algebra lineare;

la rappresentazione matriciale compare quando si sceglie una base.

3.2 Il ket

Definizione

Un simbolo della forma

$$|\psi\rangle$$

si chiama *ket*.

Esso rappresenta un vettore di uno spazio vettoriale complesso.

Se lo spazio ha dimensione n e scegliamo una base ordinata

$$\{|e_1\rangle, |e_2\rangle, \ldots, |e_n\rangle\},$$

ogni vettore può essere scritto come

$$|\psi\rangle = \sum_{j=1}^{n} \psi_j |e_j\rangle.$$

I coefficienti

$$\psi_1, \ldots, \psi_n$$

sono le coordinate del vettore in quella base.

La rappresentazione matriciale è quindi

$$\boxed{|\psi\rangle \longleftrightarrow \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix}.}$$

Esempio 3.2.1. Scegliamo la base

$$|e_1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |e_2\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Se

$$|\psi\rangle = \alpha|e_1\rangle + \beta|e_2\rangle,$$

allora

$$\begin{aligned} |\psi\rangle &= \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} \alpha \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \beta \end{pmatrix} \\ &= \boxed{\begin{pmatrix} \alpha \\ \beta \end{pmatrix}.} \end{aligned}$$

3.3 Il bra

Dal ket al bra

A ogni ket

$$|\psi\rangle$$

si associa un bra

$$\langle\psi|.$$

Se

$$|\psi\rangle \longleftrightarrow \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix},$$

allora

$$\boxed{\langle\psi| \longleftrightarrow \begin{pmatrix} \psi_1^* & \psi_2^* & \cdots & \psi_n^* \end{pmatrix}.}$$

Quindi il bra non è semplicemente la trasposta del ket: è la *trasposta coniugata*.

In notazione matriciale:

$$\boxed{\langle\psi| = |\psi\rangle^\dagger.}$$

Esempio 3.3.1. Sia

$$|\psi\rangle = \begin{pmatrix} 1+\mathrm{i} \\ 2-\mathrm{i} \end{pmatrix}.$$

La coniugata complessa è

$$\begin{pmatrix} 1-\mathrm{i} \\ 2+\mathrm{i} \end{pmatrix}.$$

Trasponendo:

$$\boxed{\langle\psi| = \begin{pmatrix} 1-\mathrm{i} & 2+\mathrm{i} \end{pmatrix}.}$$

3.4 Il prodotto interno: il bra-ket

Definizione

Dati due vettori

$$|\phi\rangle, \qquad |\psi\rangle,$$

il prodotto interno è scritto

$$\boxed{\langle\phi \mid \psi\rangle.}$$

Nella rappresentazione matriciale:

$$\boxed{\langle\phi \mid \psi\rangle = \langle\phi||\psi\rangle.}$$

Si tratta quindi del prodotto di una matrice riga $1 \times n$ per una matrice colonna $n \times 1$. Il risultato è uno scalare.

Se

$$|\phi\rangle = \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_n \end{pmatrix}, \qquad |\psi\rangle = \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_n \end{pmatrix},$$

allora

$$\langle\phi| = \begin{pmatrix}\phi_1^* & \cdots & \phi_n^*\end{pmatrix},$$

e quindi

$$\boxed{\langle\phi \mid \psi\rangle = \sum_{j=1}^{n} \phi_j^* \psi_j.}$$

Esempio 3.4.1. Siano

$$|\phi\rangle = \begin{pmatrix}1+\mathrm{i}\\ 2\end{pmatrix}, \qquad |\psi\rangle = \begin{pmatrix}2\\ 1-\mathrm{i}\end{pmatrix}.$$

Prima costruiamo il bra:

$$\langle\phi| = \begin{pmatrix}1-\mathrm{i} & 2\end{pmatrix}.$$

Quindi

$$\begin{aligned}\langle\phi \mid \psi\rangle &= \begin{pmatrix}1-\mathrm{i} & 2\end{pmatrix}\begin{pmatrix}2\\ 1-\mathrm{i}\end{pmatrix}\\ &= (1-\mathrm{i})\,2 + 2(1-\mathrm{i})\\ &= 2 - 2\mathrm{i} + 2 - 2\mathrm{i}\\ &= \boxed{4-4\mathrm{i}.}\end{aligned}$$

Ordine dei fattori

In generale

$$\langle\phi \mid \psi\rangle \neq \langle\psi \mid \phi\rangle.$$

Vale invece

$$\boxed{\langle\psi \mid \phi\rangle = \langle\phi \mid \psi\rangle^*.}$$

Infatti, se

$$\langle\phi \mid \psi\rangle = \sum_j \phi_j^* \psi_j,$$

allora

$$\langle\phi \mid \psi\rangle^* = \sum_j \phi_j \psi_j^* = \sum_j \psi_j^* \phi_j = \langle\psi \mid \phi\rangle.$$

3.5 Norma e ortogonalità

La norma di un vettore è

$$\boxed{\|\psi\| = \sqrt{\langle\psi \mid \psi\rangle}.}$$

Se

$$|\psi\rangle = \begin{pmatrix}\psi_1\\ \vdots\\ \psi_n\end{pmatrix},$$

allora

$$\langle \psi \mid \psi \rangle = \sum_j |\psi_j|^2,$$

per cui

$$\boxed{\|\psi\| = \sqrt{|\psi_1|^2 + \cdots + |\psi_n|^2}.}$$

Due vettori sono ortogonali se

$$\boxed{\langle \phi \mid \psi \rangle = 0.}$$

Esempio 3.5.1. Consideriamo

$$|u\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \qquad |v\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}.$$

Si ha

$$\langle u| = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \end{pmatrix}.$$

Allora

$$\begin{aligned} \langle u \mid v \rangle &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \\ &= \frac{1}{2}(1 - 1) \\ &= 0. \end{aligned}$$

Quindi i due vettori sono ortogonali.
La normalizzazione di u si verifica con

$$\begin{aligned} \langle u \mid u \rangle &= \frac{1}{2} \begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \frac{1}{2}(1 + 1) \\ &= 1. \end{aligned}$$

3.6 Basi ortonormali

Una base

$$\{|e_1\rangle, \ldots, |e_n\rangle\}$$

è ortonormale se

$$\boxed{\langle e_i \mid e_j \rangle = \delta_{ij},}$$

dove

$$\delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

Se la base è ortonormale e

$$|\psi\rangle = \sum_j \psi_j |e_j\rangle,$$

allora

$$\boxed{\psi_j = \langle e_j \mid \psi \rangle.}$$

Dimostriamolo:

$$\begin{aligned}\langle e_j \mid \psi \rangle &= \langle e_j | \left(\sum_k \psi_k |e_k\rangle \right) \\ &= \sum_k \psi_k \langle e_j \mid e_k \rangle \\ &= \sum_k \psi_k \delta_{jk} \\ &= \psi_j.\end{aligned}$$

Quindi il bra

$$\langle e_j |$$

estrae la j-esima coordinata del vettore rispetto alla base ortonormale.

3.7 La risoluzione dell'identità

Per una base ortonormale

$$\{|e_1\rangle, \ldots, |e_n\rangle\}$$

vale

$$\boxed{\mathbb{I} = \sum_{j=1}^{n} |e_j\rangle\langle e_j|.}$$

Per comprenderne il significato, applichiamo il secondo membro a un vettore generico:

$$\begin{aligned}\left(\sum_j |e_j\rangle\langle e_j| \right) |\psi\rangle &= \sum_j |e_j\rangle \langle e_j \mid \psi\rangle \\ &= \sum_j \psi_j |e_j\rangle \\ &= |\psi\rangle.\end{aligned}$$

Quindi l'operatore tra parentesi agisce esattamente come l'identità.

Versione matriciale

In dimensione due, con

$$|e_1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |e_2\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

si ha

$$|e_1\rangle\langle e_1| = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix},$$

e

$$|e_2\rangle\langle e_2| = \begin{pmatrix}0\\1\end{pmatrix}\begin{pmatrix}0 & 1\end{pmatrix} = \begin{pmatrix}0 & 0\\0 & 1\end{pmatrix}.$$

Sommandole:

$$\begin{pmatrix}1 & 0\\0 & 0\end{pmatrix} + \begin{pmatrix}0 & 0\\0 & 1\end{pmatrix} = \boxed{\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}}.$$

3.8 Il prodotto esterno

Definizione

Dati due vettori

$$|u\rangle, \qquad |v\rangle,$$

il prodotto esterno è

$$\boxed{|u\rangle\langle v|.}$$

A differenza del prodotto interno

$$\langle v \mid u\rangle,$$

che produce uno scalare, il prodotto esterno produce un operatore.

In rappresentazione matriciale è il prodotto di una matrice colonna per una matrice riga:

$$(n \times 1)(1 \times n) \longrightarrow n \times n.$$

Esempio 3.8.1. Siano

$$|u\rangle = \begin{pmatrix}1\\\mathrm{i}\end{pmatrix}, \qquad |v\rangle = \begin{pmatrix}2\\1-\mathrm{i}\end{pmatrix}.$$

Prima costruiamo

$$\langle v| = \begin{pmatrix}2 & 1+\mathrm{i}\end{pmatrix}.$$

Quindi

$$\begin{aligned}|u\rangle\langle v| &= \begin{pmatrix}1\\\mathrm{i}\end{pmatrix}\begin{pmatrix}2 & 1+\mathrm{i}\end{pmatrix}\\ &= \begin{pmatrix}1\cdot 2 & 1(1+\mathrm{i})\\\mathrm{i}\cdot 2 & \mathrm{i}(1+\mathrm{i})\end{pmatrix}.\end{aligned}$$

Calcoliamo l'ultimo elemento:

$$\mathrm{i}(1+\mathrm{i}) = \mathrm{i} + \mathrm{i}^2 = \mathrm{i} - 1.$$

Pertanto

$$\boxed{|u\rangle\langle v| = \begin{pmatrix}2 & 1+\mathrm{i}\\2\mathrm{i} & -1+\mathrm{i}\end{pmatrix}.}$$

Azione del prodotto esterno

Sia

$$|w\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Usando la matrice appena ottenuta:

$$\begin{aligned}(|u\rangle\langle v|)|w\rangle &= \begin{pmatrix} 2 & 1+\mathrm{i} \\ 2\mathrm{i} & -1+\mathrm{i} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 2 \\ 2\mathrm{i} \end{pmatrix}.\end{aligned}$$

Ora verifichiamo la stessa operazione usando direttamente la notazione di Dirac:

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

Prima:

$$\langle v||w\rangle = \begin{pmatrix} 2 & 1+\mathrm{i} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 2.$$

Quindi

$$|u\rangle\langle v \mid w\rangle = 2 \begin{pmatrix} 1 \\ \mathrm{i} \end{pmatrix} = \boxed{\begin{pmatrix} 2 \\ 2\mathrm{i} \end{pmatrix}.}$$

I due calcoli coincidono.

3.9 Operatori lineari

Un operatore lineare A agisce su un ket come

$$A|\psi\rangle.$$

Una volta fissata una base, A è rappresentato da una matrice e il ket da una matrice colonna.

Per esempio,

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad |\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}.$$

Allora

$$A|\psi\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}.$$

Quindi l'espressione astratta

$$A|\psi\rangle$$

e il prodotto matriciale

$$A \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

sono due modi di rappresentare la stessa operazione, una volta scelta una base.

3.10 Elementi di matrice

Un'espressione fondamentale è

$$\boxed{\langle e_i|A|e_j\rangle.}$$

Essa produce uno scalare.

In una base ortonormale, questo scalare è precisamente l'elemento di matrice di riga i e colonna j:

$$\boxed{A_{ij} = \langle e_i|A|e_j\rangle.}$$

Dimostrazione in dimensione due

Sia

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix},$$

con

$$|e_1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |e_2\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Calcoliamo

$$\langle e_1|A|e_2\rangle.$$

Prima:

$$A|e_2\rangle = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} b \\ d \end{pmatrix}.$$

Poi:

$$\langle e_1| = \begin{pmatrix} 1 & 0 \end{pmatrix}.$$

Quindi

$$\langle e_1|A|e_2\rangle = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} b \\ d \end{pmatrix} = b.$$

Ma b è proprio l'elemento

$$A_{12}.$$

Quindi

$$\boxed{\langle e_1|A|e_2\rangle = A_{12}.}$$

3.11 Ricostruzione di una matrice con i prodotti esterni

Se

$$\{|e_i\rangle\}$$

è una base ortonormale, ogni operatore A può essere scritto come

$$\boxed{A = \sum_{i,j} A_{ij} |e_i\rangle\langle e_j|.}$$

Poiché

$$A_{ij} = \langle e_i | A | e_j \rangle,$$

si può anche scrivere

$$A = \sum_{i,j} \langle e_i | A | e_j \rangle |e_i\rangle\langle e_j|.$$

Esempio 3.11.1. Sia

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

I quattro prodotti esterni di base sono

$$|e_1\rangle\langle e_1| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix},$$

$$|e_1\rangle\langle e_2| = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix},$$

$$|e_2\rangle\langle e_1| = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix},$$

$$|e_2\rangle\langle e_2| = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

Quindi

$$\begin{aligned} A &= a|e_1\rangle\langle e_1| + b|e_1\rangle\langle e_2| + c|e_2\rangle\langle e_1| + d|e_2\rangle\langle e_2| \\ &= a \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + b \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \\ &\quad + c \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} + d \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \\ &= \boxed{\begin{pmatrix} a & b \\ c & d \end{pmatrix}.} \end{aligned}$$

3.12 L'aggiunto di un operatore

Se un operatore A è rappresentato da una matrice, il suo aggiunto $A^\dagger$ è rappresentato dalla trasposta coniugata della matrice.

In notazione di Dirac vale

$$\boxed{(A|\psi\rangle)^\dagger = \langle\psi|A^\dagger.}$$

Per esempio, se

$$A = \begin{pmatrix} 1 & \mathrm{i} \\ 2 & 3 \end{pmatrix},$$

allora

$$A^* = \begin{pmatrix} 1 & -\mathrm{i} \\ 2 & 3 \end{pmatrix},$$

e

$$\boxed{A^\dagger = \begin{pmatrix} 1 & 2 \\ -\mathrm{i} & 3 \end{pmatrix}.}$$

3.13 Proiettori

Se

$$|u\rangle$$

è normalizzato,

$$\langle u \mid u \rangle = 1,$$

allora

$$\boxed{P_u = |u\rangle\langle u|}$$

è un proiettore ortogonale sul sottospazio generato da u.

Verifichiamo l'idempotenza:

$$\begin{aligned} P_u^2 &= (|u\rangle\langle u|)(|u\rangle\langle u|) \\ &= |u\rangle\langle u \mid u\rangle\langle u| \\ &= |u\rangle \cdot 1 \cdot \langle u| \\ &= P_u. \end{aligned}$$

Quindi

$$\boxed{P_u^2 = P_u.}$$

Esempio 3.13.1. Sia

$$|u\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Allora

$$\langle u| = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \end{pmatrix},$$

e

$$\begin{aligned} P_u &= |u\rangle\langle u| \\ &= \frac{1}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \end{pmatrix} \\ &= \boxed{\frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}.} \end{aligned}$$

Verifichiamo:

$$\begin{aligned} P_u^2 &= \frac{1}{4}\begin{pmatrix}1 & 1\\ 1 & 1\end{pmatrix}\begin{pmatrix}1 & 1\\ 1 & 1\end{pmatrix} \\ &= \frac{1}{4}\begin{pmatrix}2 & 2\\ 2 & 2\end{pmatrix} \\ &= \frac{1}{2}\begin{pmatrix}1 & 1\\ 1 & 1\end{pmatrix} \\ &= P_u. \end{aligned}$$

3.14 Prodotto tensoriale in notazione di Dirac

La notazione di Dirac è particolarmente compatta quando si usano prodotti tensoriali.

Se

$$|u\rangle = \begin{pmatrix}u_1\\ u_2\end{pmatrix}, \qquad |v\rangle = \begin{pmatrix}v_1\\ v_2\end{pmatrix},$$

allora

$$|u\rangle \otimes |v\rangle$$

corrisponde al prodotto di Kronecker delle due matrici colonna:

$$\boxed{|u\rangle \otimes |v\rangle \longleftrightarrow \begin{pmatrix}u_1v_1\\ u_1v_2\\ u_2v_1\\ u_2v_2\end{pmatrix}.}$$

Spesso si abbrevia

$$|u\rangle \otimes |v\rangle$$

come

$$|uv\rangle,$$

quando non vi è ambiguità.

Esempio 3.14.1. Siano

$$|u\rangle = \begin{pmatrix}1\\ 2\end{pmatrix}, \qquad |v\rangle = \begin{pmatrix}3\\ 4\end{pmatrix}.$$

Allora

$$|u\rangle \otimes |v\rangle = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \otimes \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 1 \begin{pmatrix} 3 \\ 4 \end{pmatrix} \\ 2 \begin{pmatrix} 3 \\ 4 \end{pmatrix} \end{pmatrix} = \boxed{\begin{pmatrix} 3 \\ 4 \\ 6 \\ 8 \end{pmatrix}}.$$

3.15 Tabella di corrispondenza Dirac–matrici

Notazione di Dirac	**Rappresentazione matriciale**
$\|\psi\rangle$	matrice colonna $n \times 1$
$\langle\psi\|$	matrice riga $1 \times n$ ottenuta per trasposta coniugata
$\langle\phi \mid \psi\rangle$	prodotto riga-per-colonna, quindi uno scalare
$\|u\rangle\langle v\|$	prodotto colonna-per-riga, quindi una matrice $n \times n$
$A\|\psi\rangle$	prodotto matrice-per-colonna
$\langle\phi\|A$	prodotto riga-per-matrice
$\langle\phi\|A\|\psi\rangle$	scalare ottenuto da riga $\times$ matrice $\times$ colonna
$\langle e_i\|A\|e_j\rangle$	elemento di matrice A_{ij}
$\sum_i \|e_i\rangle\langle e_i\|$	matrice identità
$\sum_{ij} A_{ij}\|e_i\rangle\langle e_j\|$	ricostruzione completa della matrice A
$\|u\rangle \otimes \|v\rangle$	prodotto di Kronecker di due matrici colonna

3.16 Come leggere le espressioni di Dirac

Una regola pratica molto utile consiste nel leggere le espressioni da sinistra a destra controllando le dimensioni matriciali implicite.

Per esempio,

$$\langle\phi|A|\psi\rangle$$

corrisponde a

$$(1 \times n)(n \times n)(n \times 1).$$

Prima:

$$(1 \times n)(n \times n) = 1 \times n.$$

Poi:

$$(1 \times n)(n \times 1) = 1 \times 1.$$

Quindi il risultato è uno scalare.

Invece

$$|\psi\rangle\langle\phi|$$

corrisponde a

$$(n \times 1)(1 \times n) = n \times n,$$

quindi produce una matrice.

Questa semplice analisi dimensionale permette di evitare molti errori.

3.17 Errori frequenti

1. **Confondere ket e bra.**

   Un ket è rappresentato da una colonna:

   $$|\psi\rangle \leftrightarrow \begin{pmatrix} \psi_1 \\ \psi_2 \end{pmatrix}.$$

   Il bra corrispondente è

   $$\langle\psi| \leftrightarrow \begin{pmatrix} \psi_1^* & \psi_2^* \end{pmatrix}.$$

2. **Dimenticare la coniugazione complessa.**

   In generale

   $$\langle\psi| \neq |\psi\rangle^T.$$

   La relazione corretta è

   $$\langle\psi| = |\psi\rangle^\dagger.$$

3. **Confondere prodotto interno e prodotto esterno.**

   $$\langle\phi \mid \psi\rangle$$

   è uno scalare, mentre

   $$|\psi\rangle\langle\phi|$$

   è una matrice.

4. **Pensare che la notazione di Dirac elimini le matrici.**

   La notazione di Dirac è indipendente dalla base; le matrici compaiono appena si sceglie una base.

5. **Confondere un operatore astratto con una particolare matrice.**

   Lo stesso operatore può avere matrici diverse in basi diverse.

6. **Dimenticare l'ordine dei fattori.**

   In generale

$$AB \neq BA,$$

   e analogamente l'ordine delle espressioni di Dirac è essenziale.

3.18 Conclusioni

La notazione di Dirac è un linguaggio compatto per esprimere operazioni di algebra lineare in spazi complessi. Il collegamento con il formalismo matriciale è diretto:

$$|\psi\rangle \leftrightarrow \text{colonna},$$

$$\langle\psi| \leftrightarrow \text{riga trasposta coniugata},$$

$$\langle\phi \mid \psi\rangle \leftrightarrow \text{prodotto riga-per-colonna},$$

$$|u\rangle\langle v| \leftrightarrow \text{prodotto colonna-per-riga},$$

$$A \leftrightarrow \text{matrice dell'operatore in una base scelta}.$$

La relazione fondamentale tra un operatore astratto e i suoi elementi di matrice è

$$\boxed{A_{ij} = \langle e_i|A|e_j\rangle.}$$

La ricostruzione completa dell'operatore è

$$\boxed{A = \sum_{i,j} A_{ij}|e_i\rangle\langle e_j|.}$$

Queste formule mostrano che notazione di Dirac e calcolo matriciale non sono due formalismi alternativi: sono due livelli di descrizione della stessa algebra lineare.

Capitolo 4

Il qubit, la sfera di Bloch e l'entanglement

Questo capitolo introduce il qubit come sistema quantistico bidimensionale, ne sviluppa la rappresentazione mediante la sfera di Bloch e mostra in forma operativa come evoluzione unitaria, misura e osservabili agiscano sul caso più semplice. La seconda parte estende la descrizione ai sistemi composti di più qubit e introduce l'entanglement, analizzandone la struttura matematica, gli stati di Bell, le correlazioni di misura e il ruolo nel calcolo quantistico. Gli aspetti generali dei postulati della meccanica quantistica saranno ripresi e formalizzati nel capitolo successivo.

4.1 Il qubit

Definizione 4.1.1. Un *qubit* è un sistema quantistico il cui spazio di Hilbert è bidimensionale. Fissata una base ortonormale

$$\{|0\rangle, |1\rangle\},$$

detta *base computazionale*, uno stato puro arbitrario del qubit è rappresentato da un vettore normalizzato

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \qquad \alpha, \beta \in \mathbb{C},$$

con

$$|\alpha|^2 + |\beta|^2 = 1.$$

Vettori normalizzati che differiscono soltanto per una fase globale rappresentano lo stesso stato fisico, cioè lo stesso raggio dello spazio di Hilbert.

Nella base computazionale

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

per cui

$$|\psi\rangle = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}.$$

I coefficienti α e β sono *ampiezze di probabilità*. Non sono probabilità: in generale sono numeri complessi. Se il qubit viene misurato nella base computazionale, le probabilità dei due risultati

sono

$$p(0) = |\alpha|^2 , \qquad p(1) = |\beta|^2 .$$

La normalizzazione garantisce infatti

$$p(0) + p(1) = |\alpha|^2 + |\beta|^2 = 1.$$

Lo stato

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

è una sovrapposizione coerente degli stati di base.

Due esempi fondamentali sono

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

Vediamo esplicitamente come si ottiene la loro rappresentazione matriciale nella base computazionale. Ricordiamo che

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Per lo stato $|+\rangle$ si ha

$$\begin{aligned} |+\rangle &= \frac{1}{\sqrt{2}} \left(|0\rangle + |1\rangle \right) \\ &= \frac{1}{\sqrt{2}} \left[\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right] \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1+0 \\ 0+1 \end{pmatrix} \\ &= \boxed{\frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}}. \end{aligned}$$

Analogamente, per lo stato $|-\rangle$:

$$\begin{aligned} |-\rangle &= \frac{1}{\sqrt{2}} \left(|0\rangle - |1\rangle \right) \\ &= \frac{1}{\sqrt{2}} \left[\begin{pmatrix} 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right] \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1-0 \\ 0-1 \end{pmatrix} \\ &= \boxed{\frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}}. \end{aligned}$$

Il fattore $1/\sqrt{2}$ è necessario per normalizzare i due vettori. Infatti, per $|+\rangle$:

$$\begin{aligned}\langle + \mid + \rangle &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \frac{1}{2} \left(1 \cdot 1 + 1 \cdot 1\right) \\ &= \frac{1}{2}(2) \\ &= 1.\end{aligned}$$

Per $|-\rangle$ si ottiene nello stesso modo

$$\begin{aligned}\langle - \mid - \rangle &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \\ &= \frac{1}{2} \left(1 \cdot 1 + (-1)(-1)\right) \\ &= \frac{1}{2}(2) \\ &= 1.\end{aligned}$$

Pertanto, nella base computazionale,

$$\boxed{|+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \qquad |-\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}}.$$

Il bra associato a $|+\rangle$ è

$$\langle + | = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \end{pmatrix},$$

poiché le componenti sono reali. Il prodotto interno tra i due stati è quindi

$$\begin{aligned}\langle + \mid - \rangle &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \\ &= \frac{1}{2} \left(1 \cdot 1 + 1 \cdot (-1)\right) \\ &= \frac{1}{2}(1 - 1) \\ &= 0.\end{aligned}$$

Pertanto

$$\boxed{\langle + \mid - \rangle = 0},$$

cioè $|+\rangle$ e $|-\rangle$ sono ortogonali.

Entrambi danno 0 e 1 con probabilità $1/2$ se misurati nella base computazionale, ma sono stati fisici distinti: ciò mostra che le sole probabilità $|\alpha|^2$ e $|\beta|^2$ non determinano completamente lo stato; conta anche la fase relativa tra le ampiezze.

Fase globale e relativa

Consideriamo uno stato normalizzato

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

e lo stato

$$|\psi'\rangle = e^{\mathrm{i}\gamma}|\psi\rangle, \qquad \gamma \in \mathbb{R}.$$

Il fattore $e^{\mathrm{i}\gamma}$ è detto *fase globale*, perché moltiplica nello stesso modo tutte le componenti dello stato.

Per mostrare che $|\psi\rangle$ e $|\psi'\rangle$ rappresentano lo stesso stato fisico, consideriamo una misura proiettiva arbitraria associata a un proiettore P. La probabilità del corrispondente risultato per lo stato $|\psi'\rangle$ è

$$p' = \langle\psi'|P|\psi'\rangle.$$

Dal momento che

$$|\psi'\rangle = e^{\mathrm{i}\gamma}|\psi\rangle,$$

il bra associato è

$$\langle\psi'| = \left(e^{\mathrm{i}\gamma}|\psi\rangle\right)^\dagger = e^{-\mathrm{i}\gamma}\langle\psi|.$$

Pertanto

$$\begin{aligned} p' &= \left(e^{-\mathrm{i}\gamma}\langle\psi|\right) P \left(e^{\mathrm{i}\gamma}|\psi\rangle\right) \\ &= e^{-\mathrm{i}\gamma}e^{\mathrm{i}\gamma}\langle\psi|P|\psi\rangle \\ &= e^{0}\langle\psi|P|\psi\rangle \\ &= \langle\psi|P|\psi\rangle \\ &= p. \end{aligned}$$

Quindi tutte le probabilità di misura sono identiche per $|\psi\rangle$ e $e^{\mathrm{i}\gamma}|\psi\rangle$. Nessuna misura può distinguere i due vettori. Per questo essi rappresentano lo stesso stato fisico puro.

La stessa conclusione si può esprimere mediante il proiettore di stato:

$$\begin{aligned} |\psi'\rangle\langle\psi'| &= \left(e^{\mathrm{i}\gamma}|\psi\rangle\right)\left(e^{-\mathrm{i}\gamma}\langle\psi|\right) \\ &= e^{\mathrm{i}\gamma}e^{-\mathrm{i}\gamma}|\psi\rangle\langle\psi| \\ &= |\psi\rangle\langle\psi|. \end{aligned}$$

Quindi il proiettore che rappresenta lo stato fisico è invariato.

Al contrario, una *fase relativa* non può essere eliminata in questo modo. Per esempio $|+\rangle$ e $|-\rangle$ differiscono per il segno della seconda componente e, come abbiamo appena calcolato,

$$\langle + \mid - \rangle = 0.$$

Sono dunque stati fisicamente distinti.

4.2 La sfera di Bloch

Partiamo dallo stato puro più generale di un qubit:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \qquad \alpha, \beta \in \mathbb{C},$$

con

$$|\alpha|^2 + |\beta|^2 = 1.$$

A prima vista α e β contengono quattro parametri reali, perché ogni numero complesso richiede una parte reale e una parte immaginaria. Scriviamoli infatti in forma polare:

$$\alpha = r_0 e^{\mathrm{i}\gamma_0}, \qquad \beta = r_1 e^{\mathrm{i}\gamma_1},$$

con

$$r_0, r_1 \geq 0, \qquad \gamma_0, \gamma_1 \in \mathbb{R}.$$

La condizione di normalizzazione diventa

$$\begin{aligned} 1 &= |\alpha|^2 + |\beta|^2 \\ &= \left|r_0 e^{\mathrm{i}\gamma_0}\right|^2 + \left|r_1 e^{\mathrm{i}\gamma_1}\right|^2 \\ &= r_0^2 \left|e^{\mathrm{i}\gamma_0}\right|^2 + r_1^2 \left|e^{\mathrm{i}\gamma_1}\right|^2. \end{aligned}$$

Poiché

$$\left|e^{\mathrm{i}\gamma}\right| = 1,$$

si ottiene

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

Questa relazione elimina un grado di libertà reale. Possiamo quindi parametrizzare i due moduli mediante un solo angolo θ:

$$r_0 = \cos\frac{\theta}{2}, \qquad r_1 = \sin\frac{\theta}{2}, \qquad 0 \leq \theta \leq \pi.$$

La scelta dell'intervallo $0 \leq \theta \leq \pi$ garantisce che entrambi i moduli siano non negativi.

Sostituendo nello stato:

$$\begin{aligned} |\psi\rangle &= \cos\frac{\theta}{2} e^{\mathrm{i}\gamma_0}|0\rangle + \sin\frac{\theta}{2} e^{\mathrm{i}\gamma_1}|1\rangle \\ &= e^{\mathrm{i}\gamma_0}\left(\cos\frac{\theta}{2}|0\rangle + e^{\mathrm{i}(\gamma_1-\gamma_0)}\sin\frac{\theta}{2}|1\rangle\right). \end{aligned}$$

Abbiamo dimostrato nella sezione precedente che il fattore globale $e^{\mathrm{i}\gamma_0}$ non modifica lo stato fisico. Possiamo quindi ometterlo. Definiamo la fase relativa

$$\phi = \gamma_1 - \gamma_0,$$

con, per esempio,

$$0 \leq \phi < 2\pi.$$

Otteniamo così

$$\boxed{|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{\mathrm{i}\phi}\sin\frac{\theta}{2}|1\rangle}.$$

Il significato della parametrizzazione è quindi il seguente. I quattro parametri reali inizialmente contenuti in α e β si riducono a due parametri fisicamente rilevanti:

$$\theta, \qquad \phi.$$

La normalizzazione elimina un parametro e l'irrilevanza della fase globale ne elimina un secondo. Restano esattamente due parametri reali, che possono essere interpretati come gli angoli polare e azimutale di un punto sulla sfera unitaria.

Il fattore $1/2$ nell'argomento di seno e coseno è essenziale. Infatti, se

$$\theta = 0,$$

allora

$$|\psi\rangle = \cos 0\,|0\rangle + e^{\mathrm{i}\phi}\sin 0\,|1\rangle = |0\rangle.$$

Se invece

$$\theta = \pi,$$

si ha

$$\begin{aligned}|\psi\rangle &= \cos\frac{\pi}{2}|0\rangle + e^{\mathrm{i}\phi}\sin\frac{\pi}{2}|1\rangle \\ &= e^{\mathrm{i}\phi}|1\rangle,\end{aligned}$$

che rappresenta fisicamente lo stato $|1\rangle$ perché $e^{\mathrm{i}\phi}$ è una fase globale.

Coordinate cartesiane

Associamo allo stato il vettore reale

$$\mathbf{r} = (x, y, z),$$

con

$$x = \sin\theta\cos\phi, \qquad y = \sin\theta\sin\phi, \qquad z = \cos\theta.$$

Verifichiamo esplicitamente che il vettore ha norma unitaria:

$$\begin{aligned}x^2 + y^2 + z^2 &= \sin^2\theta\cos^2\phi + \sin^2\theta\sin^2\phi + \cos^2\theta \\ &= \sin^2\theta\left(\cos^2\phi + \sin^2\phi\right) + \cos^2\theta \\ &= \sin^2\theta + \cos^2\theta \\ &= 1.\end{aligned}$$

Quindi $\mathbf{r}$ appartiene alla superficie della sfera unitaria.

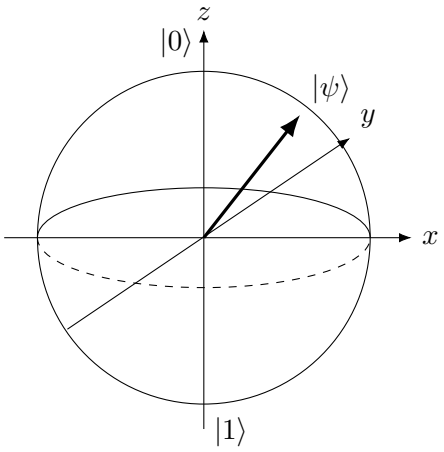


Figura 4.1: Rappresentazione schematica di uno stato puro di un qubit sulla sfera di Bloch.

Visualizzazione geometrica

Il polo nord rappresenta $|0\rangle$ e il polo sud rappresenta $|1\rangle$. Sull'equatore si ha $\theta = \pi/2$, quindi

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{\mathrm{i}\phi}|1\rangle\right).$$

Per $\phi = 0$ si ottiene

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

mentre per $\phi = \pi$:

$$\begin{aligned} |\psi\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{\mathrm{i}\pi}|1\rangle\right) \\ &= \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right) \\ &= |-\rangle. \end{aligned}$$

Per lo stato generale

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{\mathrm{i}\phi}\sin\frac{\theta}{2}|1\rangle,$$

le probabilità di misura nella base computazionale sono

$$\begin{aligned} p(0) &= \left|\cos\frac{\theta}{2}\right|^2 = \cos^2\frac{\theta}{2}, \\ p(1) &= \left|e^{\mathrm{i}\phi}\sin\frac{\theta}{2}\right|^2 \\ &= \left|e^{\mathrm{i}\phi}\right|^2 \sin^2\frac{\theta}{2} \\ &= \sin^2\frac{\theta}{2}. \end{aligned}$$

La sfera di Bloch non rappresenta lo spazio fisico in cui si trova il qubit: è uno spazio geometrico astratto che rappresenta gli stati puri di un sistema quantistico a due livelli.

4.3 Il qubit: applicazione operativa dei principi fondamentali

Il qubit permette di vedere in azione, in un caso bidimensionale, alcune regole che nel capitolo successivo saranno formulate in modo generale. L'obiettivo di questa sezione è quindi operativo: mostrare come una porta unitaria modifica le ampiezze e come una misura converte tali ampiezze in probabilità osservabili.

Evoluzione unitaria

L'evoluzione di un sistema quantistico chiuso è descritta da un operatore unitario U:

$$|\psi'\rangle = U|\psi\rangle, \qquad U^\dagger U = \mathbb{I}.$$

L'unitarietà garantisce, in particolare, che la norma dello stato sia preservata. Nel caso del qubit, la matrice di Pauli

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

scambia le due ampiezze:

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

In particolare, $X|0\rangle = |1\rangle$ e $X|1\rangle = |0\rangle$.

Misura e regola di Born

Consideriamo una misura nella base computazionale. I proiettori associati ai due possibili risultati sono

$$P_0 = |0\rangle\langle 0|, \qquad P_1 = |1\rangle\langle 1|, \qquad P_0 + P_1 = \mathbb{I}.$$

Per

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

la regola di Born fornisce

$$p(0) = \langle\psi|P_0|\psi\rangle = |\alpha|^2, \qquad p(1) = \langle\psi|P_1|\psi\rangle = |\beta|^2.$$

Se viene osservato il risultato m, il postulato di misura assegna allo stato post-misura

$$|\psi_m\rangle = \frac{P_m|\psi\rangle}{\sqrt{\langle\psi|P_m|\psi\rangle}}, \qquad p(m) > 0.$$

Nel caso della base computazionale, un esito 0 porta fisicamente allo stato $|0\rangle$ e un esito 1 allo stato $|1\rangle$. La derivazione generale della formula di normalizzazione sarà sviluppata nel capitolo dedicato ai principi della meccanica quantistica.

Misura in una base diversa

Consideriamo la base

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

Prepariamo il qubit nello stato $|0\rangle$. Calcoliamo la probabilità di ottenere il risultato associato a $|+\rangle$:

$$p(+) = |\langle + \mid 0\rangle|^2 .$$

Poiché

$$\langle +| = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \end{pmatrix}, \qquad |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

si ha

$$\begin{aligned} \langle + \mid 0\rangle &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \frac{1}{\sqrt{2}}(1 \cdot 1 + 1 \cdot 0) \\ &= \frac{1}{\sqrt{2}}. \end{aligned}$$

Pertanto

$$p(+) = \left| \frac{1}{\sqrt{2}} \right|^2 = \frac{1}{2}.$$

Analogamente

$$\langle -| = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \end{pmatrix},$$

e quindi

$$\begin{aligned} \langle - \mid 0\rangle &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \frac{1}{\sqrt{2}}(1 \cdot 1 - 1 \cdot 0) \\ &= \frac{1}{\sqrt{2}}. \end{aligned}$$

Di conseguenza

$$p(-) = \left| \frac{1}{\sqrt{2}} \right|^2 = \frac{1}{2}.$$

4.4 Osservabili nel caso del qubit

Una grandezza osservabile è rappresentata da un operatore hermitiano

$$A^\dagger = A.$$

Gli autovalori di A sono i possibili risultati della misura e gli autostati individuano le corrispondenti direzioni di misura. La dimostrazione generale della realtà degli autovalori e la formulazione spettrale saranno riprese nel capitolo successivo.

Per un qubit, l'esempio fondamentale è

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

che soddisfa $Z^\dagger = Z$ e ha autostati $|0\rangle$ e $|1\rangle$:

$$Z|0\rangle = +|0\rangle, \qquad Z|1\rangle = -|1\rangle.$$

Se

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

i risultati $+1$ e -1 hanno probabilità rispettivamente $|\alpha|^2$ e $|\beta|^2$. Il valore medio è quindi

$$\langle Z\rangle = (+1)|\alpha|^2 + (-1)|\beta|^2 = |\alpha|^2 - |\beta|^2.$$

Questo esempio mostra il collegamento fra autovalori, probabilità di Born e valore medio senza anticipare la trattazione generale degli osservabili.

4.5 Sistemi composti di qubit

Per definire l'entanglement è necessario passare da un singolo qubit a un sistema formato da due o più qubit. Il punto matematico essenziale è che lo spazio degli stati del sistema composto non è una somma diretta degli spazi dei singoli sistemi, ma il loro *prodotto tensoriale*.

Due qubit

Per un singolo qubit lo spazio di Hilbert è bidimensionale e possiede la base computazionale

$$\{|0\rangle, |1\rangle\}.$$

Per due qubit, che indicheremo con A e B, lo spazio degli stati è

$$\mathcal{H}_A \otimes \mathcal{H}_B.$$

La base computazionale naturale del sistema composto è

$$\{|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle\}.$$

Per semplicità si usa la notazione

$$|00\rangle = |0\rangle \otimes |0\rangle, \qquad |01\rangle = |0\rangle \otimes |1\rangle,$$

$$|10\rangle = |1\rangle \otimes |0\rangle, \qquad |11\rangle = |1\rangle \otimes |1\rangle.$$

Nella base computazionale:

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \qquad |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix},$$

$$|10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \qquad |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

Vediamo esplicitamente, per esempio, come si ottiene $|01\rangle$. Poiché

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

si ha

$$\begin{aligned} |01\rangle &= |0\rangle \otimes |1\rangle \\ &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 0 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}. \end{aligned}$$

Stato generale di due qubit

Uno stato puro generale di due qubit ha la forma

$$|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle,$$

dove

$$a, b, c, d \in \mathbb{C}$$

e la normalizzazione richiede

$$|a|^2 + |b|^2 + |c|^2 + |d|^2 = 1.$$

Nella base computazionale:

$$|\psi\rangle = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}.$$

Se si esegue una misura nella base computazionale, le probabilità dei quattro risultati sono

$$p(00) = |a|^2, \qquad p(01) = |b|^2,$$

$$p(10) = |c|^2, \qquad p(11) = |d|^2.$$

Stati prodotto

Supponiamo che il qubit A sia nello stato

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

e il qubit B nello stato

$$|\psi_B\rangle = \gamma|0\rangle + \delta|1\rangle.$$

Lo stato composto è

$$|\psi_A\rangle \otimes |\psi_B\rangle.$$

Sviluppiamo completamente il prodotto:

$$\begin{aligned}|\psi_A\rangle \otimes |\psi_B\rangle &= (\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle) \\ &= \alpha\gamma\,|0\rangle \otimes |0\rangle + \alpha\delta\,|0\rangle \otimes |1\rangle \\ &\quad + \beta\gamma\,|1\rangle \otimes |0\rangle + \beta\delta\,|1\rangle \otimes |1\rangle \\ &= \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle.\end{aligned}$$

Quindi uno stato prodotto ha coefficienti

$$a = \alpha\gamma, \qquad b = \alpha\delta, \qquad c = \beta\gamma, \qquad d = \beta\delta.$$

4.6 Entanglement

Definizione 4.6.1. Uno stato puro di due qubit

$$|\psi\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$$

si dice *separabile*, o *stato prodotto*, se esistono due stati $|\psi_A\rangle$ e $|\psi_B\rangle$ tali che

$$|\psi\rangle = |\psi_A\rangle \otimes |\psi_B\rangle.$$

Se una tale fattorizzazione non esiste, lo stato si dice *entangled*.

L'entanglement è quindi una proprietà dello stato globale del sistema composto. Non significa semplicemente che i risultati di due misure siano correlati: anche sistemi classici possono presentare correlazioni. La caratteristica essenziale è che lo stato quantistico globale non può essere ricondotto a due stati quantistici indipendenti dei sottosistemi.

Criterio di separabilità per due qubit

Esempio 4.6.2. Consideriamo

$$|\psi\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle).$$

Raccogliamo i termini:

$$\begin{aligned}|\psi\rangle &= \frac{1}{2}\left[|0\rangle(|0\rangle + |1\rangle) + |1\rangle(|0\rangle + |1\rangle)\right] \\ &= \frac{1}{2}(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle).\end{aligned}$$

Poiché

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

si ottiene

$$\begin{aligned} |+\rangle \otimes |+\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ &= \frac{1}{2}(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \\ &= |\psi\rangle. \end{aligned}$$

Quindi

$$\boxed{|\psi\rangle = |+\rangle \otimes |+\rangle}$$

e lo stato è separabile.

Per uno stato puro generale

$$|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle,$$

esiste un criterio molto semplice per verificare la separabilità.

Se lo stato è prodotto,

$$|\psi\rangle = (\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle).$$

Ricaviamo ora esplicitamente i coefficienti a, b, c, d. Usiamo la bilinearità del prodotto tensoriale, cioè

$$(u + v) \otimes w = u \otimes w + v \otimes w,$$

e

$$u \otimes (w + z) = u \otimes w + u \otimes z,$$

insieme alla possibilità di portare fuori gli scalari complessi:

$$(\lambda u) \otimes (\mu v) = \lambda\mu\,(u \otimes v).$$

Partendo dallo stato prodotto si ha quindi

$$\begin{aligned} |\psi\rangle &= (\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle) \\ &= \alpha|0\rangle \otimes (\gamma|0\rangle + \delta|1\rangle) + \beta|1\rangle \otimes (\gamma|0\rangle + \delta|1\rangle) \\ &= \alpha\gamma\,|0\rangle \otimes |0\rangle + \alpha\delta\,|0\rangle \otimes |1\rangle + \beta\gamma\,|1\rangle \otimes |0\rangle + \beta\delta\,|1\rangle \otimes |1\rangle. \end{aligned}$$

Usando le abbreviazioni

$$\begin{aligned} |00\rangle &= |0\rangle \otimes |0\rangle, \qquad |01\rangle = |0\rangle \otimes |1\rangle, \\ |10\rangle &= |1\rangle \otimes |0\rangle, \qquad |11\rangle = |1\rangle \otimes |1\rangle, \end{aligned}$$

segue

$$|\psi\rangle = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle.$$

Confrontando termine per termine con la forma generale

$$|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle,$$

si ottiene necessariamente

$$\boxed{a = \alpha\gamma, \qquad b = \alpha\delta, \qquad c = \beta\gamma, \qquad d = \beta\delta}.$$

Calcoliamo:

$$\begin{aligned} ad - bc &= (\alpha\gamma)(\beta\delta) - (\alpha\delta)(\beta\gamma) \\ &= \alpha\beta\gamma\delta - \alpha\beta\delta\gamma \\ &= 0. \end{aligned}$$

Pertanto ogni stato prodotto deve soddisfare

$$ad - bc = 0.$$

Per due qubit puri vale anche il viceversa: se

$$ad - bc = 0,$$

lo stato è separabile.

Quindi:

$$\boxed{|\psi\rangle \text{ separabile} \quad \Longleftrightarrow \quad ad - bc = 0}.$$

Esempio 4.6.3. Consideriamo

$$|\psi\rangle = \frac{1}{\sqrt{10}}\left(|00\rangle + 2|01\rangle + |10\rangle + 2|11\rangle\right).$$

I coefficienti sono

$$a = \frac{1}{\sqrt{10}}, \qquad b = \frac{2}{\sqrt{10}}, \qquad c = \frac{1}{\sqrt{10}}, \qquad d = \frac{2}{\sqrt{10}}.$$

Calcoliamo:

$$\begin{aligned} ad - bc &= \frac{1}{\sqrt{10}}\frac{2}{\sqrt{10}} - \frac{2}{\sqrt{10}}\frac{1}{\sqrt{10}} \\ &= \frac{2}{10} - \frac{2}{10} \\ &= 0. \end{aligned}$$

Lo stato è quindi separabile. Infatti:

$$\begin{aligned} |\psi\rangle &= \frac{1}{\sqrt{10}}(|0\rangle + |1\rangle) \otimes (|0\rangle + 2|1\rangle) \\ &= \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle + 2|1\rangle}{\sqrt{5}}\right). \end{aligned}$$

4.7 Gli stati di Bell

Gli esempi più semplici e importanti di stati entangled di due qubit sono gli *stati di Bell*:

$$|\Phi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}, \qquad |\Phi^-\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}},$$

$$|\Psi^+\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}}, \qquad |\Psi^-\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}.$$

Essi formano una base ortonormale dello spazio di Hilbert di due qubit. Vediamo con precisione perché.

Lo spazio di Hilbert di due qubit è

$$\mathcal{H}_A \otimes \mathcal{H}_B,$$

e, poiché ciascuno dei due spazi dei singoli qubit ha dimensione 2, si ha

$$\dim(\mathcal{H}_A \otimes \mathcal{H}_B) = 2 \cdot 2 = 4.$$

Per costituire una base è quindi sufficiente mostrare che i quattro stati di Bell sono quattro vettori ortonormali: quattro vettori ortonormali in uno spazio di dimensione quattro sono automaticamente linearmente indipendenti e, essendo quattro, generano l'intero spazio.

Verifichiamo innanzitutto la normalizzazione. Per $|\Phi^+\rangle$:

$$\begin{aligned}
\langle \Phi^+ \mid \Phi^+ \rangle &= \frac{1}{2}(\langle 00| + \langle 11|)(|00\rangle + |11\rangle) \\
&= \frac{1}{2}\left(\langle 00 \mid 00\rangle + \langle 00 \mid 11\rangle + \langle 11 \mid 00\rangle + \langle 11 \mid 11\rangle\right) \\
&= \frac{1}{2}(1 + 0 + 0 + 1) \\
&= 1.
\end{aligned}$$

Analogamente,

$$\begin{aligned}
\langle \Phi^- \mid \Phi^- \rangle &= \frac{1}{2}(\langle 00| - \langle 11|)(|00\rangle - |11\rangle) \\
&= \frac{1}{2}(1 - 0 - 0 + 1) \\
&= 1,
\end{aligned}$$

$$\begin{aligned}
\langle \Psi^+ \mid \Psi^+ \rangle &= \frac{1}{2}(\langle 01| + \langle 10|)(|01\rangle + |10\rangle) \\
&= \frac{1}{2}(1 + 0 + 0 + 1) \\
&= 1,
\end{aligned}$$

e

$$\begin{aligned}\langle \Psi^- \mid \Psi^- \rangle &= \frac{1}{2}(\langle 01| - \langle 10|)(|01\rangle - |10\rangle) \\ &= \frac{1}{2}(1 - 0 - 0 + 1) \\ &= 1.\end{aligned}$$

Mostriamo ora l'ortogonalità reciproca. Per i due stati Φ:

$$\begin{aligned}\langle \Phi^+ \mid \Phi^- \rangle &= \frac{1}{2}(\langle 00| + \langle 11|)(|00\rangle - |11\rangle) \\ &= \frac{1}{2}(1 - 0 + 0 - 1) \\ &= 0.\end{aligned}$$

Per i due stati Ψ:

$$\begin{aligned}\langle \Psi^+ \mid \Psi^- \rangle &= \frac{1}{2}(\langle 01| + \langle 10|)(|01\rangle - |10\rangle) \\ &= \frac{1}{2}(1 - 0 + 0 - 1) \\ &= 0.\end{aligned}$$

Resta da verificare che ogni stato di tipo Φ sia ortogonale a ogni stato di tipo Ψ. Per esempio:

$$\begin{aligned}\langle \Phi^+ \mid \Psi^+ \rangle &= \frac{1}{2}(\langle 00| + \langle 11|)(|01\rangle + |10\rangle) \\ &= \frac{1}{2}\left(\langle 00 \mid 01\rangle + \langle 00 \mid 10\rangle + \langle 11 \mid 01\rangle + \langle 11 \mid 10\rangle\right) \\ &= 0.\end{aligned}$$

Lo stesso accade per le altre tre combinazioni:

$$\begin{aligned}\langle \Phi^+ \mid \Psi^- \rangle &= \frac{1}{2}(0 - 0 + 0 - 0) = 0, \\ \langle \Phi^- \mid \Psi^+ \rangle &= \frac{1}{2}(0 + 0 - 0 - 0) = 0, \\ \langle \Phi^- \mid \Psi^- \rangle &= \frac{1}{2}(0 - 0 - 0 + 0) = 0.\end{aligned}$$

Abbiamo quindi dimostrato che

$$\langle B_i \mid B_j \rangle = \delta_{ij},$$

dove $|B_i\rangle$ e $|B_j\rangle$ indicano due qualunque dei quattro stati di Bell. I quattro stati sono dunque ortonormali. Poiché lo spazio di due qubit ha dimensione quattro, essi formano necessariamente una base completa:

$$\boxed{\{|\Phi^+\rangle, |\Phi^-\rangle, |\Psi^+\rangle, |\Psi^-\rangle\} \text{ è una base ortonormale di } \mathcal{H}_A \otimes \mathcal{H}_B}.$$

$|\Phi^+\rangle$ è entangled

Consideriamo

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

Nella forma generale

$$a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$$

abbiamo

$$a = \frac{1}{\sqrt{2}}, \qquad b = 0, \qquad c = 0, \qquad d = \frac{1}{\sqrt{2}}.$$

Calcoliamo il criterio di separabilità:

$$\begin{aligned} ad - bc &= \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}} - 0 \cdot 0 \\ &= \frac{1}{2}. \end{aligned}$$

Poiché

$$\frac{1}{2} \neq 0,$$

lo stato non è separabile:

$$\boxed{|\Phi^+\rangle \text{ è entangled.}}$$

Possiamo vedere la stessa cosa tentando direttamente una fattorizzazione. Supponiamo per assurdo che

$$|\Phi^+\rangle = (\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle).$$

Sviluppando il secondo membro:

$$|\Phi^+\rangle = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle.$$

Confrontando i coefficienti:

$$\begin{aligned} \alpha\gamma &= \frac{1}{\sqrt{2}}, \\ \alpha\delta &= 0, \\ \beta\gamma &= 0, \\ \beta\delta &= \frac{1}{\sqrt{2}}. \end{aligned}$$

Dalla prima equazione segue

$$\alpha \neq 0, \qquad \gamma \neq 0.$$

Poiché

$$\alpha\delta = 0$$

e $\alpha \neq 0$, deve essere

$$\delta = 0.$$

Ma allora

$$\beta\delta = 0,$$

in contraddizione con

$$\beta\delta = \frac{1}{\sqrt{2}}.$$

La fattorizzazione è impossibile.

Entanglement e misura

Consideriamo

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

Una misura dei due qubit nella base computazionale può produrre soltanto

$$00 \quad \text{oppure} \quad 11.$$

Le ampiezze sono

$$a_{00} = \frac{1}{\sqrt{2}}, \qquad a_{11} = \frac{1}{\sqrt{2}},$$

mentre

$$a_{01} = a_{10} = 0.$$

Per la regola di Born:

$$p(00) = \left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2},$$

$$p(11) = \left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2},$$

e

$$p(01) = p(10) = 0.$$

Quindi ciascun singolo risultato è casuale, ma i due risultati sono perfettamente correlati.

Supponiamo di misurare soltanto il primo qubit.

Il proiettore corrispondente al risultato 0 sul primo qubit è

$$P_0 = |0\rangle\langle 0| \otimes \mathbb{I}.$$

Applicandolo:

$$\begin{aligned} P_0|\Phi^+\rangle &= (|0\rangle\langle 0| \otimes \mathbb{I})\frac{|00\rangle + |11\rangle}{\sqrt{2}} \\ &= \frac{1}{\sqrt{2}}\left[(|0\rangle\langle 0| \otimes \mathbb{I})|00\rangle + (|0\rangle\langle 0| \otimes \mathbb{I})|11\rangle\right]. \end{aligned}$$

Il primo termine è

$$\begin{aligned} (|0\rangle\langle 0| \otimes \mathbb{I})|00\rangle &= (|0\rangle\langle 0||0\rangle) \otimes (\mathbb{I}|0\rangle) \\ &= |0\rangle\langle 0 \mid 0\rangle \otimes |0\rangle \\ &= |0\rangle \otimes |0\rangle \\ &= |00\rangle. \end{aligned}$$

Il secondo termine è

$$\begin{aligned}(|0\rangle\langle 0| \otimes \mathbb{I})|11\rangle &= (|0\rangle\langle 0||1\rangle) \otimes (\mathbb{I}|1\rangle) \\ &= |0\rangle\langle 0 \mid 1\rangle \otimes |1\rangle \\ &= 0.\end{aligned}$$

Quindi

$$P_0|\Phi^+\rangle = \frac{|00\rangle}{\sqrt{2}}.$$

La probabilità del risultato 0 è

$$\begin{aligned}p(0) &= \left\| P_0|\Phi^+\rangle \right\|^2 \\ &= \left\| \frac{|00\rangle}{\sqrt{2}} \right\|^2 \\ &= \frac{1}{2}.\end{aligned}$$

Dopo normalizzazione lo stato globale diventa

$$|00\rangle.$$

Analogamente, se il primo qubit viene misurato come 1, lo stato globale diventa

$$|11\rangle.$$

Da quel momento una misura del secondo qubit nella stessa base darà con certezza lo stesso risultato.

4.8 Entanglement e correlazioni classiche

È utile distinguere entanglement e semplice correlazione.

Supponiamo che una sorgente classica produca, con probabilità 1/2, la coppia

$$00$$

e, con probabilità 1/2, la coppia

$$11.$$

I due bit sono perfettamente correlati. Tuttavia la sorgente non produce una sovrapposizione coerente: in ogni singola realizzazione la coppia è già 00 oppure 11.

Lo stato quantistico

$$|\Phi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

contiene invece una relazione di fase coerente fra le due componenti. Questa differenza diventa osservabile quando si considerano misure in basi diverse dalla base computazionale ed è alla base delle correlazioni non classiche testate dalle disuguaglianze di Bell.

L'entanglement non va quindi definito come "correlazione molto forte", ma come non separabilità dello stato quantistico.

4.9 Entanglement e calcolo quantistico

Un sistema di n qubit ha uno spazio degli stati di dimensione

$$2^n.$$

Uno stato generale è

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

con

$$\sum_x |\alpha_x|^2 = 1.$$

Se tutti i qubit rimanessero sempre separabili, lo stato avrebbe la forma

$$|\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle,$$

e potrebbe essere descritto mediante un numero di parametri che cresce solo linearmente con n.

L'entanglement consente invece di occupare regioni dello spazio degli stati che non ammettono questa fattorizzazione. In questo senso permette al registro quantistico di sviluppare correlazioni globali genuine fra molti qubit.

L'importanza computazionale dell'entanglement è più precisa della semplice affermazione secondo cui "i qubit sono correlati".

L'entanglement è una risorsa importante, ma la sua sola presenza non basta a garantire un vantaggio computazionale. Esistono classi di evoluzioni con entanglement limitato che possono essere simulate efficientemente con metodi classici; inoltre, circuiti di tipo stabilizer possono generare stati entangled pur rimanendo classicamente simulabili. Il vantaggio nasce quindi dalla struttura complessiva della computazione: sovrapposizione, fasi, interferenza, entanglement e scelta delle operazioni devono cooperare in modo da rendere accessibile l'informazione desiderata.

Entanglement, interferenza e algoritmi quantistici

Nei circuiti quantistici l'entanglement non lavora isolatamente. Un algoritmo prepara sovrapposizioni, applica trasformazioni che costruiscono correlazioni fra qubit e infine sfrutta l'interferenza per amplificare o sopprimere determinate ampiezze.

Per questo motivo è fuorviante descrivere la potenza di un computer quantistico dicendo semplicemente che "calcola tutte le risposte contemporaneamente". La sovrapposizione crea molte ampiezze, ma una misura non permette di leggerle tutte. L'algoritmo deve organizzare le fasi e le correlazioni in modo che l'interferenza renda statisticamente accessibile l'informazione desiderata.

L'entanglement è uno dei meccanismi attraverso cui questa struttura globale dello stato può essere costruita.

4.10 Entanglement e fragilità

La stessa caratteristica che rende utile l'entanglement lo rende anche delicato.

Se i qubit di un registro interagiscono in modo incontrollato con l'ambiente, il sistema può diventare entangled con gradi di libertà esterni. Dal punto di vista del registro, ciò si manifesta come perdita di coerenza e degradazione delle correlazioni quantistiche utili.

Per questo il controllo dell'entanglement è duplice:

- bisogna generare intenzionalmente l'entanglement richiesto dal circuito;
- bisogna limitare l'entanglement indesiderato con l'ambiente e con gradi di libertà non controllati.

Questa osservazione prepara naturalmente lo studio successivo di rumore, decoerenza e correzione quantistica degli errori.

Ricapitolando, l'entanglement compare quando lo stato di un sistema composto non può essere fattorizzato negli stati dei singoli sottosistemi. Per due qubit puri,

$$|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle,$$

il criterio

$$ad - bc \neq 0$$

segnala l'entanglement.

Gli stati di Bell rappresentano gli esempi elementari fondamentali:

$$|\Phi^{\pm}\rangle = \frac{|00\rangle \pm |11\rangle}{\sqrt{2}}, \qquad |\Psi^{\pm}\rangle = \frac{|01\rangle \pm |10\rangle}{\sqrt{2}}.$$

Essi mostrano che i risultati locali possono essere casuali pur possedendo correlazioni globali perfette. Hadamard e CNOT permettono di costruire facilmente una coppia di Bell a partire da $|00\rangle$.

Dal punto di vista computazionale, l'entanglement permette di costruire correlazioni non fattorizzabili fra molti qubit e costituisce una risorsa essenziale in numerosi protocolli quantistici. Risultati teorici mostrano che una crescita adeguata dell'entanglement è necessaria per certe forme di accelerazione esponenziale nelle computazioni pure e che sistemi con entanglement fortemente limitato possono risultare simulabili classicamente. Tuttavia l'entanglement non è da solo sufficiente: esistono circuiti altamente entangled, come i circuiti stabilizer, che sono efficientemente simulabili su un computer classico.

La conclusione corretta è quindi più precisa:

l'entanglement è una risorsa fondamentale, ma non è l'unica risorsa del vantaggio quantistico.

Capitolo 5

Principi della meccanica quantistica rilevanti per il calcolo quantistico

Questo capitolo riprende in forma generale i principi già incontrati nel caso del qubit e introduce gli aspetti necessari per descrivere vincoli e fenomeni che non emergono dalla sola rappresentazione sulla sfera di Bloch. L'obiettivo non è fornire un corso generale di meccanica quantistica, ma isolare le strutture che diventano strumenti operativi nei circuiti e negli algoritmi: evoluzione unitaria, misura, osservabili, incompatibilità, *no-cloning theorem*, distinguibilità, stati misti, sottosistemi e decoerenza.

5.1 Stati puri, sovrapposizione e fase: richiamo

Uno stato puro è rappresentato da un vettore normalizzato $|\psi\rangle$ di uno spazio vettoriale complesso con prodotto interno:

$$\langle\psi \mid \psi\rangle = 1.$$

Il principio di sovrapposizione afferma che una combinazione lineare normalizzata di stati appartiene ancora allo spazio degli stati. Come visto nel capitolo precedente per il qubit, la fase globale non modifica lo stato fisico:

$$|\psi\rangle \sim e^{\mathrm{i}\gamma}|\psi\rangle,$$

mentre le fasi relative fra componenti diverse sono fisicamente significative perché possono modificare i fenomeni di interferenza.

In termini più precisi, uno stato puro fisico corrisponde al raggio associato a $|\psi\rangle$: vettori normalizzati che differiscono soltanto per una fase globale rappresentano lo stesso stato fisico. Nei paragrafi seguenti non ripeteremo i calcoli già svolti per il qubit, ma useremo questi risultati nella formulazione generale.

5.2 Evoluzione unitaria

L'evoluzione di un sistema quantistico chiuso è descritta da un operatore unitario U:

$$|\psi'\rangle = U|\psi\rangle.$$

Per definizione,

$$U^\dagger U = U U^\dagger = \mathbb{I},$$

quindi

$$U^{-1} = U^\dagger.$$

L'evoluzione è reversibile.

Conservazione del prodotto interno

L'unitarietà conserva non soltanto la norma, ma l'intero prodotto interno. Se $|\psi'\rangle = U|\psi\rangle$ e $|\phi'\rangle = U|\phi\rangle$, allora

$$\langle \phi' \mid \psi' \rangle = \langle \phi | U^\dagger U | \psi \rangle = \langle \phi \mid \psi \rangle.$$

Ponendo $|\phi\rangle = |\psi\rangle$ si ottiene immediatamente la conservazione della normalizzazione. Questa proprietà sarà usata più avanti nella dimostrazione del *no-cloning theorem*.

Collegamento con l'equazione di Schrödinger

Nel formalismo continuo,

$$\mathrm{i}\hbar \frac{d}{dt} |\psi(t)\rangle = H(t) |\psi(t)\rangle,$$

dove H è l'Hamiltoniana. Se H è indipendente dal tempo,

$$|\psi(t)\rangle = e^{-\mathrm{i}Ht/\hbar} |\psi(0)\rangle.$$

Poiché H è hermitiano, $H^\dagger = H$, e dunque

$$U(t) = e^{-\mathrm{i}Ht/\hbar}$$

è unitario. Nel modello a circuiti questa evoluzione viene idealizzata come una successione discreta di porte unitarie.

5.3 La misura

Una misura proiettiva è descritta da una famiglia di proiettori ortogonali $\{P_m\}$ che soddisfano

$$P_m^\dagger = P_m, \qquad P_m^2 = P_m, \qquad \sum_m P_m = \mathbb{I}.$$

Se il sistema è nello stato normalizzato $|\psi\rangle$, la regola di Born assegna all'esito m la probabilità

$$\boxed{p(m) = \langle \psi | P_m | \psi \rangle.}$$

La relazione di completezza garantisce che le probabilità sommino a uno:

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

Nel caso della misura computazionale di un qubit si recuperano, come visto in precedenza, $p(0) = |\alpha|^2$ e $p(1) = |\beta|^2$.

Stato dopo la misura

Nel formalismo della misura proiettiva, lo stato dopo la misura non si ricava dalla sola regola di Born. La regola di Born stabilisce le probabilità dei possibili risultati; la trasformazione dello stato in seguito alla misura è invece parte del postulato di misura.

Supponiamo che prima della misura il sistema sia nello stato normalizzato

$$|\psi\rangle, \qquad \langle\psi \mid \psi\rangle = 1,$$

e che il risultato osservato sia quello associato al proiettore P_m.

Il primo passo consiste nel selezionare la componente di $|\psi\rangle$ che appartiene al sottospazio associato al risultato m. Questa operazione è realizzata precisamente dal proiettore:

$$|\widetilde{\psi}_m\rangle = P_m|\psi\rangle.$$

La tilde ricorda che, in generale, il vettore così ottenuto non è ancora normalizzato.

Per ottenere uno stato fisico valido dobbiamo quindi dividerlo per la sua norma. Calcoliamo tale norma in modo esplicito:

$$\|P_m|\psi\rangle\|^2 = (P_m|\psi\rangle)^\dagger \, (P_m|\psi\rangle)\,.$$

Poiché

$$(P_m|\psi\rangle)^\dagger = \langle\psi|P_m^\dagger,$$

segue

$$\|P_m|\psi\rangle\|^2 = \langle\psi|P_m^\dagger P_m|\psi\rangle.$$

Per un proiettore ortogonale valgono

$$P_m^\dagger = P_m, \qquad P_m^2 = P_m.$$

Di conseguenza

$$P_m^\dagger P_m = P_m P_m = P_m,$$

e quindi

$$\boxed{\|P_m|\psi\rangle\|^2 = \langle\psi|P_m|\psi\rangle.}$$

La regola di Born assegna al risultato m la probabilità

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

Pertanto

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

e dunque

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

Il vettore proiettato si normalizza dividendo per questa quantità:

$$|\psi_m\rangle = \frac{P_m|\psi\rangle}{\|P_m|\psi\rangle\|}.$$

Sostituendo l'espressione della norma si ottiene infine

$$\boxed{|\psi_m\rangle = \frac{P_m|\psi\rangle}{\sqrt{\langle\psi|P_m|\psi\rangle}}.}$$

La formula dello stato post-misura contiene quindi due operazioni concettualmente distinte:

$$|\psi\rangle \xrightarrow{P_m} P_m|\psi\rangle \xrightarrow{\text{normalizzazione}} \frac{P_m|\psi\rangle}{\sqrt{\langle\psi|P_m|\psi\rangle}}.$$

Il primo passaggio seleziona la componente compatibile con il risultato osservato; il secondo riporta il vettore a norma uno.

Esempio 5.3.1. Consideriamo un qubit nello stato

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \qquad |\alpha|^2 + |\beta|^2 = 1,$$

e supponiamo che una misura nella base computazionale dia come risultato 0. Il proiettore associato è

$$P_0 = |0\rangle\langle 0|.$$

Applichiamolo allo stato:

$$\begin{aligned} P_0|\psi\rangle &= |0\rangle\langle 0|\,(\alpha|0\rangle + \beta|1\rangle) \\ &= \alpha|0\rangle\langle 0 \mid 0\rangle + \beta|0\rangle\langle 0 \mid 1\rangle. \end{aligned}$$

Poiché

$$\langle 0 \mid 0\rangle = 1, \qquad \langle 0 \mid 1\rangle = 0,$$

si ottiene

$$P_0|\psi\rangle = \alpha|0\rangle.$$

La norma del vettore proiettato è

$$\|\alpha|0\rangle\| = |\alpha|.$$

Equivalentemente,

$$\sqrt{\langle\psi|P_0|\psi\rangle} = \sqrt{|\alpha|^2} = |\alpha|.$$

Lo stato normalizzato dopo la misura è quindi

$$|\psi_0\rangle = \frac{\alpha|0\rangle}{|\alpha|} = \frac{\alpha}{|\alpha|}|0\rangle.$$

Scrivendo

$$\alpha = |\alpha|e^{\mathrm{i}\gamma},$$

si ha

$$\frac{\alpha}{|\alpha|} = e^{\mathrm{i}\gamma},$$

e pertanto

$$|\psi_0\rangle = e^{\mathrm{i}\gamma}|0\rangle.$$

Il fattore $e^{i\gamma}$ è una fase globale e non modifica lo stato fisico. Di conseguenza, fisicamente,

$$\boxed{|\psi_0\rangle \equiv |0\rangle.}$$

Per lo stesso motivo, se il risultato della misura è 1, lo stato post-misura è fisicamente $|1\rangle$.

5.4 Osservabili e operatori hermitiani

Nel formalismo standard, una grandezza osservabile è rappresentata da un operatore hermitiano

$$A = A^\dagger.$$

Gli operatori hermitiani hanno autovalori reali e una decomposizione spettrale ortogonale.

Perché gli autovalori sono reali

Supponiamo

$$A|a\rangle = a|a\rangle.$$

Moltiplicando a sinistra per $\langle a|$,

$$\langle a|A|a\rangle = a\langle a \mid a\rangle.$$

Coniugando,

$$(\langle a|A|a\rangle)^* = a^*\langle a \mid a\rangle.$$

Ma

$$(\langle a|A|a\rangle)^* = \langle a|A^\dagger|a\rangle = \langle a|A|a\rangle.$$

Quindi

$$a\langle a \mid a\rangle = a^*\langle a \mid a\rangle.$$

Poiché $\langle a \mid a\rangle > 0$,

$$\boxed{a = a^*.}$$

Dunque a è reale.

Valore medio

Se A è un osservabile,

$$\boxed{\langle A\rangle_\psi = \langle\psi|A|\psi\rangle.}$$

Se

$$A = \sum_j a_j P_j,$$

allora

$$\langle A\rangle_\psi = \sum_j a_j \langle\psi|P_j|\psi\rangle,$$

cioè la media degli esiti pesata con le probabilità di Born.

5.5 Non commutatività e principio di indeterminazione

Per due osservabili A e B si definisce

$$[A, B] = AB - BA.$$

Se $[A, B] \neq 0$, non esiste in generale una base completa di autovettori comuni.

Le deviazioni standard sono

$$(\Delta A)^2 = \langle A^2 \rangle - \langle A \rangle^2,$$

e analogamente per B. Vale la relazione di Robertson

$$\boxed{\Delta A \, \Delta B \geq \frac{1}{2} \left| \langle \psi | [A, B] | \psi \rangle \right|.}$$

Se due osservabili non commutano, non esiste in generale una base completa di autostati comuni. Di conseguenza, uno stato quantistico non può essere interpretato, in generale, come se assegnasse simultaneamente a entrambe le osservabili valori determinati preesistenti alla misura, come accadrebbe per normali variabili classiche. La non commutatività implica inoltre che una misura di una delle due osservabili può modificare lo stato in modo da rendere indeterminato il risultato di una successiva misura dell'altra.

Esempio 5.5.1. Un esempio particolarmente semplice è fornito dalle matrici di Pauli

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Esse non commutano. Infatti

$$XZ = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \qquad ZX = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix},$$

e quindi

$$[X, Z] = XZ - ZX = \begin{pmatrix} 0 & -2 \\ 2 & 0 \end{pmatrix} \neq 0.$$

Supponiamo ora che il qubit sia nello stato

$$|0\rangle.$$

Questo è un autostato di Z, perché

$$Z|0\rangle = |0\rangle.$$

Una misura di Z restituisce dunque con certezza l'autovalore $+1$.
Lo stesso stato non è invece un autostato di X. Infatti

$$X|0\rangle = |1\rangle,$$

che non è proporzionale a $|0\rangle$. Per esprimere $|0\rangle$ nella base degli autostati di X, ricordiamo

che

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

da cui

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

Pertanto una misura di X sullo stato $|0\rangle$ fornisce

$$+1 \qquad \text{con probabilità} \qquad \frac{1}{2},$$

oppure

$$-1 \qquad \text{con probabilità} \qquad \frac{1}{2}.$$

Non è quindi corretto immaginare che, nello stato $|0\rangle$, il qubit possieda semplicemente una coppia di valori classici già determinati

$$(X, Z) = (x, z)$$

che la misura si limita a leggere.
Si vede inoltre l'importanza dell'ordine delle misure. Se si misura prima X, lo stato viene proiettato in $|+\rangle$ oppure in $|-\rangle$. Nessuno dei due è un autostato di Z, per cui una successiva misura di Z torna a essere probabilistica. In questo senso, per osservabili incompatibili, la prima misura modifica lo stato rilevante per la seconda.

Il prodotto tensoriale e l'entanglement sono stati sviluppati nel capitolo precedente. Li useremo qui come strumenti già acquisiti, senza ripeterne la definizione e le dimostrazioni.

5.6 *No-cloning theorem*: impossibilità di clonare uno stato arbitrario

Nel calcolo classico copiare informazione è un'operazione elementare. Se un bit ha valore 0 oppure 1, possiamo leggerlo e produrre quante copie vogliamo dello stesso valore. Per uno stato quantistico arbitrario la situazione è diversa: non esiste un dispositivo fisico universale che, ricevendo in ingresso uno stato sconosciuto, ne produca sempre una copia perfetta senza modificare l'originale.

È importante precisare il significato dell'affermazione. Il *no-cloning theorem* non dice che sia impossibile preparare più sistemi nello stesso stato quando lo stato è noto. Se conosciamo, per esempio, come preparare $|+\rangle$, possiamo eseguire più volte la stessa procedura di preparazione. Il teorema afferma invece che non esiste un'unica trasformazione fisica capace di prendere come ingresso un *qualsiasi* stato sconosciuto $|\psi\rangle$ e produrre automaticamente due copie perfette di quello stesso stato.

Supponiamo di avere due sistemi. Il primo contiene lo stato sconosciuto $|\psi\rangle$; il secondo è inizialmente preparato in uno stato fissato $|s\rangle$, che svolge il ruolo di "registro vuoto". Un clonatore universale dovrebbe realizzare la trasformazione

$$|\psi\rangle|s\rangle \longrightarrow |\psi\rangle|\psi\rangle$$

per ogni possibile stato $|\psi\rangle$.

Per un sistema chiuso una trasformazione fisicamente ammessa deve essere unitaria e quindi, in particolare, lineare. Indichiamo con U l'ipotetico operatore di clonazione. Se U fosse davvero universale, dovrebbe clonare anche gli stati della base computazionale:

$$U|0\rangle|s\rangle = |0\rangle|0\rangle, \qquad U|1\rangle|s\rangle = |1\rangle|1\rangle.$$

Consideriamo ora lo stato

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

Poiché U è lineare,

$$\begin{aligned} U|+\rangle|s\rangle &= U\left[\frac{|0\rangle + |1\rangle}{\sqrt{2}}|s\rangle\right] \\ &= \frac{1}{\sqrt{2}}\left(U|0\rangle|s\rangle + U|1\rangle|s\rangle\right) \\ &= \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right). \end{aligned}$$

Questo è il risultato imposto dalla linearità dell'evoluzione quantistica.

Se però U fosse un clonatore perfetto, applicato a $|+\rangle$ dovrebbe produrre due copie dello stato:

$$U|+\rangle|s\rangle = |+\rangle|+\rangle.$$

Calcoliamo esplicitamente questo prodotto:

$$\begin{aligned} |+\rangle|+\rangle &= \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \\ &= \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right). \end{aligned}$$

I due risultati sono manifestamente diversi:

$$\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \neq \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right).$$

Nel primo stato mancano completamente i termini $|01\rangle$ e $|10\rangle$. Quindi la stessa trasformazione U non può contemporaneamente clonare $|0\rangle$, $|1\rangle$ e la loro sovrapposizione $|+\rangle$.

La contraddizione non dipende dalla particolare scelta di $|+\rangle$: è una conseguenza generale della linearità della meccanica quantistica.

Dimostrazione generale mediante il prodotto interno

Supponiamo che lo stesso operatore unitario U riesca a clonare due stati normalizzati arbitrari $|\psi\rangle$ e $|\phi\rangle$:

$$U|\psi\rangle|s\rangle = |\psi\rangle|\psi\rangle,$$
$$U|\phi\rangle|s\rangle = |\phi\rangle|\phi\rangle.$$

Poiché un operatore unitario conserva il prodotto interno, consideriamo il modulo del prodotto

interno fra i due possibili ingressi. Per gli stati iniziali si ha

$$|(\langle\psi|\langle s|)(|\phi\rangle|s\rangle)| = |\langle\psi \mid \phi\rangle|\,|\langle s \mid s\rangle| = |\langle\psi \mid \phi\rangle|,$$

poiché $\langle s \mid s\rangle = 1$.

Se la clonazione è perfetta, gli stati finali rappresentano due copie degli stati iniziali, eventualmente moltiplicate per fasi globali che non hanno significato fisico. Il modulo del loro prodotto interno è quindi

$$|\langle\psi \mid \phi\rangle|^2.$$

La conservazione del prodotto interno impone allora

$$|\langle\psi \mid \phi\rangle| = |\langle\psi \mid \phi\rangle|^2.$$

Ponendo

$$r = |\langle\psi \mid \phi\rangle|, \qquad 0 \leq r \leq 1,$$

si ottiene

$$r = r^2,$$

e quindi

$$\boxed{r = 0 \qquad \text{oppure} \qquad r = 1.}$$

Il caso $r = 0$ corrisponde a stati ortogonali. Il caso $r = 1$, per stati normalizzati, corrisponde a vettori che differiscono al più per una fase globale e rappresentano quindi lo stesso stato fisico. Due stati fisicamente distinti e non ortogonali non possono dunque essere entrambi clonati perfettamente dallo stesso dispositivo universale.

Questo è il contenuto essenziale del *no-cloning theorem*:

> non esiste un clonatore quantistico perfetto e universale per stati arbitrari sconosciuti.

Il risultato non impedisce di copiare stati appartenenti a un insieme ortogonale noto: in quel caso essi possono essere distinti perfettamente e l'informazione che li identifica può essere trasferita. Non impedisce neppure la correzione quantistica degli errori, che non consiste nel creare copie indipendenti di uno stato sconosciuto, ma nel codificare l'informazione quantistica in modo ridondante all'interno di uno spazio di Hilbert più grande.

Per il calcolo quantistico la conseguenza è importante: non possiamo proteggere un qubit sconosciuto semplicemente producendone molte copie, come faremmo con un bit classico. La protezione dell'informazione quantistica richiede quindi strategie specifiche, come la codifica e la correzione quantistica degli errori.

5.7 Stati non ortogonali e distinguibilità

Due stati ortogonali possono essere distinti perfettamente mediante una misura in una base che li contiene. Per stati non ortogonali la situazione è diversa: se

$$\langle\psi \mid \phi\rangle \neq 0,$$

non esiste una misura che, disponendo di una sola copia e senza informazioni aggiuntive, identifichi sempre con certezza quale dei due stati sia stato preparato.

Consideriamo, per esempio,

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

Essi non sono ortogonali, perché

$$\langle 0 \mid + \rangle = \frac{1}{\sqrt{2}} \neq 0.$$

Se misuriamo nella base computazionale, il risultato 1 identifica con certezza $|+\rangle$, poiché $|0\rangle$ non può produrlo. Il risultato 0, invece, è compatibile con entrambi gli stati: si ottiene con probabilità uno se lo stato era $|0\rangle$ e con probabilità $1/2$ se lo stato era $|+\rangle$. Quella misura non permette quindi una distinzione perfetta. Il fatto che nessun'altra misura possa riuscirvi in modo infallibile è una proprietà generale degli stati non ortogonali.

La non distinguibilità perfetta e il *no-cloning theorem* esprimono due aspetti della stessa struttura dell'informazione quantistica: uno stato sconosciuto non può essere trattato come un dato classico che possa essere prima identificato con certezza e poi copiato arbitrariamente.

5.8 Stati misti e matrice densità

La descrizione mediante un singolo ket è sufficiente quando il sistema si trova in uno stato puro ben definito. Non è invece sufficiente quando la preparazione contiene incertezza classica. Per esempio, una sorgente può preparare $|0\rangle$ con probabilità $1/2$ e $|1\rangle$ con probabilità $1/2$. Questa situazione non coincide con lo stato puro

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

entrambi forniscono 0 e 1 con probabilità $1/2$ nella base computazionale, ma $|+\rangle$ possiede una coerenza di fase che può essere rivelata mediante misure in altre basi. Occorre quindi un formalismo capace di descrivere sia stati puri sia miscele statistiche.

Se il sistema viene preparato nello stato $|\psi_i\rangle$ con probabilità p_i, con

$$\sum_i p_i = 1,$$

si introduce la matrice densità

$$\boxed{\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|.}$$

Per uno stato puro,

$$\rho = |\psi\rangle\langle\psi|.$$

Una matrice densità soddisfa

$$\rho^\dagger = \rho, \qquad \rho \geq 0, \qquad \mathrm{Tr}(\rho) = 1.$$

Per uno stato puro vale

$$\rho^2 = \rho, \qquad \mathrm{Tr}(\rho^2) = 1,$$

mentre per uno stato misto non puro

$$\mathrm{Tr}(\rho^2) < 1.$$

Più in generale, in uno spazio di Hilbert di dimensione d,

$$\frac{1}{d} \leq \mathrm{Tr}(\rho^2) \leq 1.$$

La quantità $\mathrm{Tr}(\rho^2)$ è detta *purezza*: vale 1 per uno stato puro e raggiunge il valore minimo $1/d$ per lo stato massimamente misto $\rho = \mathbb{I}/d$.

La decomposizione in ensemble non è unica

La scrittura

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$$

non identifica in modo univoco un particolare insieme statistico di preparazioni. Ensemble differenti possono descrivere esattamente lo stesso stato fisico. Per un qubit, per esempio,

$$\begin{aligned} \frac{\mathbb{I}}{2} &= \frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1| \\ &= \frac{1}{2}|+\rangle\langle +| + \frac{1}{2}|-\rangle\langle -|. \end{aligned}$$

Le due procedure di preparazione sono diverse, ma producono la stessa matrice densità e quindi le stesse statistiche per qualunque misura eseguita soltanto sul qubit. L'oggetto che rappresenta lo stato fisico è dunque ρ, non una sua particolare decomposizione in termini di probabilità p_i e vettori $|\psi_i\rangle$.

Dalla sfera di Bloch alla palla di Bloch

Per un singolo qubit ogni matrice densità può essere scritta nella forma

$$\boxed{\rho = \frac{1}{2}\left(\mathbb{I} + \boldsymbol{r}\cdot\boldsymbol{\sigma}\right)}, \qquad \boldsymbol{\sigma} = (X, Y, Z), \qquad |\boldsymbol{r}| \leq 1,$$

dove $\boldsymbol{r} = (r_x, r_y, r_z)$ è il vettore di Bloch. La rappresentazione geometrica introdotta per gli stati puri si estende così naturalmente dagli stati sulla superficie della sfera all'intero volume della *palla di Bloch* (*Bloch ball*):

- $|\boldsymbol{r}| = 1$: lo stato è puro e si trova sulla superficie della sfera di Bloch;
- $0 < |\boldsymbol{r}| < 1$: lo stato è misto e corrisponde a un punto interno alla palla;
- $\boldsymbol{r} = 0$: $\rho = \mathbb{I}/2$, lo stato massimamente misto, occupa il centro.

Nel caso di un qubit la purezza assume inoltre la forma

$$\mathrm{Tr}(\rho^2) = \frac{1 + |\boldsymbol{r}|^2}{2},$$

che rende immediato il legame tra posizione geometrica nella palla di Bloch e grado di purezza dello stato.

Una miscela equiprobabile di $|0\rangle$ e $|1\rangle$ ha

$$\begin{aligned}\rho_{\text{mix}} &= \frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1| \\ &= \frac{1}{2}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}.\end{aligned}$$

Per lo stato puro

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

si ha

$$\begin{aligned}\rho_{+} &= |+\rangle\langle +| \\ &= \frac{1}{2}(|0\rangle + |1\rangle)(\langle 0| + \langle 1|) \\ &= \frac{1}{2}\begin{pmatrix}1 & 1\\ 1 & 1\end{pmatrix}.\end{aligned}$$

Le componenti fuori diagonale distinguono la sovrapposizione coerente dalla miscela classica.

5.9 Sottosistemi e traccia parziale

Per un sistema composto descritto da ρ_{AB}, lo stato del solo sistema A è

$$\boxed{\rho_A = \text{Tr}_B(\rho_{AB}).}$$

Consideriamo

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

La matrice densità globale è

$$\rho_{AB} = |\Phi^+\rangle\langle\Phi^+|$$

e si sviluppa come

$$\rho_{AB} = \frac{1}{2}(|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|).$$

Tracciando sul secondo sistema,

$$\begin{aligned}\text{Tr}_B(|00\rangle\langle 00|) &= |0\rangle\langle 0|, \\ \text{Tr}_B(|00\rangle\langle 11|) &= 0, \\ \text{Tr}_B(|11\rangle\langle 00|) &= 0, \\ \text{Tr}_B(|11\rangle\langle 11|) &= |1\rangle\langle 1|.\end{aligned}$$

Quindi

$$\boxed{\rho_A = \frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1| = \frac{1}{2}\mathbb{I}.}$$

Il sistema globale è puro, mentre ciascun sottosistema è misto.

5.10 Evoluzione delle matrici densità

Se

$$\rho = |\psi\rangle\langle\psi|$$

e

$$|\psi'\rangle = U|\psi\rangle,$$

allora

$$\begin{aligned}\rho' &= |\psi'\rangle\langle\psi'| \\ &= U|\psi\rangle\langle\psi|U^\dagger \\ &= \boxed{U\rho U^\dagger}.\end{aligned}$$

Questa formula vale anche per stati misti soggetti a evoluzione unitaria.

5.11 Decoerenza

La decoerenza descrive la perdita delle relazioni di fase osservabili quando il sistema interagisce con gradi di libertà non controllati. Senza introdurre qui la teoria generale dei sistemi aperti, possiamo rappresentare fenomenologicamente un semplice processo di *dephasing* come attenuazione progressiva degli elementi fuori diagonale della matrice densità.

Per uno stato puro bidimensionale,

$$\rho = \begin{pmatrix} |\alpha|^2 & \alpha\beta^* \\ \alpha^*\beta & |\beta|^2 \end{pmatrix}.$$

Un processo di *dephasing* può ridurre le componenti fuori diagonale:

$$\rho \longrightarrow \begin{pmatrix} |\alpha|^2 & \lambda\alpha\beta^* \\ \lambda\alpha^*\beta & |\beta|^2 \end{pmatrix}, \qquad 0 \le \lambda \le 1.$$

Nel limite $\lambda \to 0$,

$$\rho \longrightarrow \begin{pmatrix} |\alpha|^2 & 0 \\ 0 & |\beta|^2 \end{pmatrix}.$$

L'informazione relativa alla coerenza di fase non è più accessibile dal sottosistema considerato. Per il calcolo quantistico questa perdita è critica, perché gli algoritmi sfruttano interferenza e fase relativa.

5.12 Sintesi dei principi

Principio	**Forma matematica essenziale**
Stato puro	$\|\psi\rangle$ con $\langle\psi \mid \psi\rangle = 1$
Fase globale	$\|\psi\rangle \sim e^{i\gamma}\|\psi\rangle$
Sovrapposizione	$a\|\phi\rangle + b\|\chi\rangle$
Evoluzione	$\|\psi'\rangle = U\|\psi\rangle$, $U^\dagger U = \mathbb{I}$
Misura proiettiva	$p(m) = \langle\psi\|P_m\|\psi\rangle$
Post-misura	$\|\psi_m\rangle = \dfrac{P_m\|\psi\rangle}{\sqrt{p(m)}}$
Osservabile	$A = A^\dagger$
Valore medio	$\langle A\rangle = \langle\psi\|A\|\psi\rangle$
Sistema composto	$\mathcal{H}_{AB} = \mathcal{H}_A \otimes \mathcal{H}_B$
Stato misto	$\rho = \sum_i p_i \|\psi_i\rangle\langle\psi_i\|$
Evoluzione densità	$\rho' = U\rho U^\dagger$
Stato ridotto	$\rho_A = \mathrm{Tr}_B(\rho_{AB})$

5.13 Conclusioni

Il modello di calcolo quantistico nasce direttamente dalla struttura della meccanica quantistica. Gli stati sono vettori complessi normalizzati, ma solo il raggio associato al vettore ha significato fisico. Le trasformazioni di sistemi chiusi sono unitarie e quindi reversibili. Le misure sono probabilistiche e trasformano lo stato secondo regole differenti da quelle dell'evoluzione unitaria. I sistemi composti sono descritti mediante prodotti tensoriali e possono possedere stati entangled. Gli stati arbitrari sconosciuti non possono essere clonati. La descrizione mediante matrici densità permette infine di trattare stati misti, sottosistemi e perdita di coerenza.

Questi principi costituiscono il nucleo fisico-matematico su cui vengono costruiti circuiti, algoritmi, protocolli di comunicazione e schemi di correzione degli errori.

Capitolo 6

Il modello della computazione quantistica

Nei capitoli precedenti sono stati introdotti i qubit, la notazione di Dirac, le trasformazioni unitarie, la misura, i sistemi composti e l'entanglement. Possiamo ora porre la domanda che interessa direttamente la computazione: *come si organizza una sequenza di operazioni fisiche sui qubit in modo da realizzare un calcolo?*

Per rispondere è utile procedere in parallelo con il caso classico. Un computer classico e un computer quantistico condividono infatti una struttura astratta simile: entrambi possiedono registri che codificano informazione, operazioni elementari che trasformano tali registri e circuiti ottenuti componendo le operazioni. La differenza decisiva non consiste quindi semplicemente nel sostituire i bit con i qubit. Cambiano la natura dello stato, le trasformazioni ammissibili e, soprattutto, il modo in cui l'informazione può essere codificata nelle ampiezze e nelle fasi e successivamente resa osservabile mediante interferenza e misura.

Lo scopo di questo capitolo è costruire con gradualità il modello a circuiti, mettendo continuamente a confronto computazione classica e computazione quantistica. Gli algoritmi quantistici veri e propri saranno introdotti nel capitolo successivo; qui prepariamo gli strumenti necessari per comprenderli. Per il modello a circuiti e per la terminologia adottata si vedano, tra gli altri, [1, 2, 8, 3].

6.1 Registri classici e registri quantistici

Il registro classico

Un *bit* può assumere uno dei due valori

$$0, \qquad 1.$$

Un registro classico formato da n bit si presenta quindi, in ogni istante, come una stringa

$$x = x_{n-1}x_{n-2}\cdots x_1x_0, \qquad x_j \in \{0,1\}.$$

Le possibili configurazioni sono 2^n. Se interpretiamo la stringa come la rappresentazione binaria di un intero, possiamo scrivere

$$x = \sum_{j=0}^{n-1} 2^j x_j, \qquad 0 \le x \le 2^n - 1.$$

Per esempio, per $n = 3$ i possibili contenuti del registro sono

$$000, 001, 010, 011, 100, 101, 110, 111,$$

che corrispondono agli interi da 0 a 7.

Quando si parla di un registro classico deterministico, si intende che in un certo istante esso contiene *una* di queste configurazioni. Naturalmente un calcolo classico può anche essere probabilistico: in tal caso possiamo associare una probabilità p_x alle diverse configurazioni, con

$$p_x \ge 0, \qquad \sum_x p_x = 1.$$

Questa osservazione sarà utile più avanti, perché impedisce di confondere una sovrapposizione quantistica con una semplice incertezza statistica classica.

Il registro quantistico

A ogni bit sostituiamo ora un qubit. Un registro formato da n qubit è descritto dallo spazio di Hilbert

$$\mathcal{H}_n = (\mathbb{C}^2)^{\otimes n},$$

la cui dimensione è

$$\dim \mathcal{H}_n = 2^n.$$

Gli stati

$$|00\cdots0\rangle,\ |00\cdots1\rangle,\ \ldots,\ |11\cdots1\rangle$$

formano la base computazionale. Essi sono in corrispondenza uno a uno con le 2^n configurazioni di un registro classico. Per abbreviare la notazione, indicheremo frequentemente tali stati con

$$|x\rangle, \qquad x \in \{0, 1, \ldots, 2^n - 1\}.$$

La differenza fondamentale è che lo stato puro generale del registro quantistico non deve essere necessariamente uno stato della base. Esso può essere una sovrapposizione

$$\boxed{|\psi\rangle = \sum_{x=0}^{2^n-1} \alpha_x |x\rangle}$$

con coefficienti complessi $\alpha_x \in \mathbb{C}$ soggetti alla normalizzazione

$$\sum_{x=0}^{2^n-1} |\alpha_x|^2 = 1.$$

Se il registro viene misurato nella base computazionale, il risultato x si ottiene con probabilità

$$p(x) = |\alpha_x|^2.$$

Una singola misura restituisce quindi una singola stringa di n bit, non l'intera lista dei coefficienti α_x.

Esempio 6.1.1 (Un registro di due qubit)**.** Consideriamo lo stato

$$|\psi\rangle = \frac{1}{2}|00\rangle + \frac{\mathrm{i}}{2}|01\rangle + \frac{1}{\sqrt{2}}|11\rangle.$$

La componente $|10\rangle$ è assente e ha ampiezza nulla. Verifichiamo la normalizzazione:

$$\left|\frac{1}{2}\right|^2 + \left|\frac{\mathrm{i}}{2}\right|^2 + \left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{4} + \frac{1}{4} + \frac{1}{2} = 1.$$

Una misura nella base computazionale produce pertanto

$$p(00) = \frac{1}{4}, \qquad p(01) = \frac{1}{4}, \qquad p(10) = 0, \qquad p(11) = \frac{1}{2}.$$

Il risultato più probabile è 11, ma anche in questo caso una singola misura non rivela le tre ampiezze presenti nello stato: restituisce soltanto uno fra gli esiti possibili.

Sovrapposizione e distribuzione classica non sono la stessa cosa

A prima vista si potrebbe pensare che le quantità $|\alpha_x|^2$ svolgano semplicemente lo stesso ruolo delle probabilità p_x di un registro classico casuale. Questa identificazione è però incompleta. Le ampiezze quantistiche sono numeri complessi e possiedono anche una fase. Stati con le stesse probabilità nella base computazionale possono quindi essere fisicamente distinti.

Esempio 6.1.2 (Stesse probabilità, fasi diverse)**.** Consideriamo i due stati

$$|\psi_+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}, \qquad |\psi_-\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}.$$

Misurandoli immediatamente nella base computazionale si ottiene, in entrambi i casi,

$$p(00) = \frac{1}{2}, \qquad p(11) = \frac{1}{2}.$$

Da questa sola misura sembrerebbero indistinguibili. Tuttavia

$$\langle \psi_+ \mid \psi_- \rangle = \frac{1}{2}(1 - 1) = 0,$$

quindi i due stati sono addirittura ortogonali. La differenza è contenuta nel segno relativo, cioè nella fase relativa fra le componenti. Un opportuno circuito può trasformare questa informazione di fase in una differenza osservabile nelle probabilità finali.

Osservazione 6.1.3. La rappresentazione di uno stato di n qubit utilizza 2^n ampiezze complesse. Da questo fatto non segue che il registro quantistico contenga 2^n numeri classici direttamente accessibili. La misura restituisce un esito secondo la regola di Born e non permette di "leggere"

simultaneamente tutte le ampiezze. Il vantaggio computazionale, quando esiste, deve essere ottenuto manipolando coerentemente tali ampiezze prima della misura.

Possiamo riassumere il primo confronto nel seguente schema.

Registro classico	**Registro quantistico**
n bit	n qubit
una configurazione x in un calcolo deterministico	uno stato $\|\psi\rangle$, in generale sovrapposto
2^n configurazioni possibili	spazio di Hilbert di dimensione 2^n
probabilità classiche $p_x \geq 0$ in un calcolo casuale	ampiezze complesse α_x
lettura ideale del contenuto	misura probabilistica secondo $\|\alpha_x\|^2$
nessuna fase associata a una probabilità	fasi relative fisicamente significative

6.2 Circuiti classici e circuiti quantistici

Il circuito classico come composizione di funzioni

Un circuito classico riceve una stringa di bit in ingresso e produce una stringa di bit in uscita. In forma astratta possiamo rappresentarlo come una funzione

$$f : \{0,1\}^n \longrightarrow \{0,1\}^m, \qquad x \longmapsto f(x).$$

Un circuito complesso viene costruito componendo operazioni elementari. Se una prima trasformazione f produce $y = f(x)$ e una seconda trasformazione g produce $z = g(y)$, allora

$$x \xrightarrow{f} y \xrightarrow{g} z$$

e complessivamente

$$z = (g \circ f)(x) = g(f(x)).$$

Le porte logiche classiche, come NOT, AND, OR e XOR, forniscono i mattoni necessari per costruire tali funzioni. Non tutte sono reversibili. Per esempio, la porta AND soddisfa

$$0 \wedge 0 = 0, \qquad 0 \wedge 1 = 0, \qquad 1 \wedge 0 = 0, \qquad 1 \wedge 1 = 1.$$

Dal solo risultato 0 non è possibile ricostruire quale dei primi tre ingressi sia stato applicato: l'operazione ha perso informazione.

Il circuito quantistico come composizione di operatori unitari

Nel modello quantistico ideale, prima della misura, lo stato evolve mediante operatori unitari. Una computazione elementare assume quindi la forma

$$\boxed{|\psi_{\text{in}}\rangle \longrightarrow U \longrightarrow |\psi_{\text{out}}\rangle \longrightarrow \text{misura}}$$

con

$$|\psi_{\text{out}}\rangle = U|\psi_{\text{in}}\rangle.$$

Se il circuito contiene una sequenza di porte $U_1, U_2, \ldots, U_m$, allora

$$|\psi_0\rangle \xrightarrow{U_1} |\psi_1\rangle \xrightarrow{U_2} \cdots \xrightarrow{U_m} |\psi_m\rangle,$$

dove

$$|\psi_1\rangle = U_1|\psi_0\rangle,$$
$$|\psi_2\rangle = U_2|\psi_1\rangle = U_2U_1|\psi_0\rangle,$$

e, in generale,

$$\boxed{|\psi_m\rangle = U_mU_{m-1}\cdots U_2U_1|\psi_0\rangle.}$$

La porta applicata per prima compare quindi più a destra nel prodotto matriciale. Questo fatto è semplicemente la traduzione algebrica della composizione di trasformazioni.

Poiché ogni U_j è unitario,

$$U_j^{-1} = U_j^\dagger.$$

La parte unitaria del circuito è pertanto reversibile. La misura finale, al contrario, produce un risultato classico probabilistico e non è descritta da una trasformazione unitaria sul solo sistema misurato.

Esempio 6.2.1 (Composizione di due trasformazioni unitarie)**.** Per rendere concreta la regola di composizione senza introdurre ancora porte quantistiche specifiche, consideriamo il qubit iniziale

$$|\psi_0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$$

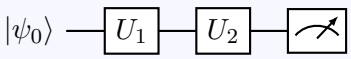
e due trasformazioni definite esplicitamente dalle matrici

$$U_1 = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}, \qquad U_2 = \begin{pmatrix}1 & 0\\ 0 & \mathrm{i}\end{pmatrix}.$$

Entrambe sono unitarie, poiché

$$U_1^\dagger U_1 = I, \qquad U_2^\dagger U_2 = I.$$

Supponiamo di applicare prima U_1 e poi U_2. Il circuito è



Dopo la prima trasformazione,

$$\begin{aligned}|\psi_1\rangle &= U_1|\psi_0\rangle \\ &= \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} \\ &= \frac{1}{2}\begin{pmatrix}2\\0\end{pmatrix} = \begin{pmatrix}1\\0\end{pmatrix} = |0\rangle.\end{aligned}$$

La seconda trasformazione dà quindi

$$|\psi_2\rangle = U_2|\psi_1\rangle = \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{i} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle.$$

La misura nella base computazionale restituisce pertanto 0 con probabilità uno.
L'operatore complessivo è

$$U = U_2 U_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ \mathrm{i} & -\mathrm{i} \end{pmatrix},$$

e infatti

$$U|\psi_0\rangle = |0\rangle.$$

Questo esempio permette anche di verificare che l'ordine delle trasformazioni è essenziale. Se applichiamo prima U_2 e poi U_1, otteniamo

$$U_2|\psi_0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ \mathrm{i} \end{pmatrix},$$

e successivamente

$$\begin{aligned} U_1 U_2|\psi_0\rangle &= \frac{1}{2} \begin{pmatrix} 1+\mathrm{i} \\ 1-\mathrm{i} \end{pmatrix} \\ &= \frac{1+\mathrm{i}}{2}|0\rangle + \frac{1-\mathrm{i}}{2}|1\rangle. \end{aligned}$$

Le probabilità dei due esiti sono ora

$$p(0) = \left|\frac{1+\mathrm{i}}{2}\right|^2 = \frac{1}{2}, \qquad p(1) = \left|\frac{1-\mathrm{i}}{2}\right|^2 = \frac{1}{2}.$$

Dunque le due successioni U_2U_1 e U_1U_2 non descrivono, in generale, la stessa computazione. La composizione degli operatori deve rispettare esattamente l'ordine temporale delle operazioni nel circuito.

Diagrammi circuitali

Nei diagrammi quantistici le linee orizzontali rappresentano i qubit e il tempo viene letto convenzionalmente da sinistra verso destra. Una porta collocata su una linea agisce sul qubit corrispondente; porte collegate fra linee diverse rappresentano operazioni a più qubit. La notazione grafica non è un elemento puramente illustrativo: specifica l'ordine e la struttura della composizione degli operatori.

Il parallelo fra i due modelli può essere sintetizzato così:

Computazione classica	Computazione quantistica
bit	qubit
registro di bit	registro di qubit
porta logica	porta quantistica
funzione booleana	operatore unitario
composizione di funzioni	prodotto di operatori
circuito logico	circuito quantistico
lettura dell'uscita	misura dell'uscita
porte anche irreversibili	evoluzione chiusa unitaria e reversibile

6.3 Porte classiche e porte quantistiche a un qubit

Una porta quantistica su un singolo qubit è rappresentata da una matrice unitaria 2×2. Se

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix},$$

la porta U produce

$$|\psi'\rangle = U|\psi\rangle.$$

Una proprietà fondamentale della linearità è che l'azione della porta sugli stati della base determina automaticamente la sua azione su qualunque stato. Se

$$U|0\rangle = |u_0\rangle, \qquad U|1\rangle = |u_1\rangle,$$

allora

$$U(\alpha|0\rangle + \beta|1\rangle) = \alpha|u_0\rangle + \beta|u_1\rangle.$$

Questa estensione lineare non ha un analogo diretto nella semplice tavola di verità di una porta classica.

NOT classico e porta X

La porta classica NOT inverte un bit:

$$0 \mapsto 1, \qquad 1 \mapsto 0.$$

La porta quantistica di Pauli X realizza la stessa azione sugli stati della base computazionale:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

$$X|0\rangle = |1\rangle, \qquad X|1\rangle = |0\rangle.$$

Ma la definizione quantistica contiene più informazione. Su uno stato generale

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

si ha

$$X|\psi\rangle = \alpha|1\rangle + \beta|0\rangle = \beta|0\rangle + \alpha|1\rangle.$$

In forma vettoriale,

$$\begin{pmatrix}\alpha\\ \beta\end{pmatrix} \xrightarrow{X} \begin{pmatrix}\beta\\ \alpha\end{pmatrix}.$$

Esempio 6.3.1 (Azione di X su uno stato non equiprobabile)**.** Sia

$$|\psi\rangle = \frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle.$$

Prima della porta le probabilità di misura sono

$$p(0) = \frac{3}{4}, \qquad p(1) = \frac{1}{4}.$$

Dopo l'applicazione di X,

$$X|\psi\rangle = \frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle,$$

e quindi

$$p'(0) = \frac{1}{4}, \qquad p'(1) = \frac{3}{4}.$$

La porta X scambia le due ampiezze e, di conseguenza, scambia le probabilità associate ai due risultati della base computazionale.

La porta Z: un'operazione senza equivalente logico classico diretto

La matrice di Pauli Z è

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

La sua azione sugli stati della base è

$$Z|0\rangle = |0\rangle, \qquad Z|1\rangle = -|1\rangle.$$

Se osservassimo soltanto il valore logico 0 o 1, potremmo essere tentati di dire che Z non modifica nulla: una misura immediata nella base computazionale produce infatti le stesse probabilità prima e dopo la porta. Tuttavia, su una sovrapposizione,

$$Z(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle - \beta|1\rangle,$$

per cui viene modificata la fase relativa fra le due componenti.

Esempio 6.3.2 (Una fase che non modifica immediatamente le probabilità)**.** Consideriamo

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

Prima di applicare Z,

$$p(0) = p(1) = \frac{1}{2}.$$

Dopo la porta,

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

ma una misura immediata nella base computazionale continua a fornire

$$p(0) = p(1) = \frac{1}{2}.$$

La trasformazione non è però irrilevante: applicando successivamente una Hadamard si ottiene

$$H|+\rangle = |0\rangle, \qquad H|-\rangle = |1\rangle.$$

La fase relativa è stata così convertita in una differenza deterministica nel risultato della misura.

La porta di Hadamard

La porta di Hadamard è

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

Agli stati della base associa

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

$$H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |-\rangle.$$

Viceversa,

$$H|+\rangle = |0\rangle, \qquad H|-\rangle = |1\rangle,$$

perché

$$H^2 = I.$$

È quindi riduttivo descrivere H soltanto come la porta che "crea una sovrapposizione". Essa realizza il passaggio fra la base computazionale

$$\{|0\rangle, |1\rangle\}$$

e la base

$$\{|+\rangle, |-\rangle\},$$

e permette perciò di convertire informazione di fase in informazione accessibile mediante una misura nella base computazionale.

Esempio 6.3.3 (Interferenza già visibile nell'azione di H)**.** Riprendiamo lo stato

$$|\psi\rangle = \frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle.$$

Applicando H,

$$H|\psi\rangle = \frac{\sqrt{3}}{2}\frac{|0\rangle + |1\rangle}{\sqrt{2}} + \frac{1}{2}\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$
$$= \frac{\sqrt{3}+1}{2\sqrt{2}}|0\rangle + \frac{\sqrt{3}-1}{2\sqrt{2}}|1\rangle.$$

Le probabilità finali sono

$$p(0) = \frac{(\sqrt{3}+1)^2}{8} = \frac{2+\sqrt{3}}{4} \simeq 0.933,$$

$$p(1) = \frac{(\sqrt{3}-1)^2}{8} = \frac{2-\sqrt{3}}{4} \simeq 0.067.$$

La Hadamard non ha semplicemente "reso casuale" il qubit. Le due ampiezze iniziali si sono sommate nel coefficiente di $|0\rangle$ e sottratte in quello di $|1\rangle$. È già presente il meccanismo dell'interferenza.

Le porte di fase S e T

Due porte di fase frequentemente utilizzate sono

$$S = \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{i} \end{pmatrix}, \qquad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{\mathrm{i}\pi/4} \end{pmatrix}.$$

Esse agiscono secondo

$$S|0\rangle = |0\rangle, \qquad S|1\rangle = \mathrm{i}|1\rangle,$$
$$T|0\rangle = |0\rangle, \qquad T|1\rangle = e^{\mathrm{i}\pi/4}|1\rangle.$$

Su uno stato generale,

$$S(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle + \mathrm{i}\beta|1\rangle,$$
$$T(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle + e^{\mathrm{i}\pi/4}\beta|1\rangle.$$

Inoltre

$$S = T^2, \qquad Z = S^2 = T^4.$$

Queste relazioni mostrano come diverse quantità di fase possano essere costruite componendo porte più elementari.

Rotazioni sulla sfera di Bloch

Le rotazioni forniscono una descrizione continua delle trasformazioni a un qubit. Introduciamo anche la matrice di Pauli

$$Y = \begin{pmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{pmatrix}.$$

Ponendo

$$\sigma_x = X, \qquad \sigma_y = Y, \qquad \sigma_z = Z,$$

si definisce

$$R_k(\theta) = \exp\left(-\frac{\mathrm{i}\theta}{2}\sigma_k\right), \qquad k = x, y, z.$$

Poiché

$$\sigma_k^2 = I,$$

l'esponenziale può essere riscritto come

$$R_k(\theta) = \cos\frac{\theta}{2}\, I - \mathrm{i}\sin\frac{\theta}{2}\,\sigma_k.$$

Si ottengono quindi

$$R_x(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -\mathrm{i}\sin\frac{\theta}{2} \\ -\mathrm{i}\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix},$$

$$R_y(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix},$$

$$R_z(\theta) = \begin{pmatrix} e^{-\mathrm{i}\theta/2} & 0 \\ 0 & e^{\mathrm{i}\theta/2} \end{pmatrix}.$$

Esempio 6.3.4 (Preparare uno stato con R_y)**.** Applichiamo $R_y(\theta)$ a $|0\rangle$:

$$R_y(\theta)|0\rangle = \cos\frac{\theta}{2}|0\rangle + \sin\frac{\theta}{2}|1\rangle.$$

Per $\theta = \pi/3$,

$$R_y\left(\frac{\pi}{3}\right)|0\rangle = \cos\frac{\pi}{6}|0\rangle + \sin\frac{\pi}{6}|1\rangle = \frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle.$$

Una misura nella base computazionale fornisce quindi

$$p(0) = \frac{3}{4}, \qquad p(1) = \frac{1}{4}.$$

L'angolo di rotazione controlla direttamente le probabilità attraverso le funzioni trigonometriche del semiangolo.

Le porte discrete sono strettamente collegate alle rotazioni. Per esempio,

$$R_x(\pi) = -\mathrm{i}X, \qquad R_z(\pi) = -\mathrm{i}Z.$$

I fattori $-\mathrm{i}$ sono fasi globali e non modificano lo stato fisico. Inoltre

$$S = e^{\mathrm{i}\pi/4}R_z\left(\frac{\pi}{2}\right), \qquad T = e^{\mathrm{i}\pi/8}R_z\left(\frac{\pi}{4}\right),$$

sempre a meno di una fase globale.

Una prima tabella operativa

Porta	Azione caratteristica	Ruolo computazionale
X	$\|0\rangle \leftrightarrow \|1\rangle$	inversione del valore logico nella base computazionale
Z	$\|1\rangle \mapsto -\|1\rangle$	modifica della fase relativa
H	$\|0\rangle \leftrightarrow \|+\rangle$, $\|1\rangle \leftrightarrow \|-\rangle$	cambio di base e interferenza
S, T	fase su $\|1\rangle$	controllo discreto della fase
R_x, R_y, R_z	rotazioni parametrizzate	controllo continuo dello stato di un qubit

6.4 Operazioni a più bit e a più qubit

Nei circuiti classici molte operazioni dipendono da più bit. Analogamente, un circuito quantistico utile deve poter realizzare trasformazioni che coinvolgono più qubit. È proprio in questo passaggio che compare una possibilità senza analogo classico diretto: le porte a più qubit possono trasformare stati separabili in stati entangled.

La porta CNOT

La controlled-NOT, o CNOT, agisce su due qubit. Il primo è detto *qubit di controllo*, il secondo *qubit bersaglio*. Sulla base computazionale la trasformazione è

$$\boxed{|a, b\rangle \longmapsto |a, b \oplus a\rangle}$$

con $a, b \in \{0, 1\}$.

La relativa tavola è

ingresso	uscita
$\|00\rangle$	$\|00\rangle$
$\|01\rangle$	$\|01\rangle$
$\|10\rangle$	$\|11\rangle$
$\|11\rangle$	$\|10\rangle$

Nella base ordinata

$$\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$$

la matrice è

$$\mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Sulla base computazionale la CNOT può essere letta come una XOR reversibile: il controllo viene conservato e il bersaglio viene invertito soltanto quando $a = 1$. Tuttavia la linearità estende questa azione a tutte le sovrapposizioni. Per uno stato generale

$$|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle,$$

si ha

$$\mathrm{CNOT}|\psi\rangle = a|00\rangle + b|01\rangle + d|10\rangle + c|11\rangle.$$

Le ampiezze associate a $|10\rangle$ e $|11\rangle$ vengono scambiate.

Esempio 6.4.1 (Dalla sovrapposizione all'entanglement)**.** Prepariamo due qubit nello stato

$$|00\rangle.$$

Applichiamo una Hadamard al primo qubit:

$$(H \otimes I)|00\rangle = \frac{|00\rangle + |10\rangle}{\sqrt{2}}.$$

Lo stato è ancora separabile, perché può essere scritto come

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

Applichiamo ora una CNOT con il primo qubit come controllo:

$$\text{CNOT}\,\frac{|00\rangle + |10\rangle}{\sqrt{2}} = \frac{|00\rangle + |11\rangle}{\sqrt{2}} = |\Phi^+\rangle.$$

Lo stato finale è entangled. La stessa CNOT, applicata invece a $|00\rangle$ senza la precedente Hadamard, lascerebbe il sistema nello stato separabile $|00\rangle$. La porta può quindi generare entanglement, ma non lo fa per ogni ingresso.

Porta controlled-U

La CNOT è un caso particolare di una porta controllata. Data una porta unitaria U, definiamo

$$C(U) = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U.$$

La sua azione è

$$|0\rangle|\psi\rangle \longmapsto |0\rangle|\psi\rangle,$$
$$|1\rangle|\psi\rangle \longmapsto |1\rangle U|\psi\rangle.$$

Per $U = X$ si ottiene la CNOT.

La forma operatoriale permette di verificare immediatamente l'azione su una sovrapposizione del qubit di controllo. Se

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

allora

$$C(U)\,(|\chi\rangle|\psi\rangle) = \alpha|0\rangle|\psi\rangle + \beta|1\rangle U|\psi\rangle.$$

In generale lo stato risultante non è più separabile.

La porta controlled-Z

Un'altra porta importante è

$$CZ = \text{diag}(1, 1, 1, -1),$$

cioè

$$CZ|00\rangle = |00\rangle, \quad CZ|01\rangle = |01\rangle, \quad CZ|10\rangle = |10\rangle, \quad CZ|11\rangle = -|11\rangle.$$

Essa non modifica i valori logici degli stati della base, ma introduce una fase -1 quando entrambi i qubit valgono 1. Anche in questo caso la sola tavola dei valori classici non descriverebbe l'effetto computazionale completo.

CNOT e CZ sono collegate dalla relazione

$$\boxed{\mathrm{CNOT} = (I \otimes H)\, CZ\, (I \otimes H).}$$

La relazione discende da

$$HZH = X :$$

la Hadamard sul bersaglio trasforma un controllo di fase in un controllo di inversione.

SWAP e Toffoli

La porta SWAP scambia due qubit:

$$\mathrm{SWAP}|a, b\rangle = |b, a\rangle.$$

La matrice è

$$\mathrm{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Può essere decomposta in tre CNOT:

$$\mathrm{SWAP}_{12} = \mathrm{CNOT}_{12}\, \mathrm{CNOT}_{21}\, \mathrm{CNOT}_{12}\,.$$

La porta Toffoli, o controlled-controlled-NOT, agisce invece su tre qubit:

$$|a, b, c\rangle \longmapsto |a, b, c \oplus ab\rangle.$$

Il terzo qubit viene invertito soltanto se i primi due valgono entrambi 1. Per esempio,

$$|110\rangle \mapsto |111\rangle, \qquad |111\rangle \mapsto |110\rangle,$$

mentre

$$|100\rangle \mapsto |100\rangle.$$

Sulla base computazionale la Toffoli realizza un'operazione classica reversibile ed è quindi un ponte naturale fra circuiti classici reversibili e circuiti quantistici.

6.5 Composizione delle porte: serie e parallelo

Per leggere correttamente un circuito è essenziale distinguere due modi diversi di combinare le porte: l'applicazione successiva nel tempo e l'applicazione su sottosistemi diversi.

Operazioni successive: prodotto di matrici

Se U_1 agisce per prima e U_2 per seconda,

$$|\psi\rangle \xrightarrow{U_1} U_1|\psi\rangle \xrightarrow{U_2} U_2U_1|\psi\rangle.$$

L'operatore complessivo è quindi

$$U_{\text{tot}} = U_2U_1.$$

L'ordine è importante perché, in generale,

$$U_2U_1 \neq U_1U_2.$$

Esempio 6.5.1 (L'ordine delle porte modifica il risultato)**.** Consideriamo le porte X e H applicate a $|0\rangle$.
Se applichiamo prima X e poi H,

$$HX|0\rangle = H|1\rangle = |-\rangle.$$

Se invertiamo l'ordine,

$$XH|0\rangle = X|+\rangle = |+\rangle.$$

Quindi

$$HX|0\rangle \neq XH|0\rangle.$$

I due circuiti non sono equivalenti. Il fatto che $X|+\rangle = |+\rangle$ mostra inoltre che conoscere l'azione di una porta sui soli stati $|0\rangle$ e $|1\rangle$ non è sufficiente per ragionare intuitivamente su un circuito: occorre seguire lo stato effettivo presente a ogni passo.

Operazioni su qubit distinti: prodotto tensoriale

Se U agisce sul primo qubit e V sul secondo, l'operazione complessiva è

$$U \otimes V.$$

Per stati separabili,

$$(U \otimes V)(|\psi\rangle \otimes |\phi\rangle) = U|\psi\rangle \otimes V|\phi\rangle.$$

Esempio 6.5.2 (Hadamard sul primo qubit e NOT sul secondo)**.** Partiamo da

$$|00\rangle.$$

Applichiamo H al primo qubit e X al secondo:

$$(H \otimes X)|00\rangle = H|0\rangle \otimes X|0\rangle = |+\rangle \otimes |1\rangle.$$

Espandendo,

$$(H \otimes X)|00\rangle = \frac{|01\rangle + |11\rangle}{\sqrt{2}}.$$

Se ora applichiamo una CNOT con il primo qubit come controllo,

$$\text{CNOT}\,\frac{|01\rangle + |11\rangle}{\sqrt{2}} = \frac{|01\rangle + |10\rangle}{\sqrt{2}} = |\Psi^+\rangle.$$

Il circuito trasforma quindi uno stato di base in uno stato di Bell mediante una combinazione di operazioni parallele e successive.

Dimensioni degli operatori

Il prodotto tensoriale consente anche un controllo immediato delle dimensioni. Una porta a un qubit è una matrice 2×2; su un registro di due qubit,

$$H \otimes I$$

è una matrice 4×4. In generale, un operatore che agisce su n qubit è rappresentato da una matrice $2^n \times 2^n$.

Per esempio,

$$H \otimes I = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix}$$

con la convenzione di ordinamento

$$\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}.$$

Applicando questa matrice al vettore

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

si ottiene

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \frac{|00\rangle + |10\rangle}{\sqrt{2}},$$

in accordo con il calcolo effettuato direttamente in notazione di Dirac.

Osservazione 6.5.3. Quando si passa dal formalismo matematico a un ambiente software è necessario controllare la convenzione utilizzata per ordinare i qubit e i bit nella rappresentazione degli stati. In queste dispense useremo sempre, salvo avviso esplicito, l'ordinamento $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ per due qubit e la sua naturale generalizzazione a n qubit.

6.6 Funzioni classiche reversibili e *oracle* quantistici

Un algoritmo quantistico deve spesso valutare una funzione classica. Sorge allora un problema: una funzione classica generica non è reversibile, mentre l'evoluzione quantistica chiusa deve essere

unitaria. Occorre quindi incorporare la funzione in una trasformazione reversibile su uno spazio più grande.

Embedding reversibile di una funzione classica

Sia

$$f : \{0,1\}^n \longrightarrow \{0,1\}^m.$$

Introduciamo un secondo registro di m qubit e definiamo

$$\boxed{U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle.}$$

Il simbolo $\oplus$ indica la XOR bit a bit.

Questa trasformazione conserva x e modifica il secondo registro. È reversibile perché applicandola due volte si ottiene

$$\begin{aligned} U_f^2|x\rangle|y\rangle &= U_f|x\rangle|y \oplus f(x)\rangle \\ &= |x\rangle|y \oplus f(x) \oplus f(x)\rangle \\ &= |x\rangle|y\rangle. \end{aligned}$$

Poiché

$$f(x) \oplus f(x) = 0,$$

segue

$$U_f^2 = I, \qquad U_f^{-1} = U_f.$$

L'azione di U_f è una permutazione degli stati della base computazionale e pertanto si estende a un operatore unitario.

Se il secondo registro parte da $|0\rangle^{\otimes m}$,

$$U_f|x\rangle|0\rangle = |x\rangle|f(x)\rangle.$$

Questa formula ricorda la valutazione classica di f, ma la presenza del registro x rende la trasformazione complessiva reversibile.

Esempio 6.6.1 (*Embedding* reversibile della funzione AND)**.** Consideriamo

$$f(x_1, x_0) = x_1 x_0,$$

cioè la funzione AND di due bit. Introduciamo un terzo bit y e definiamo

$$U_f|x_1x_0\rangle|y\rangle = |x_1x_0\rangle|y \oplus x_1x_0\rangle.$$

Se $y = 0$,

$$\begin{aligned} |00\rangle|0\rangle &\mapsto |00\rangle|0\rangle, \\ |01\rangle|0\rangle &\mapsto |01\rangle|0\rangle, \\ |10\rangle|0\rangle &\mapsto |10\rangle|0\rangle, \\ |11\rangle|0\rangle &\mapsto |11\rangle|1\rangle. \end{aligned}$$

L'informazione sull'ingresso non viene cancellata. Se applichiamo di nuovo la stessa trasformazione, per esempio,

$$|11\rangle|1\rangle \mapsto |11\rangle|1 \oplus 1\rangle = |11\rangle|0\rangle,$$

e recuperiamo lo stato iniziale.

L'*oracle*

In molti algoritmi la trasformazione U_f viene trattata come una "scatola nera" che consente di interrogare la funzione f. In questo contesto si parla di *oracle*. Il circuito può utilizzare l'*oracle* una o più volte senza necessariamente specificare, a quel livello di descrizione, come esso sia decomposto in porte elementari.

È importante distinguere due livelli di analisi. Nella *query complexity* si conta quante volte l'algoritmo interroga l'*oracle*. In un'implementazione completa, invece, anche il costo necessario a realizzare U_f mediante porte elementari deve essere preso in considerazione. Questa distinzione diventerà rilevante quando confronteremo algoritmi classici e quantistici.

Qubit ausiliari, *garbage* e *uncomputation*

L'*embedding* reversibile richiede spesso registri di lavoro aggiuntivi, detti qubit ausiliari o *ancilla*. Supponiamo, per esempio, che una parte del circuito debba calcolare temporaneamente una funzione $g(x)$. Un'unitaria U_g può realizzare

$$|x\rangle|0\rangle \xrightarrow{U_g} |x\rangle|g(x)\rangle.$$

Il secondo registro contiene ora informazione utile per passi successivi, ma non può essere semplicemente "azzerato" al termine del suo impiego. Una mappa

$$|g(x)\rangle \longmapsto |0\rangle$$

per valori differenti di $g(x)$ sarebbe infatti molti-a-uno e quindi non unitaria.

La tecnica standard consiste nel *discomputare* l'informazione temporanea. Dopo che $g(x)$ è stata utilizzata in modo reversibile per modificare un altro registro, si applica il circuito inverso:

$$|x\rangle|g(x)\rangle \xrightarrow{U_g^{-1}} |x\rangle|0\rangle.$$

In forma schematica, con un registro bersaglio iniziale $|y\rangle$,

$$|x\rangle|0\rangle|y\rangle \longrightarrow |x\rangle|g(x)\rangle|y\rangle \longrightarrow |x\rangle|g(x)\rangle|y \oplus h(g(x))\rangle \longrightarrow |x\rangle|0\rangle|y \oplus h(g(x))\rangle.$$

L'ultimo passaggio ripristina l'*ancilla* senza cancellare irreversibilmente informazione. Questo procedimento è chiamato *uncomputation*.

Osservazione 6.6.2. Lasciare qubit di lavoro in uno stato dipendente dall'ingresso produce *garbage*. In un circuito quantistico tale informazione residua può restare entangled con il registro principale e compromettere l'interferenza nei passi successivi. Gli *ancilla* sono quindi una risorsa da conteggiare e, quando possibile, da riportare in uno stato noto prima di proseguire il calcolo.

Dall'*oracle* a bit all'*oracle* di fase

Se $f(x)$ assume un solo bit come valore, cioè

$$f : \{0,1\}^n \to \{0,1\},$$

si può ottenere un'azione particolarmente utile preparando il qubit bersaglio nello stato

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

Poiché

$$X|-\rangle = -|-\rangle,$$

si ha

$$U_f|x\rangle|-\rangle = (-1)^{f(x)}|x\rangle|-\rangle.$$

La funzione non compare più come valore scritto nel secondo registro: compare come fase $(-1)^{f(x)}$ associata al primo registro. Questo fenomeno è un caso di *phase kickback* e sarà analizzato in modo più generale in seguito.

Esempio 6.6.3 (La funzione $f(x) = x$ trasformata in una fase)**.** Consideriamo un solo qubit di ingresso e la funzione

$$f(0) = 0, \qquad f(1) = 1.$$

Prepariamo

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

Applicando l'*oracle*,

$$\begin{aligned} U_f|+\rangle|-\rangle &= \frac{1}{\sqrt{2}}\left[(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle\right]|-\rangle \\ &= \frac{|0\rangle - |1\rangle}{\sqrt{2}}|-\rangle \\ &= |-\rangle|-\rangle. \end{aligned}$$

La valutazione della funzione ha modificato soltanto la fase relativa del primo qubit. Applicando una Hadamard al primo qubit,

$$H|-\rangle = |1\rangle,$$

quella fase diventa un risultato osservabile nella base computazionale.

6.7 Parallelismo classico e "parallelismo quantistico"

La crescita esponenziale della dimensione dello spazio di Hilbert suggerisce una proprietà importante, ma spesso interpretata in modo troppo semplicistico. Un registro quantistico può essere preparato in una sovrapposizione di molti stati della base e una stessa trasformazione lineare agisce simultaneamente su tutte le componenti. Questo fatto viene spesso chiamato *parallelismo quantistico*. Occorre però chiarire esattamente che cosa significa e, soprattutto, che cosa non

significa.

Preparazione della sovrapposizione uniforme

Per un singolo qubit,

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}.$$

Applicando Hadamard indipendenti a n qubit inizialmente in $|0\rangle^{\otimes n}$ si ottiene

$$H^{\otimes n}|0\rangle^{\otimes n} = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)^{\otimes n}.$$

Espandendo il prodotto tensoriale,

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

Tutti gli stati della base compaiono con la stessa ampiezza $1/\sqrt{2^n}$.

Esempio 6.7.1 (Sovrapposizione uniforme di due qubit)**.** Per $n = 2$,

$$\begin{aligned}(H \otimes H)|00\rangle &= \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \\ &= \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right).\end{aligned}$$

Una misura immediata produrrebbe ciascuna stringa con probabilità

$$\left|\frac{1}{2}\right|^2 = \frac{1}{4}.$$

Valutazione coerente di una funzione su una sovrapposizione

Prepariamo il primo registro nella sovrapposizione uniforme e il secondo in $|0\rangle$:

$$\frac{1}{\sqrt{2^n}} \sum_x |x\rangle|0\rangle.$$

Per linearità,

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

Formalmente, una sola applicazione di U_f ha agito su tutte le componenti x presenti nella sovrapposizione.

Esempio 6.7.2 (La funzione AND su tutti gli ingressi a due bit)**.** Riprendiamo

$$f(x_1, x_0) = x_1 x_0.$$

Prepariamo

$$\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)|0\rangle.$$

Dopo U_f,

$$|\Psi\rangle = \frac{1}{2}\left(|00\rangle|0\rangle + |01\rangle|0\rangle + |10\rangle|0\rangle + |11\rangle|1\rangle\right).$$

La funzione è stata valutata coerentemente su tutte e quattro le componenti. Se però misuriamo tutti i qubit, otteniamo soltanto uno dei quattro risultati

$$000, \qquad 010, \qquad 100, \qquad 111$$

con probabilità $1/4$ ciascuno. Non otteniamo l'intera tavola di verità in una singola esecuzione.

Perché il parallelismo da solo non basta

Un computer classico realmente parallelo può calcolare più valori di una funzione utilizzando più unità di elaborazione e può, in linea di principio, memorizzare separatamente i diversi risultati. Nel caso quantistico, invece, i diversi valori sono codificati nelle componenti di un unico stato e non sono tutti direttamente accessibili tramite misura.

La domanda decisiva non è quindi

> "Come possiamo calcolare simultaneamente molti valori di f?"

ma piuttosto

> "Come possiamo far interferire le ampiezze in modo che una proprietà globale di f emerga con alta probabilità in pochi risultati misurabili?"

Questo passaggio dal parallelismo all'interferenza è il punto concettuale che rende il modello quantistico realmente diverso da un semplice calcolo probabilistico classico.

6.8 Fase, interferenza e *phase kickback*

Probabilità classiche e ampiezze quantistiche

In un processo probabilistico classico, se due alternative mutuamente esclusive conducono allo stesso risultato, le loro probabilità si sommano. Le probabilità sono numeri reali non negativi e non possono cancellarsi.

Nel formalismo quantistico si sommano invece le *ampiezze* associate ai cammini indistinguibili; soltanto alla fine si prende il modulo quadro. Se due ampiezze hanno segno o fase opposta, possono cancellarsi. Se hanno la stessa fase, possono rinforzarsi.

Il circuito più semplice che mostra questo meccanismo è costituito da due Hadamard successive.

Esempio 6.8.1 (Interferenza costruttiva e distruttiva con H^2)**.** Partiamo da $|0\rangle$. Dopo la prima Hadamard,

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}.$$

Applichiamo una seconda Hadamard:

$$\begin{aligned} H\frac{|0\rangle + |1\rangle}{\sqrt{2}} &= \frac{1}{\sqrt{2}}\left(H|0\rangle + H|1\rangle\right) \\ &= \frac{1}{2}\left[(|0\rangle + |1\rangle) + (|0\rangle - |1\rangle)\right] \\ &= \frac{1}{2}\left(2|0\rangle + 0|1\rangle\right) \\ &= |0\rangle. \end{aligned}$$

Per l'esito 0, le due ampiezze $1/2$ si sommano:

$$\frac{1}{2} + \frac{1}{2} = 1.$$

Per l'esito 1, invece, si cancellano:

$$\frac{1}{2} - \frac{1}{2} = 0.$$

Quindi

$$p(0) = 1, \qquad p(1) = 0.$$

Se interpretassimo erroneamente la prima Hadamard come una semplice "moneta equa" classica e dimenticassimo la fase, dopo la seconda trasformazione penseremmo di dover sommare probabilità positive e non potremmo spiegare il ritorno deterministico a 0. È proprio la coerenza delle ampiezze a rendere possibile la cancellazione.

Una fase intermedia può cambiare completamente l'uscita

Inseriamo ora una porta Z fra le due Hadamard:

$$|0\rangle \xrightarrow{H} |+\rangle \xrightarrow{Z} |-\rangle \xrightarrow{H} |1\rangle.$$

La Z non modifica le probabilità se misuriamo immediatamente nella base computazionale, ma cambia il segno di una delle ampiezze. La seconda Hadamard fa quindi interferire le componenti in modo opposto rispetto al caso precedente.

Esempio 6.8.2 (Confronto numerico fra HH e HZH)**.** Nel circuito HH, appena prima della seconda Hadamard le ampiezze sono

$$\alpha_0 = \frac{1}{\sqrt{2}}, \qquad \alpha_1 = \frac{1}{\sqrt{2}}.$$

Dopo H,

$$a_0 = \frac{\alpha_0 + \alpha_1}{\sqrt{2}} = 1, \qquad a_1 = \frac{\alpha_0 - \alpha_1}{\sqrt{2}} = 0.$$

Nel circuito HZH, invece,

$$\alpha_0 = \frac{1}{\sqrt{2}}, \qquad \alpha_1 = -\frac{1}{\sqrt{2}}.$$

Dopo l'ultima Hadamard,

$$a_0 = \frac{\alpha_0 + \alpha_1}{\sqrt{2}} = 0, \qquad a_1 = \frac{\alpha_0 - \alpha_1}{\sqrt{2}} = 1.$$

Una sola modifica di fase ha scambiato completamente l'esito deterministico finale da 0 a 1.

Phase kickback da un autovalore unitario

Il *phase kickback* non è limitato agli *oracle* booleani. Consideriamo una controlled-U e supponiamo che il qubit o registro bersaglio sia preparato in un autostato $|u\rangle$ di U:

$$U|u\rangle = e^{\mathrm{i}\phi}|u\rangle.$$

Prepariamo il controllo nello stato

$$\alpha|0\rangle + \beta|1\rangle.$$

Prima della controlled-U lo stato complessivo è

$$(\alpha|0\rangle + \beta|1\rangle)|u\rangle.$$

Dopo la porta,

$$\begin{aligned} C(U)(\alpha|0\rangle + \beta|1\rangle)|u\rangle &= \alpha|0\rangle|u\rangle + \beta|1\rangle U|u\rangle \\ &= \alpha|0\rangle|u\rangle + \beta e^{\mathrm{i}\phi}|1\rangle|u\rangle \\ &= \left(\alpha|0\rangle + \beta e^{\mathrm{i}\phi}|1\rangle\right)|u\rangle. \end{aligned}$$

La fase associata all'autovalore di U è comparsa nel qubit di controllo, mentre il bersaglio resta nello stesso autostato $|u\rangle$. Da qui il termine *kickback*: l'informazione di fase viene trasferita al controllo.

Esempio 6.8.3 (Trasformare una fase in probabilità)**.** Poniamo il controllo inizialmente in

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

Dopo il *phase kickback*,

$$|\chi\rangle = \frac{|0\rangle + e^{\mathrm{i}\phi}|1\rangle}{\sqrt{2}}.$$

Una misura immediata nella base computazionale darebbe ancora

$$p(0) = p(1) = \frac{1}{2},$$

indipendentemente da ϕ. Applichiamo invece una Hadamard:

$$H|\chi\rangle = \frac{1}{2}\left[(1 + e^{\mathrm{i}\phi})|0\rangle + (1 - e^{\mathrm{i}\phi})|1\rangle\right].$$

Le probabilità diventano

$$p(0) = \frac{|1 + e^{i\phi}|^2}{4} = \frac{1 + \cos\phi}{2} = \cos^2\frac{\phi}{2},$$

$$p(1) = \frac{|1 - e^{i\phi}|^2}{4} = \frac{1 - \cos\phi}{2} = \sin^2\frac{\phi}{2}.$$

Per esempio, se

$$\phi = \frac{\pi}{3},$$

allora

$$p(0) = \cos^2\frac{\pi}{6} = \frac{3}{4}, \qquad p(1) = \sin^2\frac{\pi}{6} = \frac{1}{4}.$$

La fase, che non era direttamente visibile in una misura computazionale, è stata convertita in una distribuzione di probabilità misurabile. Questo principio sarà alla base della stima quantistica di fase.

Possiamo quindi anticipare una regola guida per molti algoritmi quantistici:

$$\boxed{\text{sovrapposizione} \longrightarrow \text{codifica in fase} \longrightarrow \text{interferenza} \longrightarrow \text{misura.}}$$

6.9 Misura, *shot* e risultato computazionale

Il circuito quantistico produce alla fine uno stato, mentre l'utente riceve un risultato classico mediante misura. È quindi essenziale distinguere la descrizione teorica dello stato dalla statistica degli esiti osservati.

Un singolo *shot*

Nel linguaggio della computazione quantistica, uno *shot* è una singola esecuzione del circuito comprendente la preparazione iniziale, l'applicazione delle porte e la misura finale. Se lo stato finale è

$$|\psi\rangle = \sum_x \alpha_x |x\rangle,$$

un singolo *shot* restituisce un solo valore x con probabilità

$$p(x) = |\alpha_x|^2.$$

È importante sottolineare che ripetere gli *shot* significa, idealmente, *ripreparare lo stato iniziale e rieseguire il circuito*. Non significa misurare molte volte lo stesso esemplare già collassato dopo la prima misura.

Esempio 6.9.1 (Un qubit con probabilità 70% e 30%)**.** Consideriamo lo stato

$$|\psi\rangle = \sqrt{0.7}|0\rangle + e^{i\phi}\sqrt{0.3}|1\rangle.$$

Una misura nella base computazionale produce

$$p(0) = 0.7, \qquad p(1) = 0.3.$$

La fase ϕ non influenza questa specifica misura. In $N = 1000$ *shot* ci aspettiamo, in media,

$$N_0 = 700, \qquad N_1 = 300.$$

I numeri osservati non saranno però necessariamente esattamente 700 e 300: ogni insieme finito di *shot* presenta fluttuazioni statistiche.

Fluttuazioni statistiche

Per un risultato binario con probabilità p, il numero K di successi in N *shot* segue una distribuzione binomiale. Il valore medio e la deviazione standard sono

$$\mathbb{E}[K] = Np, \qquad \sigma_K = \sqrt{Np(1-p)}.$$

Per la frequenza relativa

$$\hat{p} = \frac{K}{N}$$

si ha

$$\sigma_{\hat{p}} = \sqrt{\frac{p(1-p)}{N}}.$$

Esempio 6.9.2 (Errore statistico su 1000 *shot*)**.** Nel caso precedente, per $p = 0.3$ e $N = 1000$,

$$\sigma_K = \sqrt{1000 \cdot 0.3 \cdot 0.7} = \sqrt{210} \simeq 14.5.$$

La deviazione standard della frequenza è

$$\sigma_{\hat{p}} = \sqrt{\frac{0.3 \cdot 0.7}{1000}} \simeq 0.0145.$$

Una frequenza sperimentale come 0.286 oppure 0.314 sarebbe quindi del tutto compatibile con una probabilità teorica pari a 0.3.
La dipendenza

$$\sigma_{\hat{p}} \propto \frac{1}{\sqrt{N}}$$

mostra inoltre che, per dimezzare l'errore statistico tipico, occorre quadruplicare il numero di *shot*.

Algoritmi deterministici e probabilistici

Un algoritmo quantistico ideale può produrre la risposta corretta con probabilità uno oppure con probabilità minore di uno. Nel secondo caso il circuito deve essere progettato in modo che la risposta desiderata abbia una probabilità sufficientemente elevata e, se necessario, l'esecuzione può essere ripetuta.

Supponiamo, per esempio, che un certo circuito restituisca la risposta corretta con probabilità

$$p_{\text{succ}} = 0.8.$$

Su 100 esecuzioni indipendenti il numero medio di successi è

$$100 \times 0.8 = 80,$$

con deviazione standard

$$\sqrt{100 \times 0.8 \times 0.2} = 4.$$

Questo non significa che ogni algoritmo richieda necessariamente molti *shot*: la quantità di ripetizioni necessaria dipende dalla struttura del problema, dalla probabilità di successo e da quale informazione si vuole stimare.

Misurare in una base diversa

Nel modello circuitale è utile ricondurre le misure in basi diverse a una misura finale nella base computazionale preceduta da un opportuno cambio di base. Per esempio, misurare nella base

$$\{|+\rangle, |-\rangle\}$$

può essere realizzato applicando H e misurando poi nella base

$$\{|0\rangle, |1\rangle\},$$

perché

$$H|+\rangle = |0\rangle, \qquad H|-\rangle = |1\rangle.$$

Questa semplice osservazione è operativamente importante: molte informazioni codificate nelle fasi diventano accessibili soltanto dopo un cambio di base.

Osservazione 6.9.3. Il modello introduttivo utilizzato in questo capitolo colloca la misura alla fine del circuito. Circuiti più generali possono includere misure intermedie, operazioni condizionate dal risultato della misura e *reset* di qubit. Queste estensioni non modificano i principi di base discussi qui, ma saranno considerate soltanto quando necessarie.

6.10 Risorse di un circuito quantistico

Dire che un algoritmo utilizza un circuito quantistico non è sufficiente per valutarne l'efficienza. Come nel calcolo classico, occorre specificare quali risorse vengono consumate. Nel caso quantistico sono particolarmente utili le seguenti grandezze.

Numero di qubit

Il numero di qubit utilizzati è spesso chiamato *width* del circuito. Va distinto fra qubit che contengono l'input, qubit di lavoro o qubit ausiliari (*ancilla*), e qubit eventualmente utilizzati per memorizzare risultati intermedi.

Numero di porte

Il *gate count* conta quante porte elementari vengono applicate. Due circuiti che realizzano lo stesso operatore possono avere costi molto diversi a seconda della decomposizione scelta.

Profondità del circuito

Porte che agiscono su qubit disgiunti possono, nel modello ideale, essere eseguite nello stesso livello logico. La *depth* è il numero di livelli successivi necessari quando le operazioni indipendenti vengono parallelizzate quanto possibile.

Esempio 6.10.1 (*gate count* e *depth* non coincidono)**.** Consideriamo tre qubit. Applichiamo prima una Hadamard a ciascun qubit e poi una CNOT dal primo al secondo e una CNOT dal secondo al terzo.
Le tre Hadamard possono essere eseguite nello stesso livello:

$$H \otimes H \otimes H.$$

Le due CNOT, invece, condividono il secondo qubit e non possono appartenere allo stesso livello del circuito standard. Abbiamo quindi:

$$\text{numero di qubit} = 3,$$

$$\textit{gate count} = 3 + 2 = 5,$$

$$\text{numero di porte a due qubit} = 2,$$

$$\textit{depth} = 3.$$

Il *gate count* misura quante operazioni compaiono complessivamente; la *depth* misura quante fasi temporali sequenziali restano dopo avere sfruttato il parallelismo disponibile.

Porte a più qubit

È spesso utile contare separatamente le porte che coinvolgono due o più qubit. Dal punto di vista logico esse sono le operazioni che permettono di creare correlazioni ed entanglement. Dal punto di vista di un'implementazione fisica, il loro costo può inoltre essere differente da quello delle porte a un qubit; il dettaglio dipende però dall'architettura hardware e non verrà assunto qui come universale.

Numero di *query*

Quando un problema viene formulato mediante un *oracle* U_f, una risorsa centrale è il numero di *query* all'*oracle*. Se un algoritmo classico necessita di N interrogazioni e un algoritmo quantistico ne richiede $O(\sqrt{N})$, si parla di vantaggio nella *query complexity*. Per trasformare questo vantaggio in un vantaggio complessivo occorre tuttavia considerare anche il costo di preparazione degli stati, delle altre porte, della realizzazione dell'*oracle* e della misura.

Numero di *shot*

Gli *shot* costituiscono una risorsa distinta dalla profondità o dal numero di porte. Un circuito può essere molto corto ma dover essere ripetuto molte volte per stimare una probabilità con sufficiente precisione. Viceversa, un algoritmo ideale che produce in modo deterministico un singolo bit corretto può richiedere, in linea di principio, una sola esecuzione finale.

Il costo complessivo di una procedura quantistica deve quindi essere descritto con più parametri, non mediante un unico numero.

Risorsa	**Significato**
qubit / *width*	dimensione fisica-logica del registro utilizzato
gate count	numero complessivo di porte elementari
depth	numero di livelli sequenziali dopo la parallelizzazione
porte a due o più qubit	costo delle operazioni che correlano sottosistemi
query	numero di interrogazioni a un oracle
shot	numero di esecuzioni complete necessarie per ottenere statistiche

6.11 Confronto operativo fra computazione classica e quantistica

A questo punto possiamo mettere in parallelo i due modelli in modo più preciso. Alcune strutture sono direttamente corrispondenti; altre segnano una rottura concettuale.

Le analogie

In entrambi i casi:

- l'informazione è organizzata in registri;
- il calcolo è costruito mediante operazioni elementari;
- le operazioni vengono composte in circuiti;
- un problema complesso viene ridotto a una successione di trasformazioni più semplici;
- l'efficienza richiede di contare risorse come memoria, operazioni e profondità.

Le differenze decisive

Nel modello quantistico:

- lo stato può essere una sovrapposizione coerente di stati della base;
- le ampiezze sono complesse e possiedono fasi relative;
- l'evoluzione chiusa è unitaria e quindi reversibile;
- le porte a più qubit possono generare entanglement;
- le ampiezze possono interferire costruttivamente o distruttivamente;

- la misura produce risultati classici probabilistici e limita l'informazione direttamente accessibile.

Il punto centrale è che nessuno di questi elementi, preso isolatamente, può essere identificato automaticamente con un vantaggio computazionale. Una grande sovrapposizione non è sufficiente; l'entanglement non è sempre necessario in ogni protocollo; una misura probabilistica non rende un calcolo più efficiente. Gli algoritmi quantistici utili combinano questi ingredienti in modo strutturato per concentrare l'ampiezza sui risultati informativi.

Esempio 6.11.1 (Perché "calcolare tutti i valori" non è ancora un algoritmo)**.** Supponiamo di avere $n = 10$ qubit di ingresso. La sovrapposizione uniforme contiene

$$2^{10} = 1024$$

stati della base:

$$\frac{1}{32}\sum_{x=0}^{1023}|x\rangle.$$

Dopo un *oracle*,

$$\frac{1}{32}\sum_{x=0}^{1023}|x\rangle|f(x)\rangle.$$

La funzione ha agito coerentemente su 1024 componenti. Tuttavia una misura restituisce un solo valore x e il corrispondente $f(x)$. Se il nostro obiettivo fosse ricostruire l'intera tabella di 1024 valori, questa preparazione non fornirebbe magicamente tutti i risultati in un solo *shot*.
Per ottenere un vantaggio occorre invece progettare trasformazioni successive che facciano interferire le 1024 componenti in modo che la proprietà cercata – non necessariamente l'intera tabella – venga concentrata in pochi esiti misurabili.

6.12 Errori concettuali frequenti

- **"Un registro di n qubit memorizza 2^n numeri classici."** Non in un senso direttamente accessibile. La descrizione dello stato richiede 2^n ampiezze, ma una misura non restituisce tali ampiezze una per una.
- **"La Hadamard è una porta casuale."** No. H è una trasformazione unitaria deterministica dello stato. La casualità compare nella misura; prima della misura le ampiezze conservano relazioni di fase coerenti.
- **"Una porta Z non fa nulla perché non cambia 0 e 1."** Falso. Modifica la fase relativa e può quindi cambiare completamente il risultato di operazioni successive basate sull'interferenza.
- **"La CNOT crea sempre entanglement."** Falso. Può creare entanglement per opportuni ingressi, ma lascia molti stati separabili, per esempio $|00\rangle$ e $|10\rangle$.
- **"Applicare U_f a una sovrapposizione permette di leggere tutti gli $f(x)$ contemporaneamente."** No. L'*oracle* agisce coerentemente su tutte le componenti, ma la misura finale fornisce soltanto un esito. Il vantaggio nasce dall'interferenza successiva.
- **"Mille *shot* significano misurare mille volte lo stesso qubit."** No. Uno *shot*

corrisponde a una nuova preparazione ed esecuzione del circuito, seguita da una misura.

- **"Il numero di porte è l'unica misura del costo."** No. *width*, *depth*, porte a più qubit, *query* e *shot* descrivono aspetti diversi della complessità di una procedura.

6.13 Sintesi

Il passaggio dal calcolo classico al calcolo quantistico può essere letto come una serie di corrispondenze:

bit	$\longrightarrow$	qubit
registro di bit	$\longrightarrow$	registro di qubit
porta logica	$\longrightarrow$	porta unitaria
circuito logico	$\longrightarrow$	circuito quantistico
funzione classica	$\longrightarrow$	*embedding reversibile/oracle.*

La corrispondenza, tuttavia, non è completa. Il modello quantistico introduce strumenti senza analogo diretto nella computazione classica deterministica: sovrapposizione coerente, fase relativa, entanglement e interferenza. Allo stesso tempo impone un vincolo fondamentale: lo stato non può essere letto arbitrariamente, perché l'uscita viene ottenuta mediante misura.

Il modello operativo studiato in questo capitolo può essere riassunto nella catena

$$\begin{aligned}\text{input classico} &\longrightarrow \text{preparazione} \longrightarrow \text{porte unitarie}\\ &\longrightarrow \text{codifica in ampiezze e fasi} \longrightarrow \text{interferenza}\\ &\longrightarrow \text{misura} \longrightarrow \text{output classico.}\end{aligned}$$

Il compito di un algoritmo quantistico consiste nel progettare la parte centrale di questa catena in modo che l'informazione rilevante, inizialmente distribuita nelle ampiezze dello stato, venga trasformata in pochi risultati classici osservabili con probabilità elevata. Nel capitolo successivo vedremo come questa strategia si concretizza in algoritmi specifici, partendo dai casi più semplici e arrivando progressivamente alle tecniche di amplificazione di ampiezza, trasformata di Fourier quantistica e stima di fase.

Capitolo 7

Rappresentazione grafica dei circuiti quantistici

Nel capitolo precedente abbiamo descritto una computazione quantistica in termini di registri, operatori unitari, composizione di trasformazioni e misura. Per lavorare con circuiti di qualche complessità, tuttavia, la sola notazione matriciale diventa rapidamente poco maneggevole. La rappresentazione grafica dei circuiti quantistici fornisce un linguaggio compatto nel quale qubit, porte, controlli e misure vengono disposti secondo l'ordine in cui agiscono.

Lo scopo di questo capitolo non è introdurre nuove trasformazioni fisiche, ma imparare a *leggere* e *scrivere* un circuito quantistico, passando in modo sistematico dalla rappresentazione grafica alla corrispondente descrizione matematica e viceversa.

7.1 Le porte come elementi grafici

La rappresentazione grafica parte dagli oggetti più semplici: le porte a un qubit.

Un qubit è rappresentato mediante una linea orizzontale, detta comunemente *wire*. Lo stato iniziale viene indicato a sinistra, mentre le operazioni vengono collocate lungo la linea. Per convenzione, il circuito viene letto da sinistra verso destra.

Se un qubit viene preparato nello stato $|\psi\rangle$ e successivamente sottoposto a una trasformazione unitaria U, la rappresentazione è

$$|\psi\rangle \;\text{—}\boxed{U}\text{—}$$

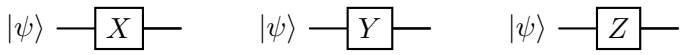
Il rettangolo contenente U rappresenta la porta. Dal punto di vista matematico il diagramma significa semplicemente

$$|\psi\rangle \longmapsto U|\psi\rangle.$$

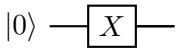
La linea non rappresenta una traiettoria nello spazio fisico: identifica il qubit logico sul quale agiscono, nell'ordine, le diverse operazioni. In un circuito ideale, una porzione di *wire* priva di simboli indica che sul qubit non viene applicata alcuna trasformazione esplicita; equivale quindi all'azione dell'identità.

Porte X, Y e Z

Le tre porte di Pauli vengono rappresentate semplicemente scrivendo il loro simbolo all'interno del rettangolo:



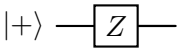
Per esempio,



rappresenta la trasformazione

$$X|0\rangle = |1\rangle.$$

Nel caso della porta Z, il diagramma



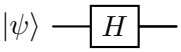
corrisponde invece a

$$Z|+\rangle = |-\rangle.$$

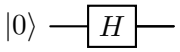
È importante osservare che la stessa forma grafica viene utilizzata sia per porte che modificano direttamente gli stati della base computazionale, come X, sia per porte che agiscono soprattutto sulla fase relativa, come Z. Il significato del simbolo è determinato dall'operatore associato alla porta.

La porta di Hadamard

La porta di Hadamard viene rappresentata come



Per un qubit inizialmente nello stato $|0\rangle$,



significa

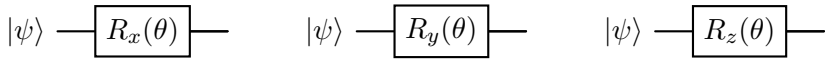
$$|0\rangle \longmapsto H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}.$$

La rappresentazione grafica non sostituisce quindi il calcolo algebrico: lo rende più compatto. Per sapere quale stato produce un circuito è sempre possibile tradurre ciascun simbolo nella corrispondente matrice e applicare gli operatori allo stato iniziale.

Porte di fase e rotazioni

Le porte S e T vengono indicate nello stesso modo:

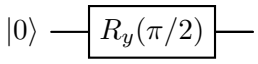
Le rotazioni parametrizzate riportano invece anche il valore dell'angolo:



Per esempio, se

$$\theta = \frac{\pi}{2},$$

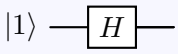
il circuito



produce

$$R_y\left(\frac{\pi}{2}\right)|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = |+\rangle.$$

Esempio 7.1.1 (Dal simbolo grafico al calcolo matriciale)**.** Consideriamo il circuito



Il simbolo grafico ci dice che dobbiamo applicare la matrice

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

al vettore

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Pertanto

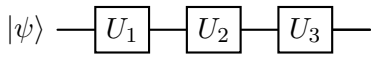
$$H|1\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

Il circuito compatto e il calcolo matriciale descrivono quindi esattamente la stessa trasformazione.

7.2 Più porte sullo stesso qubit

Quando più porte compaiono sulla stessa linea, esse agiscono nell'ordine in cui vengono incontrate procedendo da sinistra verso destra.

Il circuito



rappresenta la successione

$$|\psi\rangle \xrightarrow{U_1} U_1|\psi\rangle \xrightarrow{U_2} U_2U_1|\psi\rangle \xrightarrow{U_3} U_3U_2U_1|\psi\rangle.$$

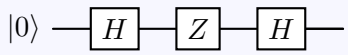
Perciò l'operatore complessivo è

$$U_{\text{tot}} = U_3U_2U_1.$$

Questo punto è essenziale: il circuito si legge graficamente da sinistra verso destra, mentre nel

prodotto matriciale l'operatore che agisce per primo compare più a destra.

Esempio 7.2.1 (Un circuito equivalente a una porta X)**.** Consideriamo



L'operatore complessivo è

$$U = HZH.$$

Calcoliamo esplicitamente lo stato dopo ogni porta:

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

$$|+\rangle \xrightarrow{Z} \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |-\rangle,$$

e infine

$$|-\rangle \xrightarrow{H} |1\rangle.$$

Sul particolare stato iniziale $|0\rangle$ il circuito produce dunque $|1\rangle$. In realtà, moltiplicando le matrici,

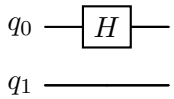
$$HZH = X,$$

per cui il circuito è equivalente alla porta X su qualunque stato di ingresso.

7.3 Circuiti con più qubit

Un circuito con n qubit contiene n *wire* orizzontali. Per esempio, un registro di due qubit viene rappresentato come

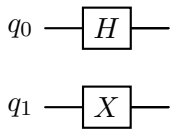
Quando una porta agisce su un solo qubit, il suo simbolo compare soltanto sulla *wire* corrispondente. Per esempio,



significa che si applica H al primo qubit e l'identità al secondo. L'operatore complessivo è quindi

$$H \otimes I.$$

Analogamente,



rappresenta l'operatore

$$H \otimes X.$$

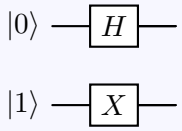
Le porte poste nella stessa colonna agiscono su qubit diversi e possono essere interpretate, nel

modello ideale, come operazioni appartenenti allo stesso livello circuitale.

Esempio 7.3.1 (Due porte in parallelo)**.** Consideriamo lo stato iniziale

$$|01\rangle = |0\rangle \otimes |1\rangle$$

e il circuito



L'operatore è

$$H \otimes X.$$

Poiché

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \qquad X|1\rangle = |0\rangle,$$

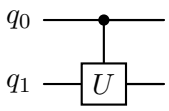
si ottiene

$$\begin{aligned}(H \otimes X)|01\rangle &= H|0\rangle \otimes X|1\rangle \\ &= \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |0\rangle \\ &= \frac{|00\rangle + |10\rangle}{\sqrt{2}}.\end{aligned}$$

7.4 Porte controllate

Una porta controllata coinvolge almeno due qubit. Un qubit determina se una trasformazione debba essere applicata a un altro qubit.

La controlled-U viene rappresentata mediante un punto pieno sul qubit di controllo, collegato verticalmente alla porta U posta sul qubit bersaglio:

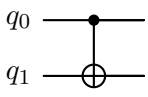


Il significato è

$$|0\rangle|\psi\rangle \longmapsto |0\rangle|\psi\rangle,$$
$$|1\rangle|\psi\rangle \longmapsto |1\rangle U|\psi\rangle.$$

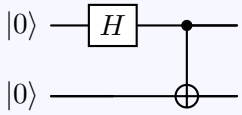
La CNOT

Per la controlled-X, cioè la CNOT, il bersaglio viene tradizionalmente rappresentato con il simbolo $\oplus$:



Il punto pieno identifica il controllo; il simbolo $\oplus$ identifica il bersaglio sul quale agisce X quando il controllo vale 1.

Esempio 7.4.1 (Lettura di un circuito a due qubit)**.** Consideriamo



Dopo la porta H sul primo qubit,

$$|00\rangle \longmapsto (H \otimes I)|00\rangle = \frac{|00\rangle + |10\rangle}{\sqrt{2}}.$$

La CNOT agisce quindi sulle due componenti:

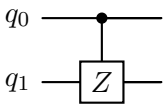
$$|00\rangle \mapsto |00\rangle, \qquad |10\rangle \mapsto |11\rangle.$$

Per linearità,

$$\frac{|00\rangle + |10\rangle}{\sqrt{2}} \longmapsto \frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$

Controlled-Z

Una controlled-Z può essere rappresentata come



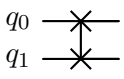
Essa applica una fase -1 soltanto alla componente $|11\rangle$:

$$|00\rangle \mapsto |00\rangle, \qquad |01\rangle \mapsto |01\rangle,$$

$$|10\rangle \mapsto |10\rangle, \qquad |11\rangle \mapsto -|11\rangle.$$

7.5 SWAP e Toffoli

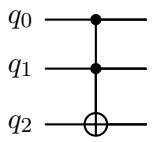
La porta SWAP scambia due qubit. Graficamente si utilizzano due croci collegate:



La trasformazione corrispondente è

$$|a, b\rangle \longmapsto |b, a\rangle.$$

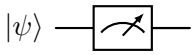
La porta Toffoli, o controlled-controlled-X, contiene invece due controlli e un bersaglio:



Il bersaglio viene invertito soltanto se entrambi i controlli valgono 1.

7.6 Misura e registri classici

La misura costituisce il punto di collegamento tra la parte quantistica e quella classica della computazione. Graficamente una misura viene indicata con il simbolo di uno strumento di misura:



Concettualmente,

$$|\psi\rangle \longrightarrow \text{misura} \longrightarrow c \in \{0,1\}.$$

Se

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

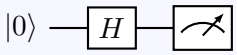
la misura nella base computazionale produce

$$c = 0 \quad \text{con probabilità} \quad |\alpha|^2,$$

e

$$c = 1 \quad \text{con probabilità} \quad |\beta|^2.$$

Esempio 7.6.1 (Circuito con misura finale)**.** Il circuito



prepara

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}.$$

La misura finale produce quindi

$$0 \quad \text{con probabilità} \quad \frac{1}{2},$$

e

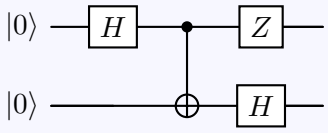
$$1 \quad \text{con probabilità} \quad \frac{1}{2}.$$

7.7 Come leggere un circuito completo

Un circuito deve essere interpretato seguendo alcune regole semplici ma rigorose:

1. si identifica lo stato iniziale di ogni *wire*;
2. si procede da sinistra verso destra;
3. le porte sulla stessa *wire* vengono composte nell'ordine temporale in cui compaiono;
4. le porte su *wire* diverse e nella stessa colonna rappresentano operazioni su sottosistemi differenti;
5. i collegamenti verticali identificano porte controllate o operazioni che coinvolgono più qubit;
6. le misure trasformano l'informazione quantistica in risultati classici.

Esempio 7.7.1 (Traduzione completa di un diagramma)**.** Consideriamo



Il primo livello contiene

$$U_1 = H \otimes I,$$

il secondo

$$U_2 = \text{CNOT},$$

e il terzo

$$U_3 = Z \otimes H.$$

L'operatore complessivo è quindi

$$U_{\text{tot}} = (Z \otimes H)\ \text{CNOT}\ (H \otimes I).$$

Applicandolo a $|00\rangle$, dopo il primo livello si ottiene

$$|\psi_1\rangle = \frac{|00\rangle + |10\rangle}{\sqrt{2}}.$$

Dopo la CNOT,

$$|\psi_2\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$

Infine,

$$\begin{aligned} |\psi_3\rangle &= (Z \otimes H)\frac{|00\rangle + |11\rangle}{\sqrt{2}} \\ &= \frac{1}{2}\left(|00\rangle + |01\rangle - |10\rangle + |11\rangle\right). \end{aligned}$$

Una misura dei due qubit nella base computazionale restituisce quindi ciascuno dei quattro risultati con probabilità

$$\frac{1}{4}.$$

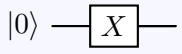
7.8 Esercizi svolti

Gli esercizi che seguono hanno lo scopo di consolidare la lettura dei diagrammi circuitali e il passaggio sistematico tra tre descrizioni equivalenti:

$$\boxed{\text{diagramma} \longleftrightarrow \text{sequenza di operatori} \longleftrightarrow \text{stato finale}}$$

Gli esempi sono ordinati per difficoltà crescente. In ogni caso conviene procedere identificando prima lo stato iniziale, poi le porte nell'ordine temporale e infine l'eventuale misura.

Esempio 7.8.1 (Una sola porta X)**.** Consideriamo il circuito



La porta X agisce come

$$X|0\rangle = |1\rangle.$$

Pertanto lo stato finale è

$$\boxed{|\psi_{\mathrm{f}}\rangle = |1\rangle}.$$

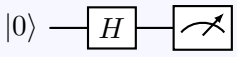
Se al termine del circuito si eseguisse una misura nella base computazionale, il risultato sarebbe 1 con probabilità uno.
In forma matriciale,

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

e quindi

$$X|0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle.$$

Esempio 7.8.2 (Hadamard e probabilità di misura)**.** Consideriamo



Dopo la porta di Hadamard,

$$|\psi\rangle = H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}.$$

Le ampiezze sono

$$\alpha_0 = \frac{1}{\sqrt{2}}, \qquad \alpha_1 = \frac{1}{\sqrt{2}},$$

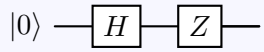
e quindi

$$p(0) = \left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2}, \qquad p(1) = \left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2}.$$

Dunque

$$\boxed{p(0) = p(1) = \frac{1}{2}}.$$

Esempio 7.8.3 (Due porte in successione)**.** Consideriamo



Le porte si leggono da sinistra verso destra, quindi

$$|\psi_{\mathrm{f}}\rangle = ZH|0\rangle.$$

Dopo la prima porta,

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}.$$

Applicando poi Z,

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

Pertanto

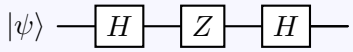
$$\boxed{|\psi_{\mathrm{f}}\rangle = |-\rangle}.$$

L'operatore complessivo è

$$U_{\mathrm{tot}} = ZH,$$

non HZ: la porta che agisce per prima compare a destra nel prodotto.

Esempio 7.8.4 (Verifica dell'identità $HZH = X$)**.** Consideriamo



L'operatore complessivo è

$$U = HZH.$$

Poiché

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

si ha

$$ZH = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}.$$

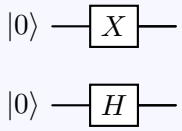
Moltiplicando a sinistra per H,

$$\begin{aligned} HZH &= \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \\ &= \frac{1}{2}\begin{pmatrix} 0 & 2 \\ 2 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = X. \end{aligned}$$

Quindi

$$\boxed{HZH = X}.$$

Esempio 7.8.5 (Porte parallele su un ingresso diverso)**.** Consideriamo il circuito

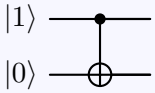


che rappresenta l'operatore $X \otimes H$. Sullo stato iniziale $|00\rangle$ si ha

$$\begin{aligned}(X \otimes H)|00\rangle &= X|0\rangle \otimes H|0\rangle \\ &= |1\rangle \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} \\ &= \frac{|10\rangle + |11\rangle}{\sqrt{2}}.\end{aligned}$$

Una misura nella base computazionale restituisce quindi 10 oppure 11 con probabilità $1/2$ ciascuno.

Esempio 7.8.6 (CNOT su uno stato della base)**.** Consideriamo



Il primo qubit è il controllo e vale 1; il secondo è il bersaglio e vale 0. Poiché il controllo è attivo, sul secondo qubit viene applicata X:

$$|10\rangle \longmapsto |11\rangle.$$

Quindi

$$\boxed{\text{CNOT}|10\rangle = |11\rangle}.$$

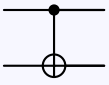
Se lo stato iniziale fosse $|00\rangle$, il controllo varrebbe 0 e il bersaglio non verrebbe modificato:

$$\text{CNOT}|00\rangle = |00\rangle.$$

Esempio 7.8.7 (CNOT su una sovrapposizione)**.** Consideriamo

$$|\psi_{\text{i}}\rangle = \frac{|00\rangle + |10\rangle}{\sqrt{2}}$$

e applichiamo una CNOT con il primo qubit come controllo:



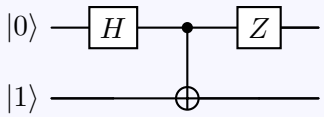
Per linearità,

$$\begin{aligned}\mathrm{CNOT}|\psi_{\mathrm{i}}\rangle &= \frac{1}{\sqrt{2}}\left(\mathrm{CNOT}|00\rangle + \mathrm{CNOT}|10\rangle\right)\\ &= \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right).\end{aligned}$$

Pertanto

$$\boxed{|\psi_{\mathrm{f}}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}}.$$

Esempio 7.8.8 (Preparazione dello stato di Bell $|\Psi^-\rangle$)**.** Partiamo da $|01\rangle$ e consideriamo



Dopo la porta H sul primo qubit,

$$(H \otimes I)|01\rangle = \frac{|01\rangle + |11\rangle}{\sqrt{2}}.$$

La CNOT produce $|\Psi^+\rangle = (|01\rangle + |10\rangle)/\sqrt{2}$; infine la porta Z sul primo qubit cambia il segno della sola componente $|10\rangle$. Pertanto

$$\boxed{|\psi_{\mathrm{f}}\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}} = |\Psi^-\rangle}.$$

Esempio 7.8.9 (Controlled-Z su una sovrapposizione uniforme)**.** Consideriamo

$$|\psi\rangle = \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right).$$

Una controlled-Z lascia inalterati $|00\rangle$, $|01\rangle$ e $|10\rangle$, mentre cambia il segno della componente $|11\rangle$. Pertanto

$$\boxed{CZ|\psi\rangle = \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle - |11\rangle\right)}.$$

Le probabilità di misura nella base computazionale rimangono tutte pari a

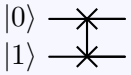
$$\frac{1}{4}.$$

È cambiata però una fase relativa, che può influenzare il risultato di operazioni successive.

Esempio 7.8.10 (Verifica della porta SWAP)**.** Consideriamo

$$|01\rangle$$

e la porta



Per definizione,

$$\mathrm{SWAP}|a,b\rangle = |b,a\rangle.$$

Pertanto

$$\mathrm{SWAP}|01\rangle = |10\rangle.$$

In forma matriciale,

$$\mathrm{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

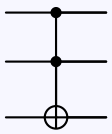
Quindi

$$\mathrm{SWAP}|01\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = |10\rangle.$$

Esempio 7.8.11 (Porta Toffoli su una sovrapposizione)**.** Consideriamo

$$|\psi_{\mathrm{i}}\rangle = \frac{|100\rangle + |110\rangle}{\sqrt{2}}$$

e una Toffoli con i primi due qubit come controlli:



Sulla componente $|100\rangle$ i due controlli valgono 1 e 0, quindi

$$|100\rangle \mapsto |100\rangle.$$

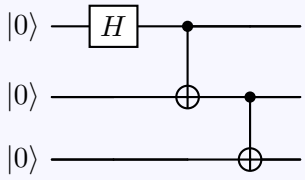
Sulla componente $|110\rangle$ entrambi i controlli valgono 1, quindi

$$|110\rangle \mapsto |111\rangle.$$

Per linearità,

$$\boxed{|\psi_{\mathrm{f}}\rangle = \frac{|100\rangle + |111\rangle}{\sqrt{2}}}.$$

Esempio 7.8.12 (Preparazione di uno stato GHZ)**.** Consideriamo il circuito a tre qubit



Partendo da $|000\rangle$, dopo la porta H si ottiene

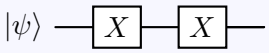
$$\frac{|000\rangle + |100\rangle}{\sqrt{2}}.$$

La prima CNOT correla i primi due qubit e la seconda trasferisce la correlazione al terzo:

$$\frac{|000\rangle + |110\rangle}{\sqrt{2}} \longmapsto \boxed{\frac{|000\rangle + |111\rangle}{\sqrt{2}}}.$$

Una misura completa restituisce soltanto 000 oppure 111, con probabilità $1/2$ ciascuno.

Esempio 7.8.13 (Riconoscere un circuito equivalente)**.** Consideriamo



Poiché

$$X^2 = I,$$

l'operatore complessivo è

$$U = XX = I.$$

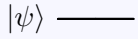
Quindi, per qualunque

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

si ha

$$XX|\psi\rangle = |\psi\rangle.$$

Il circuito è equivalente a



e realizza quindi l'identità.

Esempio 7.8.14 (Dallo stato finale alle probabilità)**.** Supponiamo che l'analisi di un circuito conduca allo stato

$$|\psi_{\mathrm{f}}\rangle = \frac{1}{2}|00\rangle + \frac{\sqrt{3}}{2}|11\rangle.$$

La normalizzazione è corretta:

$$\left|\frac{1}{2}\right|^2 + \left|\frac{\sqrt{3}}{2}\right|^2 = \frac{1}{4} + \frac{3}{4} = 1.$$

Una misura nella base computazionale produce

$$00 \quad \text{con probabilità} \quad \frac{1}{4},$$

e

$$11 \quad \text{con probabilità} \quad \frac{3}{4}.$$

Gli stati 01 e 10 hanno probabilità nulla.
Se il circuito viene eseguito idealmente 1000 volte, ci si aspetta un numero di risultati dell'ordine di

$$250$$

per 00 e

$$750$$

per 11, con normali fluttuazioni statistiche.

7.9 Sintesi

La rappresentazione grafica dei circuiti quantistici costituisce un linguaggio compatto ma rigorosamente collegato alla descrizione matematica. Una *wire* identifica un qubit, una porta rappresenta un operatore unitario, la disposizione orizzontale stabilisce l'ordine temporale e i collegamenti verticali indicano operazioni che coinvolgono più qubit.

La lettura corretta di un circuito richiede quindi di saper passare continuamente tra

$$\text{diagramma}, \qquad \text{operatore}, \qquad \text{stato}.$$

Questa capacità sarà fondamentale quando i circuiti verranno utilizzati per rappresentare algoritmi quantistici veri e propri.

Capitolo 8

Algoritmi quantistici

Nei capitoli precedenti abbiamo introdotto il modello circuitale della computazione quantistica e la sua rappresentazione grafica. Possiamo ora utilizzare questi strumenti per studiare i primi algoritmi quantistici completi.

Gli algoritmi di Deutsch e Deutsch–Jozsa non sono importanti per le applicazioni pratiche dirette, ma svolgono un ruolo didattico fondamentale. Essi mostrano, in una forma particolarmente trasparente, come un algoritmo quantistico possa utilizzare una sovrapposizione, codificare informazione nelle fasi, far interferire le ampiezze e infine trasformare una proprietà globale di una funzione in un risultato di misura.

Il punto di partenza sarà sempre il confronto con la computazione classica. In particolare, confronteremo il numero di volte in cui un algoritmo deve interrogare una funzione, cioè la sua *query complexity*.

8.1 *Promise problems* e modello a *oracle*

Prima di affrontare gli algoritmi di Deutsch e Deutsch–Jozsa è utile chiarire il tipo di problema che essi risolvono. Entrambi appartengono alla classe dei cosiddetti *promise problems*.

Che cos'è un *promise problem*

In un problema ordinario si richiede a un algoritmo di fornire una risposta corretta per tutti gli input ammessi dalla definizione del problema. In un *promise problem*, invece, si dispone fin dall'inizio di un'informazione aggiuntiva: è garantito che l'input appartenga a una determinata classe di casi.

Questa informazione prende il nome di *promise*. La *promise* non è qualcosa che l'algoritmo deve verificare: è un'ipotesi del problema e può quindi essere utilizzata nel procedimento di soluzione.

Un'analogia elementare si incontra frequentemente anche in fisica. Se un problema afferma che un corpo si muove senza attrito, non è richiesto di verificare sperimentalmente l'assenza di attrito: questa condizione viene assunta come parte dei dati del problema. Analogamente, in un *promise problem* l'algoritmo può dare per certa la condizione specificata dalla *promise*.

La conseguenza è importante. Poiché alcune configurazioni dell'input sono escluse a priori, l'algoritmo può talvolta risolvere il problema senza dover acquisire tutta l'informazione che

sarebbe necessaria nel caso generale.

Esempio 8.1.1 (Alcuni esempi di *promise problem*)**.** Consideriamo tre situazioni molto semplici.

- Una sequenza di bit è garantita essere costituita oppure da bit tutti uguali, oppure da un numero uguale di 0 e di 1. Si vuole stabilire quale dei due casi si presenta.
- Una macchina produce componenti di due soli tipi, A e B. Il controllo di produzione garantisce che un lotto sia oppure interamente costituito da componenti dello stesso tipo, oppure formato esattamente per metà da componenti A e per metà da componenti B. Si vuole classificare il lotto.
- Un insieme di sensori restituisce soltanto i valori 0 e 1. È garantito che oppure tutti i sensori restituiscano lo stesso valore, oppure esattamente metà restituisca 0 e metà restituisca 1. Si vuole riconoscere quale situazione si verifica.

Nei tre esempi il problema non consiste nel ricostruire completamente l'input, ma nel determinarne una proprietà globale facendo uso dell'informazione fornita dalla *promise*.

Esempio 8.1.2 (Un problema classico con otto dispositivi)**.** Consideriamo ora otto dispositivi elettronici, numerati da 0 a 7. Ciascun dispositivo può trovarsi in uno dei due stati

$$0 \qquad \text{oppure} \qquad 1.$$

Supponiamo che il processo di produzione garantisca che il sistema si trovi in una, e soltanto una, delle seguenti condizioni:

- tutti gli otto dispositivi si trovano nello stesso stato;
- quattro dispositivi si trovano nello stato 0 e quattro nello stato 1.

Questa è la *promise*. Configurazioni come sei dispositivi nello stato 0 e due nello stato 1 sono escluse dal problema.
Il nostro compito è stabilire se il sistema sia *uniforme* oppure *bilanciato*, potendo controllare un dispositivo alla volta.
Supponiamo che i primi quattro controlli forniscano

$$0, \qquad 0, \qquad 0, \qquad 0.$$

Dopo quattro controlli non possiamo ancora decidere.
I risultati osservati sono infatti compatibili con un sistema uniforme:

$$0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,$$

ma sono anche compatibili con un sistema bilanciato:

$$0,\ 0,\ 0,\ 0,\ 1,\ 1,\ 1,\ 1.$$

Occorre quindi controllare almeno un altro dispositivo.
Se il quinto dispositivo si trova nello stato 1, abbiamo osservato sia uno 0 sia un 1 e, grazie alla *promise*, possiamo concludere immediatamente che il sistema è bilanciato.

Se invece il quinto risultato è ancora 0, abbiamo osservato cinque dispositivi nello stato 0. Un sistema bilanciato potrebbe contenerne soltanto quattro. La *promise* ci permette quindi di concludere che il sistema è uniforme, senza controllare i tre dispositivi rimanenti.
Nel caso peggiore sono pertanto necessari cinque controlli.
Il punto essenziale è che la *promise* modifica il problema computazionale. Senza di essa, dopo aver trovato cinque zeri non potremmo dedurre lo stato dei tre dispositivi non ancora esaminati. Con la *promise*, invece, molte configurazioni sono escluse e l'informazione già raccolta può essere sufficiente.

Dai dispositivi a una funzione

Possiamo descrivere lo stesso problema in modo più astratto associando a ogni dispositivo un indice x e indicando con

$$f(x)$$

lo stato del dispositivo corrispondente.

Non ci interessa conoscere l'intera tabella dei valori della funzione. Vogliamo soltanto stabilire se essa possiede una delle due proprietà ammesse dalla *promise*:

- f è *costante*, se restituisce lo stesso valore per tutti gli input;
- f è *bilanciata*, se restituisce 0 per metà degli input e 1 per l'altra metà.

Il problema diventa quindi:

> Sapendo che la funzione è costante oppure bilanciata, determinare quale delle due possibilità si verifica.

Questa formulazione è precisamente quella che incontreremo nell'algoritmo di Deutsch–Jozsa. Il problema di Deutsch ne rappresenterà il caso più semplice, con un solo bit di input.

L'*oracle* classico

Negli esempi precedenti abbiamo implicitamente supposto di poter scegliere un dispositivo e conoscerne lo stato. In termini computazionali, stiamo assumendo di poter interrogare la funzione f.

Il meccanismo astratto che fornisce questa possibilità viene chiamato *oracle*. Possiamo immaginarlo come una *black box*: non conosciamo necessariamente il modo in cui la funzione sia implementata al suo interno, ma possiamo fornire un input x e ricevere in risposta il valore $f(x)$.

Ogni interrogazione dell'*oracle* viene chiamata *query*.

Nel problema degli otto dispositivi, controllare il dispositivo numero 3 significa quindi effettuare una *query* con input $x = 3$. Se il dispositivo si trova nello stato 0, la risposta dell'*oracle* è semplicemente

$$f(3) = 0.$$

Il procedimento classico descritto in precedenza richiede dunque, nel caso peggiore, cinque *query*. Questa modalità di conteggio sarà particolarmente utile quando confronteremo algoritmi classici

e quantistici: invece di preoccuparci inizialmente di come sia costruita internamente la funzione, conteremo quante volte l'algoritmo ha bisogno di interrogarla.

Perché l'*oracle* quantistico deve essere reversibile

Il passaggio alla computazione quantistica richiede una modifica importante. Una porta quantistica rappresenta una trasformazione unitaria e deve quindi essere reversibile.

Potremmo essere tentati di rappresentare una valutazione della funzione mediante

$$|x\rangle \longmapsto |f(x)\rangle.$$

Questa trasformazione, tuttavia, in generale non è reversibile.

Esempio 8.1.3 (Perché $|x\rangle \mapsto |f(x)\rangle$ non è in generale reversibile)**.** Supponiamo che due input diversi producano lo stesso valore:

$$f(0) = f(1) = 0.$$

Avremmo allora

$$|0\rangle \longmapsto |0\rangle, \qquad |1\rangle \longmapsto |0\rangle.$$

Due stati iniziali diversi verrebbero trasformati nello stesso stato finale. Dallo stato finale non sarebbe più possibile ricostruire quale fosse l'input iniziale, e una trasformazione di questo tipo non può essere unitaria.

Per rendere reversibile l'operazione si conserva quindi l'input x e si introduce un secondo registro, costituito in questo caso da un solo qubit. L'*oracle* quantistico viene definito da

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

dove $\oplus$ indica la somma modulo 2.

Il primo registro conserva l'indice x; il secondo viene modificato in funzione del valore $f(x)$.

Se il secondo qubit è inizializzato nello stato $|0\rangle$, otteniamo in particolare

$$U_f|x\rangle|0\rangle = |x\rangle|f(x)\rangle.$$

In questo modo l'*oracle* quantistico riproduce una normale interrogazione della funzione, ma lo fa attraverso una trasformazione reversibile.

Esempio 8.1.4 (Una *query* all'*oracle* quantistico)**.** Supponiamo di avere tre qubit per rappresentare l'indice degli otto dispositivi e che

$$f(3) = 1.$$

Poiché

$$3 = 011$$

in rappresentazione binaria, una *query* con il secondo registro inizializzato in $|0\rangle$ produce

$$U_f|011\rangle|0\rangle = |011\rangle|1\rangle.$$

L'indice del dispositivo viene conservato e il valore della funzione compare nel secondo registro.

Interrogare l'*oracle* con una sovrapposizione

Fin qui l'*oracle* quantistico sembra fare poco più di una normale interrogazione classica. La differenza emerge quando il registro di ingresso non contiene un singolo valore di x, ma una sovrapposizione.

Esempio 8.1.5 (Interrogare l'*oracle* con due input in sovrapposizione)**.** Consideriamo due possibili input x_1 e x_2. Se prepariamo lo stato

$$\frac{|x_1\rangle + |x_2\rangle}{\sqrt{2}}|0\rangle,$$

la linearità dell'*oracle* produce

$$U_f\left(\frac{|x_1\rangle + |x_2\rangle}{\sqrt{2}}|0\rangle\right) = \frac{|x_1\rangle|f(x_1)\rangle + |x_2\rangle|f(x_2)\rangle}{\sqrt{2}}.$$

Con una singola applicazione di U_f, l'evoluzione dello stato dipende quindi dai valori della funzione associati a entrambe le componenti della sovrapposizione.
È però essenziale interpretare correttamente questo risultato. Esso non significa che possiamo misurare lo stato e leggere contemporaneamente $f(x_1)$ e $f(x_2)$. Una misura nella base computazionale restituisce un solo esito. La presenza simultanea di più input in sovrapposizione non equivale quindi ad avere a disposizione una tabella leggibile di tutti i valori della funzione.

Per ottenere un vantaggio computazionale bisogna utilizzare l'*oracle* in modo più sottile: l'informazione associata ai diversi valori di $f(x)$ deve essere codificata nelle ampiezze e nelle loro fasi relative, e queste ampiezze devono successivamente interferire in modo da rendere osservabile la proprietà globale che interessa.

Gli algoritmi di Deutsch e Deutsch–Jozsa forniscono il primo esempio completo di questo meccanismo:

$$\boxed{\text{sovrapposizione} \longrightarrow \textit{oracle} \longrightarrow \text{interferenza} \longrightarrow \text{misura}.}$$

Nel prossimo paragrafo vedremo come questa idea possa essere realizzata nel caso più semplice.

8.2 L'algoritmo di Deutsch

Il problema di Deutsch e la soluzione classica

Il problema di Deutsch può essere visto come il caso più semplice del *promise problem* introdotto nel paragrafo precedente. Consideriamo una funzione booleana con un solo bit di input,

$$f : \{0,1\} \longrightarrow \{0,1\}.$$

Poiché il dominio contiene soltanto i due valori 0 e 1, una funzione di questo tipo è completamente determinata dai due numeri

$$f(0), \qquad f(1).$$

Esistono quindi soltanto quattro funzioni possibili:

	$f(0)$	$f(1)$	tipo di funzione
f_0	0	0	costante
f_1	0	1	bilanciata
f_2	1	0	bilanciata
f_3	1	1	costante

Nel problema di Deutsch la *promise* stabilisce che la funzione appartenga a una delle due classi indicate nella tabella:

- la funzione è *costante*, cioè
$$f(0) = f(1);$$
- la funzione è *bilanciata*, cioè
$$f(0) \neq f(1).$$

Nel caso di un solo bit di input non esistono altre possibilità: una funzione booleana è necessariamente costante oppure bilanciata. La formulazione come *promise problem* acquisterà un significato più sostanziale nella generalizzazione di Deutsch–Jozsa, dove il dominio conterrà molti più elementi e saranno possibili anche funzioni né costanti né bilanciate.

Il compito dell'algoritmo non è stabilire quale delle quattro funzioni della tabella sia stata scelta. In particolare, non interessa distinguere f_0 da f_3. Si vuole determinare soltanto la proprietà globale

$$\boxed{f(0) = f(1) \quad \text{oppure} \quad f(0) \neq f(1).}$$

Questa distinzione può anche essere espressa mediante la somma modulo 2:

$$f(0) \oplus f(1) = \begin{cases} 0, & f \text{ costante}, \\ 1, & f \text{ bilanciata}. \end{cases}$$

L'informazione cercata è dunque un solo bit, ma quel bit dipende dal confronto tra i valori della funzione nei due diversi punti del dominio.

Come procede un algoritmo classico

Supponiamo che la funzione sia disponibile soltanto attraverso un *oracle*. Possiamo scegliere un input x e ottenere il corrispondente valore $f(x)$.

Cominciamo interrogando l'*oracle* in

$$x = 0.$$

Supponiamo che la risposta sia

$$f(0) = 0.$$

Questa singola informazione non permette ancora di decidere. Restano infatti compatibili due funzioni diverse:

	$f(0)$	$f(1)$	
f_0	0	0	costante
f_1	0	1	bilanciata

Per distinguere i due casi è necessario effettuare una seconda *query*, questa volta con

$$x = 1.$$

Se otteniamo

$$f(1) = 0,$$

concludiamo che la funzione è costante. Se invece otteniamo

$$f(1) = 1,$$

concludiamo che è bilanciata.

Naturalmente lo stesso ragionamento vale se la prima risposta è

$$f(0) = 1.$$

In quel caso rimangono possibili una funzione costante,

$$(f(0), f(1)) = (1, 1),$$

e una funzione bilanciata,

$$(f(0), f(1)) = (1, 0),$$

e anche in questo caso è necessario conoscere $f(1)$.

Esempio 8.2.1 (Una procedura classica completa)**.** Supponiamo che l'*oracle* nasconda la funzione

$$f(0) = 1, \qquad f(1) = 0.$$

Una possibile procedura classica è la seguente.
La prima *query* usa l'input $x = 0$:

$$x = 0 \quad \longrightarrow \quad f(0) = 1.$$

Dopo questa risposta non possiamo ancora stabilire se la funzione sia costante o bilanciata. Effettuiamo quindi una seconda *query*:

$$x = 1 \quad \longrightarrow \quad f(1) = 0.$$

Ora possiamo confrontare i due risultati:

$$f(0) \neq f(1),$$

e concludere che la funzione è bilanciata.
La procedura ha utilizzato due *query*.

Il punto importante non è che abbiamo scelto di interrogare prima $x = 0$. Avremmo potuto iniziare da $x = 1$ e il risultato non sarebbe cambiato: dopo una sola *query* rimarrebbero sempre possibili una funzione costante e una funzione bilanciata.

Perciò nessun algoritmo classico deterministico che debba fornire sempre la risposta esatta può risolvere il problema con una sola *query*. Nel caso peggiore sono necessarie

$$\boxed{Q_{\mathrm{C}} = 2}$$

query all'*oracle*.

Questo costituisce il riferimento classico con cui confronteremo l'algoritmo quantistico. La domanda che guiderà il prossimo passaggio è quindi molto precisa:

> È possibile determinare con certezza se f è costante oppure bilanciata utilizzando una sola *query* all'*oracle* quantistico?

L'idea quantistica

Il confronto classico ha mostrato che una singola *query* non è sufficiente se l'*oracle* viene interrogato con un solo valore di input alla volta. Dopo aver conosciuto soltanto $f(0)$, oppure soltanto $f(1)$, rimangono sempre compatibili una funzione costante e una funzione bilanciata.

La computazione quantistica mette però a disposizione una possibilità che non ha un equivalente classico diretto: il registro di ingresso dell'*oracle* può essere preparato in una sovrapposizione dei due valori

$$0 \qquad \text{e} \qquad 1.$$

Invece di interrogare separatamente l'*oracle* con $|0\rangle$ e con $|1\rangle$, possiamo preparare il primo qubit nello stato

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

Se il secondo qubit dell'*oracle* viene inizializzato in $|0\rangle$, una singola applicazione di U_f produce

$$U_f \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} |0\rangle \right) = \frac{|0\rangle|f(0)\rangle + |1\rangle|f(1)\rangle}{\sqrt{2}}.$$

A prima vista questo risultato potrebbe sembrare sufficiente: nello stato finale compaiono infatti sia $f(0)$ sia $f(1)$. Tuttavia la conclusione sarebbe errata.

Se misuriamo i due qubit nella base computazionale, otteniamo soltanto uno dei due possibili rami:

$$|0\rangle|f(0)\rangle \qquad \text{oppure} \qquad |1\rangle|f(1)\rangle.$$

La misura non ci restituisce contemporaneamente i due valori della funzione. In questo uso "diretto" della sovrapposizione, quindi, una singola *query* quantistica non ci fornisce ancora l'informazione necessaria per stabilire se la funzione sia costante o bilanciata.

Esempio 8.2.2 (Perché la semplice sovrapposizione non basta)**.** Supponiamo che la funzione sia bilanciata e soddisfi

$$f(0) = 0, \qquad f(1) = 1.$$

Preparando

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

e applicando l'*oracle*, otteniamo

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

Se misuriamo lo stato, otteniamo

$$00$$

con probabilità $1/2$ oppure

$$11$$

con probabilità $1/2$.
Nel primo caso veniamo a conoscenza soltanto del fatto che

$$f(0) = 0;$$

nel secondo soltanto del fatto che

$$f(1) = 1.$$

In entrambi i casi, considerato isolatamente, il risultato di una singola misura non è sufficiente a ricostruire il confronto tra $f(0)$ e $f(1)$.
La sovrapposizione ha quindi permesso all'*oracle* di agire coerentemente su entrambi gli input, ma la misura diretta non consente di leggere simultaneamente i due risultati.

Il problema deve allora essere affrontato in modo diverso.

L'obiettivo dell'algoritmo di Deutsch non sarà quello di memorizzare separatamente i due valori

$$f(0) \qquad \text{e} \qquad f(1),$$

ma di fare in modo che la loro *relazione* venga codificata nello stato quantistico.

In altre parole, non vogliamo rispondere alle due domande

$$\text{"quanto vale } f(0)\text{?"} \qquad \text{e} \qquad \text{"quanto vale } f(1)\text{?"},$$

ma direttamente alla domanda

$$\boxed{\text{"i due valori sono uguali oppure diversi?"}}$$

Per ottenere questo risultato, l'*oracle* verrà utilizzato in una configurazione particolare. Il secondo qubit non sarà preparato in $|0\rangle$, ma in uno stato scelto appositamente affinché l'informazione contenuta in $f(x)$ non rimanga registrata come un normale bit di uscita. Essa verrà invece trasferita nella fase delle ampiezze del primo qubit.

Una successiva operazione di interferenza convertirà questa differenza di fase in un risultato direttamente osservabile mediante una misura.

L'idea dell'algoritmo può quindi essere anticipata schematicamente come

$$\boxed{\text{sovrapposizione} \longrightarrow \text{codifica nella fase} \longrightarrow \text{interferenza} \longrightarrow \text{misura.}}$$

Il prossimo passo consiste nel costruire lo stato iniziale necessario per realizzare questa strategia.

Preparazione dello stato

Per realizzare la strategia appena descritta l'algoritmo utilizza due qubit.

Il primo qubit rappresenta l'input della funzione f e verrà utilizzato per interrogare l'*oracle* in sovrapposizione. Il secondo è un *ancilla qubit*: non rappresenta un secondo input indipendente, ma serve a rendere possibile l'azione reversibile dell'*oracle* e, come vedremo tra poco, a trasferire l'informazione su $f(x)$ nella fase del primo qubit.

Lo stato iniziale scelto dall'algoritmo è

$$\boxed{|\psi_0\rangle = |0\rangle|1\rangle.}$$

La scelta dei due valori iniziali non è casuale. Il primo qubit viene inizializzato in $|0\rangle$ perché vogliamo trasformarlo, mediante una porta di Hadamard, nella sovrapposizione uniforme dei due possibili input della funzione. Il secondo viene invece inizializzato in $|1\rangle$ perché la stessa porta di Hadamard lo trasformerà nello stato $|-\rangle$, che avrà un ruolo essenziale nel *phase kickback*.

Applichiamo quindi una porta di Hadamard a entrambi i qubit:

$$|\psi_1\rangle = (H \otimes H)|0\rangle|1\rangle.$$

Per il primo qubit,

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = |+\rangle.$$

Per il secondo,

$$H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |-\rangle.$$

Pertanto

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

Se sviluppiamo il prodotto tensoriale otteniamo

$$|\psi_1\rangle = \frac{1}{2}\left(|00\rangle - |01\rangle + |10\rangle - |11\rangle\right).$$

Le due forme sono equivalenti, ma in questa fase la forma fattorizzata

$$|+\rangle|-\rangle$$

è più utile perché rende immediatamente visibile il ruolo distinto dei due qubit.

Il primo qubit si trova ora nella sovrapposizione

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

e quindi contiene, con la stessa ampiezza, entrambi i possibili input della funzione.

L'*ancilla qubit*, invece, si trova nello stato

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

È importante osservare che questo secondo qubit non viene preparato in una sovrapposizione per "provare due valori di y contemporaneamente". La sua funzione è diversa: la particolare differenza di segno tra $|0\rangle$ e $|1\rangle$ farà sì che l'azione dell'*oracle* produca un segno dipendente da $f(x)$ senza modificare lo stato dell'*ancilla qubit*.

Esempio 8.2.3 (Lo stato prima dell'*oracle*)**.** Partendo da

$$|\psi_0\rangle = |0\rangle|1\rangle,$$

le due porte Hadamard producono

$$(H \otimes H)|0\rangle|1\rangle = \frac{1}{2}\left(|00\rangle - |01\rangle + |10\rangle - |11\rangle\right).$$

Raggruppando i termini rispetto al primo qubit,

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}\left[|0\rangle\frac{|0\rangle - |1\rangle}{\sqrt{2}} + |1\rangle\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right].$$

Quindi

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

La riscrittura mostra chiaramente che il primo qubit è pronto a interrogare la funzione sui due input 0 e 1 in sovrapposizione, mentre il secondo è stato preparato nello stato $|-\rangle$.

A questo punto la preparazione è completa. Prima di applicare l'*oracle* dobbiamo però capire perché lo stato $|-\rangle$ sia così speciale e in che modo consenta di convertire il valore $f(x)$ in una fase.

Perché si prepara l'*ancilla qubit* in $|-\rangle$: il *phase kickback*

A questo punto dobbiamo comprendere la ragione della scelta

$$|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

per l'*ancilla qubit*.

Ricordiamo che l'*oracle* quantistico agisce secondo

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

Per un valore fissato di x, il comportamento del secondo qubit dipende quindi soltanto dal valore $f(x)$:

- se $f(x) = 0$, il secondo qubit rimane invariato;
- se $f(x) = 1$, il secondo qubit viene invertito, cioè $|0\rangle$ e $|1\rangle$ vengono scambiati.

Vediamo che cosa accade quando il secondo qubit è proprio nello stato $|-\rangle$.

Caso $f(x) = 0$

Se

$$f(x) = 0,$$

l'*oracle* non modifica il secondo qubit. Pertanto

$$U_f|x\rangle|-\rangle = |x\rangle|-\rangle.$$

Possiamo verificarlo esplicitamente:

$$\begin{aligned} U_f|x\rangle|-\rangle &= U_f\left[|x\rangle\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \\ &= \frac{1}{\sqrt{2}}\left(U_f|x\rangle|0\rangle - U_f|x\rangle|1\rangle\right) \\ &= \frac{1}{\sqrt{2}}\left(|x\rangle|0\rangle - |x\rangle|1\rangle\right) \\ &= |x\rangle|-\rangle. \end{aligned}$$

Non compare quindi alcun cambiamento di segno.

Caso $f(x) = 1$

Se invece

$$f(x) = 1,$$

l'*oracle* inverte il secondo qubit:

$$|0\rangle \longleftrightarrow |1\rangle.$$

Applicandolo allo stato $|-\rangle$ otteniamo

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

Il secondo qubit è ancora nello stato $|-\rangle$, ma lo stato complessivo ha acquistato un fattore -1.

I due casi possono essere riassunti in una sola espressione:

$$\boxed{U_f|x\rangle|-\rangle = (-1)^{f(x)}|x\rangle|-\rangle.}$$

Infatti,

$$(-1)^{f(x)} = \begin{cases} +1, & f(x) = 0, \\ -1, & f(x) = 1. \end{cases}$$

Questo meccanismo prende il nome di *phase kickback*.

Che cosa è successo all'informazione $f(x)$?

Se avessimo preparato l'*ancilla qubit* nello stato $|0\rangle$, l'informazione sarebbe comparsa nel secondo registro nel modo più diretto:

$$|x\rangle|0\rangle \longmapsto |x\rangle|f(x)\rangle.$$

Preparandolo invece in $|-\rangle$, il valore $f(x)$ non rimane scritto come un bit nel secondo registro. Esso determina il fattore

$$(-1)^{f(x)}$$

che moltiplica la componente associata a $|x\rangle$.

L'*ancilla qubit*, dopo l'applicazione dell'*oracle*, rimane nello stesso stato $|-\rangle$:

$$|-\rangle \longmapsto |-\rangle.$$

L'informazione fornita dalla funzione compare invece come una fase.

Bisogna però precisare un punto importante. Se il registro di ingresso contenesse un solo stato $|x\rangle$, il fattore $(-1)^{f(x)}$ sarebbe una fase globale e non produrrebbe da solo alcun effetto osservabile. L'utilità del meccanismo emerge quando il primo qubit si trova in sovrapposizione.

Supponiamo infatti di avere due componenti,

$$\alpha|0\rangle + \beta|1\rangle.$$

L'*oracle* produce

$$\alpha(-1)^{f(0)}|0\rangle + \beta(-1)^{f(1)}|1\rangle,$$

mentre l'*ancilla qubit* rimane in $|-\rangle$.

Se

$$f(0) = f(1),$$

le due componenti acquistano lo stesso segno. La modifica è allora soltanto una fase globale.

Se invece

$$f(0) \neq f(1),$$

una componente acquista un segno opposto rispetto all'altra. Si crea quindi una *fase relativa*, che può avere conseguenze osservabili quando le due componenti vengono fatte interferire.

È precisamente questa seconda situazione che l'algoritmo di Deutsch sfrutta.

Esempio 8.2.4 (Il segno introdotto dall'*oracle*)**.** Consideriamo separatamente i due possibili valori di $f(x)$.
Se

$$f(x) = 0,$$

si ha

$$|x\rangle|-\rangle \longmapsto +|x\rangle|-\rangle.$$

Se

$$f(x) = 1,$$

si ha invece

$$|x\rangle|-\rangle \longmapsto -|x\rangle|-\rangle.$$

Supponiamo ora che il primo qubit sia nello stato

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

Se

$$f(0) = 0, \qquad f(1) = 1,$$

le due componenti subiscono trasformazioni diverse:

$$|0\rangle \longmapsto +|0\rangle, \qquad |1\rangle \longmapsto -|1\rangle.$$

Pertanto

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

L'informazione "i due valori della funzione sono diversi" non è stata letta come una coppia di bit: è stata trasformata in una differenza di fase tra le due componenti del primo qubit.

Possiamo anche riconoscere il motivo algebrico per cui $|-\rangle$ possiede questa proprietà. L'inversione del secondo qubit è realizzata dalla porta X, e

$$X|-\rangle = -|-\rangle.$$

Lo stato $|-\rangle$ è quindi un autostato di X con autovalore -1. Quando $f(x) = 1$, l'*oracle* applica di fatto X all'*ancilla qubit*; poiché questo si trova in un autostato di X, l'azione della porta si riduce alla comparsa del suo autovalore, cioè del fattore -1.

Abbiamo così ottenuto il meccanismo necessario: una singola applicazione dell'*oracle* può codificare

i valori della funzione nelle fasi delle diverse componenti della sovrapposizione.

Nel passo successivo applicheremo questo risultato allo stato

$$|+\rangle|-\rangle$$

preparato in precedenza e vedremo come le due possibilità, funzione costante e funzione bilanciata, producano due strutture di fase differenti.

Applicazione dell'*oracle* allo stato preparato

Possiamo ora applicare il risultato del *phase kickback* allo stato preparato in precedenza,

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

Per linearità,

$$\begin{aligned} |\psi_2\rangle &= U_f|\psi_1\rangle \\ &= \frac{1}{\sqrt{2}}\left(U_f|0\rangle|-\rangle + U_f|1\rangle|-\rangle\right). \end{aligned}$$

Usando

$$U_f|x\rangle|-\rangle = (-1)^{f(x)}|x\rangle|-\rangle,$$

otteniamo

$$\begin{aligned} |\psi_2\rangle &= \frac{1}{\sqrt{2}}\left[(-1)^{f(0)}|0\rangle|-\rangle + (-1)^{f(1)}|1\rangle|-\rangle\right] \\ &= \frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}} \otimes |-\rangle. \end{aligned}$$

Quindi

$$\boxed{|\psi_2\rangle = \frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}} \otimes |-\rangle.}$$

A questo punto l'*ancilla qubit* può essere, per il momento, messo da parte: è ancora nello stato $|-\rangle$ e non contiene l'informazione che vogliamo leggere. Tutta l'informazione utile è stata trasferita nella struttura delle fasi del primo qubit.

Consideriamo ora separatamente i due tipi di funzione ammessi.

Se la funzione è costante

Se

$$f(0) = f(1) = 0,$$

si ha

$$(-1)^{f(0)} = (-1)^{f(1)} = 1,$$

e quindi il primo qubit rimane

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

Se invece

$$f(0) = f(1) = 1,$$

entrambi i coefficienti acquistano un segno meno:

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

Le due possibilità differiscono soltanto per una fase globale. Perciò, dal punto di vista fisico, nel caso costante il primo qubit si trova nello stato

$$\boxed{|+\rangle}$$

a meno di una fase globale.

Se la funzione è bilanciata

Supponiamo ora che

$$f(0) = 0, \qquad f(1) = 1.$$

Allora

$$(-1)^{f(0)} = 1, \qquad (-1)^{f(1)} = -1,$$

e il primo qubit diventa

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

Nell'altro caso bilanciato,

$$f(0) = 1, \qquad f(1) = 0,$$

si ottiene

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

Anche in questo caso le due possibilità differiscono soltanto per una fase globale. Quindi, se la funzione è bilanciata, il primo qubit si trova nello stato

$$\boxed{|-\rangle}$$

a meno di una fase globale.

Abbiamo così raggiunto un punto decisivo. Dopo una sola *query* all'*oracle*, le due classi di funzioni corrispondono a due stati diversi del primo qubit:

$$\boxed{\begin{array}{ccc} f \text{ costante} & \Longrightarrow & |+\rangle \\ f \text{ bilanciata} & \Longrightarrow & |-\rangle \end{array}}$$

dove abbiamo omesso le eventuali fasi globali.

Esempio 8.2.5 (Le quattro funzioni dopo l'*oracle*)**.** Le quattro funzioni possibili producono i seguenti stati del primo qubit:

	$f(0)$	$f(1)$	stato dopo U_f
f_0	0	0	$\|+\rangle$
f_1	0	1	$\|-\rangle$
f_2	1	0	$-\|-\rangle$
f_3	1	1	$-\|+\rangle$

Le due funzioni costanti producono quindi $|+\rangle$, a meno di una fase globale, mentre le due funzioni bilanciate producono $|-\rangle$, sempre a meno di una fase globale.

Sembrerebbe quindi che il problema sia già risolto: dobbiamo soltanto distinguere $|+\rangle$ da $|-\rangle$. Tuttavia la misura che abbiamo finora utilizzato nei circuiti è la misura nella base computazionale,

$$\{|0\rangle, |1\rangle\}.$$

Se misurassimo direttamente $|+\rangle$ oppure $|-\rangle$ in questa base, otterremmo in entrambi i casi

$$0 \quad \text{oppure} \quad 1$$

con probabilità $1/2$.

La differenza tra funzione costante e funzione bilanciata è quindi già presente nello stato quantistico, ma non è ancora espressa nella base in cui vogliamo effettuare la misura.

Serve un'ultima trasformazione: dobbiamo convertire

$$|+\rangle \quad \text{e} \quad |-\rangle$$

rispettivamente in due stati della base computazionale distinti. Questo è precisamente il ruolo della seconda porta di Hadamard.

Interferenza mediante la seconda Hadamard

Applichiamo ora H soltanto al primo qubit.

Per brevità poniamo

$$a = (-1)^{f(0)}, \qquad b = (-1)^{f(1)}.$$

Il primo qubit è

$$\frac{a|0\rangle + b|1\rangle}{\sqrt{2}}.$$

Poiché

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \qquad H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}},$$

si ottiene

$$\begin{aligned} H\frac{a|0\rangle + b|1\rangle}{\sqrt{2}} &= \frac{1}{\sqrt{2}}\left(aH|0\rangle + bH|1\rangle\right) \\ &= \frac{1}{2}\left[a(|0\rangle + |1\rangle) + b(|0\rangle - |1\rangle)\right] \\ &= \frac{1}{2}\left[(a+b)|0\rangle + (a-b)|1\rangle\right]. \end{aligned}$$

Sostituendo a e b,

$$\boxed{|\psi_3\rangle = \frac{1}{2}\left[\left((-1)^{f(0)} + (-1)^{f(1)}\right)|0\rangle + \left((-1)^{f(0)} - (-1)^{f(1)}\right)|1\rangle\right] \otimes |-\rangle.}$$

Ora distinguiamo i due casi.

Se f è costante,

$$f(0) = f(1),$$

quindi

$$(-1)^{f(0)} = (-1)^{f(1)}.$$

Il coefficiente di $|1\rangle$ è nullo e il primo qubit diventa

$$\pm|0\rangle.$$

Se f è bilanciata,

$$f(0) \neq f(1),$$

quindi

$$(-1)^{f(0)} = -(-1)^{f(1)}.$$

Il coefficiente di $|0\rangle$ è nullo e il primo qubit diventa

$$\pm|1\rangle.$$

Il segno complessivo non ha significato fisico, perché costituisce una fase globale.

La misura del primo qubit fornisce quindi

$$\boxed{0 \Longrightarrow f \text{ costante}}$$

e

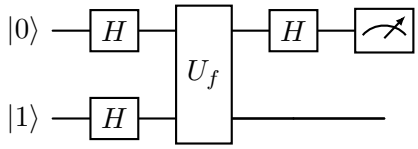
$$\boxed{1 \Longrightarrow f \text{ bilanciata.}}$$

La risposta è deterministica: nel modello ideale la probabilità di successo è

$$1.$$

Circuito dell'algoritmo

Il circuito completo può essere rappresentato come



Il circuito contiene una sola chiamata a

$$U_f.$$

Le porte Hadamard prima dell'*oracle* preparano la sovrapposizione e lo stato dell'*ancilla qubit* $|-\rangle$; l'*oracle* trasferisce l'informazione su f nelle fasi; la Hadamard finale trasforma la fase relativa in un valore della base computazionale che può essere letto mediante misura.

Esempio 8.2.6 (Funzione costante $f(0) = f(1) = 0$)**.** Consideriamo

$$f(0) = 0, \qquad f(1) = 0.$$

Dopo le prime Hadamard,

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

Poiché

$$(-1)^{f(0)} = (-1)^{f(1)} = 1,$$

l'*oracle* lascia invariata la fase di entrambe le componenti:

$$|\psi_2\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |-\rangle = |+\rangle|-\rangle.$$

La Hadamard finale sul primo qubit dà

$$H|+\rangle = |0\rangle.$$

Pertanto

$$|\psi_3\rangle = |0\rangle|-\rangle.$$

La misura del primo qubit restituisce

$$\boxed{0}$$

con probabilità uno, identificando correttamente la funzione come costante.

Esempio 8.2.7 (Funzione costante $f(0) = f(1) = 1$)**.** Consideriamo ora

$$f(0) = 1, \qquad f(1) = 1.$$

Dopo l'*oracle*,

$$|\psi_2\rangle = \frac{-|0\rangle - |1\rangle}{\sqrt{2}} \otimes |-\rangle$$
$$= -|+\rangle|-\rangle.$$

La differenza rispetto all'esempio precedente è soltanto una fase globale -1.
Dopo la Hadamard finale,

$$|\psi_3\rangle = -|0\rangle|-\rangle.$$

La probabilità di misurare 0 sul primo qubit è ancora

$$|-1|^2 = 1.$$

Quindi anche in questo caso il risultato è

$$\boxed{0}.$$

L'esempio mostra concretamente perché una fase globale non modifica le probabilità di misura.

Esempio 8.2.8 (Funzione bilanciata $f(x) = x$)**.** Consideriamo

$$f(0) = 0, \qquad f(1) = 1.$$

Dopo le prime Hadamard,

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

L'*oracle* introduce le fasi

$$(-1)^{f(0)} = 1, \qquad (-1)^{f(1)} = -1,$$

per cui

$$|\psi_2\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes |-\rangle = |-\rangle|-\rangle.$$

Applicando la Hadamard finale,

$$H|-\rangle = |1\rangle.$$

Quindi

$$|\psi_3\rangle = |1\rangle|-\rangle.$$

La misura del primo qubit restituisce

$$\boxed{1}$$

con probabilità uno, e la funzione viene riconosciuta come bilanciata.

Il riepilogo dei quattro casi è riportato nella definition 8.2.5. La seconda Hadamard utilizza

$$H|+\rangle = |0\rangle, \qquad H|-\rangle = |1\rangle.$$

Perciò f_0 e f_3, costanti, producono il risultato 0, mentre f_1 e f_2, bilanciate, producono il risultato

1, indipendentemente dall'eventuale fase globale.

Che cosa ha ottenuto l'algoritmo

L'algoritmo classico deterministico richiede

$$2$$

query.

L'algoritmo quantistico richiede

$$1$$

query.

Il punto concettualmente importante, tuttavia, non è il risparmio di una singola interrogazione. Il problema di Deutsch mostra per la prima volta la struttura che ricorrerà in molti algoritmi quantistici:

$$\boxed{\text{sovrapposizione} \longrightarrow \textit{oracle} \longrightarrow \text{fase} \longrightarrow \text{interferenza} \longrightarrow \text{misura.}}$$

L'*oracle* non permette di leggere simultaneamente $f(0)$ e $f(1)$. Esso permette invece di far dipendere le fasi delle ampiezze dai valori della funzione. La successiva interferenza elimina l'informazione non necessaria e rende osservabile direttamente la proprietà globale

$$f(0) \oplus f(1).$$

8.3 L'algoritmo di Deutsch–Jozsa

L'algoritmo di Deutsch–Jozsa generalizza la stessa idea a una funzione di n bit.

Il problema

Consideriamo

$$f : \{0,1\}^n \longrightarrow \{0,1\}$$

e supponiamo che sia garantita una delle seguenti due possibilità:

1. f è *costante*, cioè assume lo stesso valore per tutti i 2^n input;
2. f è *bilanciata*, cioè assume il valore 0 su esattamente metà degli input e il valore 1 sull'altra metà.

L'obiettivo è stabilire quale delle due proprietà sia verificata.

La *promise* è essenziale. Una funzione booleana generica può infatti essere né costante né bilanciata.

Esempio 8.3.1 (Significato di funzione bilanciata per $n = 3$)**.** Con

$$n = 3$$

esistono

$$2^3 = 8$$

possibili input. Una funzione bilanciata deve assumere il valore 0 esattamente quattro volte e il valore 1 esattamente quattro volte.

Costo classico deterministico

Nel caso peggiore, un algoritmo classico deterministico deve eseguire

$$\boxed{2^{n-1} + 1}$$

query.

Vediamo perché.

Supponiamo di interrogare la funzione su

$$2^{n-1}$$

input distinti e di ottenere sempre lo stesso risultato, diciamo

$$0.$$

A questo punto sono ancora possibili due casi:

- la funzione è costante e vale 0 anche sui restanti 2^{n-1} input;
- la funzione è bilanciata e vale 1 su tutti i restanti 2^{n-1} input.

È quindi necessaria almeno un'altra interrogazione.

Dopo

$$2^{n-1} + 1$$

risultati uguali, invece, una funzione bilanciata è impossibile, perché non può assumere lo stesso valore su più della metà dei suoi input.

Esempio 8.3.2 (Costo classico per $n = 3$)**.** Per

$$n = 3$$

ci sono

$$2^3 = 8$$

input.
Supponiamo che le prime quattro interrogazioni restituiscano

$$0, 0, 0, 0.$$

Non possiamo ancora decidere. Potremmo avere una funzione costante,

$$(0,0,0,0,0,0,0,0),$$

oppure una funzione bilanciata,

$$(0,0,0,0,1,1,1,1),$$

dove l'ordine degli input non è rilevante per il ragionamento.
È necessaria una quinta *query*.
Quindi

$$2^{3-1}+1=5.$$

Per $n=10$, analogamente, il numero massimo di *query* richiesto è

$$2^9+1=513.$$

Registri utilizzati

L'algoritmo quantistico utilizza:

- un registro di n qubit, inizializzato in

$$|0\rangle^{\otimes n};$$

- un *ancilla qubit* inizializzato in

$$|1\rangle.$$

Lo stato iniziale è quindi

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

Si applica una Hadamard a tutti gli $n+1$ qubit.

Per il registro principale,

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

Per l'*ancilla qubit*,

$$H|1\rangle = |-\rangle.$$

Otteniamo pertanto

$$\boxed{|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle|-\rangle.}$$

Tutti gli input compaiono con la stessa ampiezza

$$\frac{1}{\sqrt{2^n}}.$$

Applicazione dell'*oracle*

Usando il *phase kickback*,

$$U_f|x\rangle|-\rangle = (-1)^{f(x)}|x\rangle|-\rangle,$$

si ottiene

$$\begin{aligned}|\psi_2\rangle &= U_f|\psi_1\rangle \\ &= \frac{1}{\sqrt{2^n}}\sum_x (-1)^{f(x)}|x\rangle|-\rangle.\end{aligned}$$

Quindi

$$\boxed{|\psi_2\rangle = \frac{1}{\sqrt{2^n}}\sum_x (-1)^{f(x)}|x\rangle|-\rangle.}$$

Ancora una volta, l'informazione rilevante non è stata scritta come una sequenza di bit accessibili direttamente. Essa è codificata nel pattern di segni

$$(-1)^{f(x)}.$$

L'azione di $H^{\otimes n}$

Per comprendere l'ultimo passo dell'algoritmo dobbiamo derivare una relazione importante.

Per un singolo bit

$$x_j \in \{0,1\},$$

possiamo scrivere

$$H|x_j\rangle = \frac{1}{\sqrt{2}}\sum_{y_j=0}^{1}(-1)^{x_j y_j}|y_j\rangle.$$

Verifichiamolo.

Se

$$x_j = 0,$$

allora

$$(-1)^{x_j y_j} = 1$$

per entrambi i valori di y_j, e quindi

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}.$$

Se

$$x_j = 1,$$

otteniamo

$$(-1)^{y_j},$$

e quindi

$$H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

Per

$$x = x_1 x_2 \cdots x_n$$

abbiamo

$$|x\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle.$$

Applicando una Hadamard a ciascun qubit,

$$H^{\otimes n}|x\rangle = \bigotimes_{j=1}^{n} \left[\frac{1}{\sqrt{2}} \sum_{y_j=0}^{1} (-1)^{x_j y_j} |y_j\rangle \right].$$

Raccogliendo tutti i fattori,

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

dove

$$x \cdot y = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n \pmod 2.$$

Il prodotto

$$x \cdot y$$

è quindi un prodotto scalare binario, calcolato modulo 2.

Esempio 8.3.3 (Prodotto scalare binario)**.** Consideriamo

$$x = 101, \qquad y = 110.$$

Allora

$$\begin{aligned} x \cdot y &= 1 \cdot 1 + 0 \cdot 1 + 1 \cdot 0 \pmod 2 \\ &= 1 \pmod 2 \\ &= 1. \end{aligned}$$

Pertanto

$$(-1)^{x \cdot y} = -1.$$

Se invece

$$y = 101,$$

si ha

$$x \cdot y = 1 \cdot 1 + 0 \cdot 0 + 1 \cdot 1 = 2 \equiv 0 \pmod 2,$$

per cui

$$(-1)^{x \cdot y} = 1.$$

Interferenza finale

Applichiamo ora $H^{\otimes n}$ al registro principale:

$$\begin{aligned}|\psi_3\rangle &= \frac{1}{\sqrt{2^n}} \sum_x (-1)^{f(x)} H^{\otimes n} |x\rangle \otimes |-\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_x (-1)^{f(x)} \left[\frac{1}{\sqrt{2^n}} \sum_y (-1)^{x \cdot y} |y\rangle \right] \otimes |-\rangle.\end{aligned}$$

Quindi

$$\boxed{|\psi_3\rangle = \frac{1}{2^n} \sum_y \left[\sum_x (-1)^{f(x)+x\cdot y} \right] |y\rangle \otimes |-\rangle.}$$

L'ampiezza associata a un particolare risultato y è

$$\boxed{\alpha_y = \frac{1}{2^n} \sum_x (-1)^{f(x)+x\cdot y}.}$$

L'algoritmo non ha bisogno di conoscere tutte queste ampiezze. È sufficiente considerare quella corrispondente a

$$y = 00\cdots0.$$

In questo caso

$$x \cdot 0 = 0$$

per ogni x, quindi

$$\boxed{\alpha_{0\cdots0} = \frac{1}{2^n} \sum_x (-1)^{f(x)}.}$$

Ora la *promise* sul tipo di funzione produce un'interferenza completamente diversa nei due casi.

Caso costante

Se

$$f(x) = 0$$

per ogni x, allora ogni termine vale

$$(-1)^{f(x)} = 1.$$

Ci sono 2^n termini, quindi

$$\alpha_{0\cdots0} = \frac{2^n}{2^n} = 1.$$

Se invece

$$f(x) = 1$$

per ogni x, tutti i termini valgono -1:

$$\alpha_{0\cdots0} = -\frac{2^n}{2^n} = -1.$$

In entrambi i casi,

$$|\alpha_{0\cdots0}|^2 = 1.$$

Pertanto, se la funzione è costante,

$$\boxed{\text{la misura restituisce } 00\cdots0 \text{ con probabilità 1.}}$$

Caso bilanciato

Se f è bilanciata, esattamente metà dei valori di x soddisfa

$$f(x) = 0$$

e metà soddisfa

$$f(x) = 1.$$

Nella somma

$$\sum_x (-1)^{f(x)}$$

abbiamo quindi

$$2^{n-1}$$

termini uguali a $+1$ e

$$2^{n-1}$$

termini uguali a -1.

Essi si cancellano:

$$\sum_x (-1)^{f(x)} = 2^{n-1} - 2^{n-1} = 0.$$

Quindi

$$\boxed{\alpha_{0\cdots0} = 0.}$$

Il risultato

$$00\cdots0$$

non può comparire.

Pertanto:

$$\boxed{00\cdots0 \Longrightarrow f \text{ costante}}$$

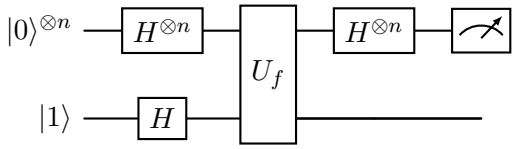
mentre

$$\boxed{y \neq 00\cdots0 \Longrightarrow f \text{ bilanciata.}}$$

Anche l'algoritmo di Deutsch–Jozsa è esatto nel modello ideale.

Circuito dell'algoritmo

In forma compatta, considerando la prima *wire* come un intero registro di n qubit, il circuito è



La linea superiore rappresenta schematicamente un registro di n qubit. La misura finale deve essere intesa come misura di tutti gli n qubit del registro nella base computazionale.

L'*oracle* compare una sola volta.

Esempio 8.3.4 (Funzione costante con $n = 3$)**.** Consideriamo

$$n = 3$$

e

$$f(x) = 1$$

per tutti gli otto input.
Dopo le Hadamard iniziali, il registro principale è

$$\frac{1}{\sqrt{8}} \sum_{x=000}^{111} |x\rangle.$$

L'*oracle* introduce la stessa fase -1 su ogni componente:

$$-\frac{1}{\sqrt{8}} \sum_{x} |x\rangle.$$

Questa è soltanto una fase globale:

$$-\left(H^{\otimes 3}|000\rangle\right).$$

Applicando nuovamente $H^{\otimes 3}$,

$$|\psi_{\text{out}}\rangle = -|000\rangle$$

sul registro principale.
La misura restituisce quindi

$$\boxed{000}$$

con probabilità uno.

Esempio 8.3.5 (Funzione bilanciata semplice con $n = 2$)**.** Consideriamo

$$n = 2$$

e definiamo

$$f(x_1 x_2) = x_1.$$

La tabella della funzione è

x	$f(x)$
00	0
01	0
10	1
11	1

La funzione è bilanciata.
Dopo le Hadamard iniziali, il registro principale si trova nello stato

$$|s\rangle = \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right).$$

L'*oracle* trasforma i valori della funzione in fasi:

$$|s\rangle \longmapsto \frac{1}{2}\left(|00\rangle + |01\rangle - |10\rangle - |11\rangle\right).$$

Lo stato può essere fattorizzato:

$$\begin{aligned}\frac{1}{2}\left(|00\rangle + |01\rangle - |10\rangle - |11\rangle\right) &= \frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} \\ &= |-\rangle|+\rangle.\end{aligned}$$

Applicando $H \otimes H$,

$$|-\rangle|+\rangle \longmapsto |1\rangle|0\rangle = |10\rangle.$$

La misura produce quindi

$$\boxed{10}$$

con probabilità uno.
Poiché

$$10 \neq 00,$$

l'algoritmo conclude correttamente che la funzione è bilanciata.

Esempio 8.3.6 (Una funzione bilanciata che non produce un unico output)**.** L'esempio precedente potrebbe suggerire erroneamente che una funzione bilanciata conduca sempre a un unico stato di base diverso da zero. Questo non è vero.
Consideriamo $n = 3$ e definiamo la funzione mediante la tabella

x	$f(x)$
000	1
001	1
010	1
011	0
100	1
101	0
110	0
111	0

La funzione è bilanciata perché assume quattro volte il valore 1 e quattro volte il valore 0.

Dopo l'*oracle*, il registro principale è

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

Dopo le Hadamard finali, l'ampiezza di ciascun $|y\rangle$ è

$$\alpha_y = \frac{1}{8}\sum_x (-1)^{f(x)+x\cdot y}.$$

Calcolando le otto somme si ottiene

y	$8\alpha_y$	α_y
000	0	0
001	-4	$-1/2$
010	-4	$-1/2$
011	0	0
100	-4	$-1/2$
101	0	0
110	0	0
111	4	$1/2$

Lo stato finale del registro principale è quindi

$$\boxed{|\psi_{\text{out}}\rangle = \frac{1}{2}\left(-|001\rangle - |010\rangle - |100\rangle + |111\rangle\right).}$$

Le possibili misure sono

$$001, \qquad 010, \qquad 100, \qquad 111,$$

ognuna con probabilità

$$\frac{1}{4}.$$

Il risultato non è deterministico come stringa specifica, ma è deterministica la proprietà che interessa all'algoritmo:

$$\boxed{\text{000 non può comparire.}}$$

Perciò una singola misura permette comunque di concludere con certezza che f è bilanciata.

Perché l'algoritmo funziona: interpretazione in termini di interferenza

L'algoritmo di Deutsch–Jozsa può essere letto come una trasformazione di informazione tra rappresentazioni diverse.

Dopo la prima Hadamard,

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

tutti gli input sono presenti con la stessa ampiezza.

Dopo l'*oracle*,

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

i valori della funzione sono codificati esclusivamente nelle fasi.

Le Hadamard finali fanno interferire tali ampiezze. Per lo stato

$$|0\rangle^{\otimes n}$$

tutte le componenti contribuiscono senza ulteriori segni, quindi la sua ampiezza è precisamente la media

$$\frac{1}{2^n}\sum_x (-1)^{f(x)}.$$

Se f è costante, tutti i contributi hanno lo stesso segno e interferiscono costruttivamente.

Se f è bilanciata, metà dei contributi è positiva e metà negativa: l'interferenza è completamente distruttiva per $|0\rangle^{\otimes n}$.

L'algoritmo non determina i singoli valori

$$f(x).$$

Determina direttamente una proprietà globale dell'intera funzione.

Confronto della complessità in *query*

Nel modello deterministico esatto:

$$\boxed{Q_{\text{classico}} = 2^{n-1} + 1}$$

nel caso peggiore, mentre

$$\boxed{Q_{\text{quantistico}} = 1.}$$

Il numero di *query* classiche cresce quindi esponenzialmente con n, mentre il numero di *query* quantistiche rimane costante.

Questa affermazione deve però essere interpretata correttamente.

Osservazione 8.3.7. Il confronto precedente riguarda algoritmi *deterministici ed esatti* nel modello a *oracle*. Se si ammette un algoritmo classico probabilistico con una piccola probabilità di errore, è possibile campionare alcuni input scelti casualmente e distinguere con alta probabilità una funzione costante da una bilanciata utilizzando molte meno di $2^{n-1} + 1$ *query*.

Il vantaggio esponenziale di Deutsch–Jozsa è quindi una separazione nella *exact* query *complexity* rispetto agli algoritmi classici deterministici, non una dimostrazione generale che ogni implementazione quantistica del problema sia esponenzialmente più veloce di qualunque procedura classica probabilistica.

Deutsch come caso particolare di Deutsch–Jozsa

Per

$$n = 1$$

il problema di Deutsch–Jozsa coincide esattamente con il problema di Deutsch.

Infatti il dominio contiene soltanto

$$0, \qquad 1.$$

Una funzione costante soddisfa

$$f(0) = f(1),$$

mentre una funzione bilanciata soddisfa

$$f(0) \neq f(1).$$

La condizione

$$\alpha_0 = \frac{1}{2}\left[(-1)^{f(0)} + (-1)^{f(1)}\right]$$

è precisamente quella derivata per l'algoritmo di Deutsch.

Deutsch–Jozsa non introduce quindi un principio computazionale completamente diverso: estende a n qubit lo stesso meccanismo di codifica nelle fasi e interferenza.

8.4 L'algoritmo di Bernstein–Vazirani

L'algoritmo di Bernstein–Vazirani costituisce un passo naturale dopo Deutsch e Deutsch–Jozsa. La struttura del circuito è quasi la stessa, ma cambia il tipo di informazione che vogliamo estrarre dalla funzione.

Nei problemi precedenti cercavamo una proprietà globale molto semplice: stabilire se una funzione fosse costante oppure bilanciata. Ora, invece, la funzione nasconde un'intera stringa binaria e il compito dell'algoritmo è ricostruirla.

Il problema

Consideriamo una stringa binaria sconosciuta

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

A questa stringa associamo la funzione

$$\boxed{f_s(x) = s \cdot x \pmod 2},$$

dove

$$x = x_1 x_2 \cdots x_n$$

e

$$s \cdot x = s_1 x_1 + s_2 x_2 + \cdots + s_n x_n \pmod 2.$$

L'*oracle* ci permette di interrogare la funzione f_s, ma la stringa s non è nota.

Il problema consiste nel determinarla.

Esempio 8.4.1 (Una funzione di Bernstein–Vazirani)**.** Supponiamo che

$$s = 101.$$

Allora

$$f_s(x_1x_2x_3) = x_1 + x_3 \pmod 2.$$

Per esempio,

$$f_s(000) = 0,$$
$$f_s(001) = 1,$$
$$f_s(010) = 0,$$
$$f_s(101) = 0,$$

perché

$$1 + 1 = 2 \equiv 0 \pmod 2.$$

La funzione dipende quindi dalla stringa nascosta $s = 101$, che specifica quali bit dell'input partecipano alla somma modulo 2.

Soluzione classica

Cominciamo, come nei casi precedenti, dalla strategia classica.

Per determinare il primo bit s_1 possiamo interrogare la funzione con l'input

$$e_1 = 100 \cdots 0.$$

Infatti,

$$f_s(e_1) = s \cdot e_1 = s_1.$$

Analogamente,

$$e_2 = 010 \cdots 0$$

permette di determinare

$$f_s(e_2) = s_2,$$

e così via.

Utilizzando i vettori della base computazionale,

$$e_1, e_2, \ldots, e_n,$$

otteniamo uno alla volta tutti i bit della stringa:

$$f_s(e_j) = s_j.$$

Sono quindi sufficienti

$$n$$

query.

Esempio 8.4.2 (Ricostruzione classica di $s = 101$)**.** Per

$$s = 101$$

effettuiamo tre interrogazioni.
Con

$$x = 100$$

otteniamo

$$f_s(100) = 1,$$

quindi

$$s_1 = 1.$$

Con

$$x = 010$$

otteniamo

$$f_s(010) = 0,$$

quindi

$$s_2 = 0.$$

Infine, con

$$x = 001$$

otteniamo

$$f_s(001) = 1,$$

quindi

$$s_3 = 1.$$

Dopo tre *query* abbiamo ricostruito

$$s = 101.$$

Nel modello classico deterministico esatto, n *query* sono anche necessarie nel caso peggiore. La stringa sconosciuta può infatti assumere

$$2^n$$

valori diversi, e ciascuna interrogazione classica restituisce un solo bit di informazione.

Il riferimento classico è pertanto

$$\boxed{Q_{\mathrm{C}} = n.}$$

La domanda quantistica è ora:

> È possibile ricostruire tutti gli n bit della stringa nascosta s utilizzando una sola *query* all'*oracle* quantistico?

L'idea quantistica

L'algoritmo utilizza lo stesso meccanismo fondamentale già incontrato in Deutsch e Deutsch–Jozsa.

Il registro di ingresso viene preparato in una sovrapposizione uniforme di tutti i possibili valori di x:

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

L'*ancilla qubit* viene invece preparato nello stato

$$|-\rangle.$$

Grazie al *phase kickback*,

$$U_f|x\rangle|-\rangle = (-1)^{f_s(x)}|x\rangle|-\rangle.$$

Poiché

$$f_s(x) = s \cdot x \pmod 2,$$

la fase introdotta dall'*oracle* è

$$(-1)^{s\cdot x}.$$

Dopo una sola *query*, quindi, il registro principale si trova nello stato

$$\boxed{\frac{1}{\sqrt{2^n}}\sum_x (-1)^{s\cdot x}|x\rangle.}$$

L'intera stringa s è ora codificata nella struttura delle fasi relative.

Il problema consiste nel trasformare questa informazione di fase in una stringa leggibile mediante misura.

Preparazione dello stato

L'algoritmo utilizza n qubit per il registro principale e un *ancilla qubit*.

Lo stato iniziale è

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

Applichiamo una Hadamard a tutti i qubit:

$$|\psi_1\rangle = \left(H^{\otimes n} \otimes H\right)|0\rangle^{\otimes n}|1\rangle.$$

Per il registro principale,

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

mentre

$$H|1\rangle = |-\rangle.$$

Pertanto

$$\boxed{|\psi_1\rangle = \frac{1}{\sqrt{2^n}}\sum_x |x\rangle|-\rangle.}$$

Tutti i possibili input x compaiono con la stessa ampiezza.

Applicazione dell'*oracle*

Applichiamo ora una sola volta U_f:

$$\begin{aligned}|\psi_2\rangle &= U_f|\psi_1\rangle \\ &= \frac{1}{\sqrt{2^n}}\sum_x U_f|x\rangle|-\rangle \\ &= \frac{1}{\sqrt{2^n}}\sum_x (-1)^{f_s(x)}|x\rangle|-\rangle \\ &= \frac{1}{\sqrt{2^n}}\sum_x (-1)^{s\cdot x}|x\rangle|-\rangle.\end{aligned}$$

L'*ancilla qubit* è ancora nello stato $|-\rangle$ e può essere separato:

$$\boxed{|\psi_2\rangle = \left[\frac{1}{\sqrt{2^n}}\sum_x (-1)^{s\cdot x}|x\rangle\right]\otimes|-\rangle.}$$

Il passaggio fondamentale è avvenuto: una sola *query* ha fatto dipendere la fase di ogni componente $|x\rangle$ dal prodotto scalare binario $s \cdot x$.

Come le fasi codificano i singoli bit di s

Per comprendere il passo successivo, scriviamo

$$x = x_1 x_2 \cdots x_n$$

e

$$s = s_1 s_2 \cdots s_n.$$

Poiché

$$s \cdot x = s_1 x_1 + s_2 x_2 + \cdots + s_n x_n \pmod 2,$$

si ha

$$(-1)^{s\cdot x} = \prod_{j=1}^{n} (-1)^{s_j x_j}.$$

La somma sul registro può quindi essere fattorizzata:

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

Esaminiamo il singolo fattore.

Se

$$s_j = 0,$$

allora

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

Se invece

$$s_j = 1,$$

si ottiene

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

Quindi ogni bit della stringa nascosta determina direttamente se il corrispondente qubit del registro si trovi in $|+\rangle$ oppure in $|-\rangle$:

$$\boxed{s_j = 0 \Longrightarrow |+\rangle, \qquad s_j = 1 \Longrightarrow |-\rangle.}$$

In forma compatta,

$$\boxed{\frac{1}{\sqrt{2^n}} \sum_x (-1)^{s\cdot x}|x\rangle = H^{\otimes n}|s\rangle.}$$

Questa relazione rende immediato l'ultimo passo dell'algoritmo.

Le Hadamard finali

Applichiamo nuovamente $H^{\otimes n}$ al registro principale:

$$H^{\otimes n}\left(H^{\otimes n}|s\rangle\right).$$

Poiché

$$H^2 = I,$$

si ha

$$H^{\otimes n}H^{\otimes n} = I^{\otimes n}.$$

Pertanto

$$\boxed{|\psi_{\text{out}}\rangle = |s\rangle}$$

sul registro principale.

La misura nella base computazionale restituisce quindi direttamente la stringa nascosta:

$$\boxed{s_1 s_2 \cdots s_n.}$$

La probabilità di successo, nel modello ideale, è

$$1.$$

Esempio 8.4.3 (Bernstein–Vazirani con $s = 101$)**.** Consideriamo nuovamente

$$s = 101.$$

La funzione è

$$f_s(x_1x_2x_3) = x_1 + x_3 \pmod 2.$$

La sua tabella è

x	$s \cdot x$	$f_s(x)$
000	0	0
001	1	1
010	0	0
011	1	1
100	1	1
101	0	0
110	1	1
111	0	0

Dopo le Hadamard iniziali, il registro principale è

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

Dopo una sola applicazione dell'*oracle*, le componenti per cui $f_s(x) = 1$ cambiano segno:

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

Questo stato può essere fattorizzato:

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

Quindi

$$|\psi_2\rangle = |-\rangle|+\rangle|-\rangle.$$

Le Hadamard finali producono

$$H|-\rangle \otimes H|+\rangle \otimes H|-\rangle = |1\rangle|0\rangle|1\rangle.$$

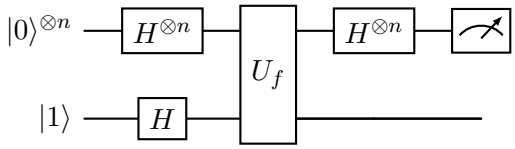
La misura restituisce pertanto

$$\boxed{101},$$

cioè esattamente la stringa nascosta s.

Circuito dell'algoritmo

In forma compatta, rappresentando il registro superiore come un registro di n qubit, il circuito è



La struttura è quasi identica a quella di Deutsch–Jozsa. La differenza non è nel numero di applicazioni dell'*oracle*, ma nell'informazione che viene estratta dalle fasi.

Confronto con la procedura classica

Il confronto nel modello deterministico esatto è

$$\boxed{Q_{\mathrm{C}} = n, \qquad Q_{\mathrm{Q}} = 1.}$$

Classicamente, la strategia naturale ricostruisce un bit di s per ogni *query*. L'algoritmo quantistico, invece, utilizza una sovrapposizione di tutti gli input e una sola applicazione dell'*oracle* per codificare simultaneamente la struttura lineare della funzione nelle fasi.

Le Hadamard finali non "leggono tutti i valori di f_s". Esse fanno interferire le ampiezze in modo che la fase associata a ciascun bit s_j venga trasformata direttamente nel valore computazionale 0 oppure 1.

Relazione con Deutsch–Jozsa

Bernstein–Vazirani è particolarmente istruttivo perché utilizza essenzialmente lo stesso schema circuitale di Deutsch–Jozsa:

$$H^{\otimes n} \longrightarrow U_f \longrightarrow H^{\otimes n}.$$

Cambia però il problema.

In Deutsch–Jozsa si vuole stabilire soltanto se una funzione sia costante oppure bilanciata.

In Bernstein–Vazirani la funzione possiede una struttura particolare,

$$f_s(x) = s \cdot x \pmod 2,$$

e si vuole ricostruire completamente il parametro nascosto s.

Se

$$s = 00\cdots0,$$

allora

$$f_s(x) = 0$$

per ogni x e la funzione è costante.

Se invece

$$s \neq 00\cdots0,$$

la funzione f_s è bilanciata.

Deutsch–Jozsa permetterebbe quindi di distinguere il caso $s = 0$ dal caso $s \neq 0$, ma non determinerebbe quale stringa non nulla sia presente. Bernstein–Vazirani sfrutta la struttura specifica della funzione per ottenere, con la stessa singola *query*, tutti gli n bit di s.

Il passaggio concettuale può essere riassunto così:

Deutsch–Jozsa	:	estrarre una proprietà globale di f,
Bernstein–Vazirani	:	estrarre un'intera stringa nascosta nelle fasi.

8.5 L'algoritmo di Grover

Gli algoritmi studiati finora sfruttano una struttura molto specifica della funzione interrogata dall'*oracle*. Deutsch distingue due classi di funzioni; Deutsch–Jozsa estende questa distinzione a n bit; Bernstein–Vazirani sfrutta invece la struttura lineare

$$f_s(x) = s \cdot x \pmod 2$$

per ricostruire una stringa nascosta.

L'algoritmo di Grover affronta un problema diverso: cercare un elemento che soddisfa una certa condizione quando non si dispone di alcuna struttura particolare che permetta di restringere la ricerca.

Il problema della ricerca non strutturata

Consideriamo un insieme di

$$N = 2^n$$

elementi, etichettati mediante stringhe binarie di n bit:

$$x \in \{0,1\}^n.$$

Supponiamo che uno solo di questi elementi, che indicheremo con

$$x^\star,$$

sia la soluzione cercata.

La funzione booleana

$$f : \{0,1\}^n \longrightarrow \{0,1\}$$

è definita in modo che

$$f(x) = \begin{cases} 1, & x = x^\star, \\ 0, & x \neq x^\star. \end{cases}$$

L'*oracle* permette quindi di verificare se un dato valore x sia oppure no la soluzione, ma non fornisce alcuna informazione aggiuntiva sulla posizione di $x^\star$.

Esempio 8.5.1 (Ricerca in un archivio non ordinato)**.** Supponiamo di avere un archivio contenente

$$N = 8$$

record, numerati da 0 a 7, e di sapere che uno solo di essi possiede una certa proprietà. Possiamo interrogare un record e ottenere una risposta binaria:

$$f(x) = \begin{cases} 1, & \text{se il record } x \text{ possiede la proprietà}, \\ 0, & \text{altrimenti}. \end{cases}$$

Se l'archivio non è ordinato e non possiede una struttura utile, il risultato

$$f(2) = 0$$

non fornisce alcuna indicazione su quale degli altri sette record possa essere quello corretto. Il problema è quindi una ricerca *non strutturata*.

Soluzione classica

Classicamente, in assenza di struttura, non esiste una strategia che permetta di eliminare una frazione significativa degli elementi dopo una singola interrogazione.

Nel caso peggiore un algoritmo deterministico deve esaminare un numero di elementi proporzionale a N. Se è noto che esiste esattamente una soluzione, dopo aver escluso $N-1$ elementi l'ultimo può essere identificato per eliminazione.

La complessità classica nel modello a *query* è quindi

$$\boxed{Q_{\mathrm{C}} = \Theta(N).}$$

Nel modello elementare di ricerca sequenziale, se la posizione dell'elemento marcato è uniformemente distribuita, il numero medio di interrogazioni necessarie per incontrarlo è

$$\boxed{\frac{N+1}{2}.}$$

Esempio 8.5.2 (Numero medio di interrogazioni classiche per $N = 1000$)**.** Con

$$N = 1000,$$

si ottiene

$$\frac{1000+1}{2} = 500.5.$$

Nel caso peggiore la ricerca sequenziale può richiedere N interrogazioni. Se si sfrutta anche l'informazione che esiste esattamente una soluzione, l'ultimo elemento può naturalmente essere identificato per eliminazione dopo aver escluso gli altri; questa precisazione modifica soltanto una costante e non la legge di scala

$$\Theta(N).$$

Il problema che Grover risolve è quindi molto semplice da formulare ma, classicamente, non consente un'accelerazione sostanziale senza informazioni aggiuntive.

L'idea quantistica

La prima possibilità che viene spontaneo considerare è preparare una sovrapposizione uniforme di tutti gli elementi:

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle.$$

In questo stato ogni elemento possiede la stessa ampiezza

$$\frac{1}{\sqrt{N}}$$

e quindi, se misurassimo immediatamente, la probabilità di ottenere la soluzione sarebbe soltanto

$$\frac{1}{N}.$$

La sola sovrapposizione non è dunque sufficiente.

L'idea di Grover è modificare ripetutamente le ampiezze in modo che l'ampiezza associata allo stato soluzione aumenti, mentre quella associata agli stati non soluzione diminuisca.

Il meccanismo è una forma di

$$\boxed{\textit{amplitude amplification.}}$$

Ogni iterazione di Grover è costituita da due passaggi fondamentali:

1. l'*oracle* modifica la fase dello stato soluzione;
2. un operatore di diffusione trasforma questa differenza di fase in un aumento dell'ampiezza della soluzione.

Dopo un numero opportuno di iterazioni, la probabilità di misurare $x^\star$ diventa prossima a 1.

L'*oracle* di fase

L'*oracle* standard può essere scritto come

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

Come abbiamo già visto negli algoritmi precedenti, preparando l'*ancilla qubit* nello stato

$$|-\rangle,$$

il *phase kickback* produce

$$U_f|x\rangle|-\rangle = (-1)^{f(x)}|x\rangle|-\rangle.$$

Poiché in Grover

$$f(x^\star) = 1$$

e

$$f(x) = 0 \qquad \text{per } x \neq x^\star,$$

l'effetto sul registro principale è

$$\boxed{|x\rangle \longmapsto (-1)^{f(x)}|x\rangle.}$$

Possiamo quindi descrivere l'*oracle* mediante l'operatore

$$O_f|x\rangle = (-1)^{f(x)}|x\rangle.$$

Esso lascia invariati tutti gli stati non soluzione e cambia soltanto il segno dello stato marcato:

$$O_f|x^\star\rangle = -|x^\star\rangle.$$

Esempio 8.5.3 (Effetto dell'*oracle* per $N = 4$)**.** Consideriamo due qubit, quindi

$$N = 4,$$

e supponiamo che la soluzione sia

$$x^\star = 10.$$

Lo stato uniforme è

$$|s\rangle = \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right).$$

Dopo l'*oracle*,

$$O_f|s\rangle = \frac{1}{2}\left(|00\rangle + |01\rangle - |10\rangle + |11\rangle\right).$$

Le probabilità di misura sono però ancora

$$\frac{1}{4}$$

per tutti e quattro gli stati, perché il cambio di segno non modifica il modulo delle ampiezze. L'*oracle*, da solo, non aumenta quindi la probabilità di trovare la soluzione.

Questo punto è essenziale: il ruolo dell'*oracle* è soltanto *marcare* la soluzione mediante una fase. Serve una seconda trasformazione che converta tale informazione di fase in una variazione dei moduli delle ampiezze.

L'operatore di diffusione

Indichiamo con

$$|s\rangle = \frac{1}{\sqrt{N}}\sum_x |x\rangle$$

lo stato uniforme.

L'operatore di diffusione di Grover è

$$\boxed{D = 2|s\rangle\langle s| - I.}$$

Per capire il significato di questa formula consideriamo uno stato generico

$$|\psi\rangle = \sum_{x=0}^{N-1} a_x|x\rangle.$$

Definiamo la media delle ampiezze

$$\bar{a} = \frac{1}{N}\sum_{x=0}^{N-1} a_x.$$

Poiché

$$\langle s|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x a_x = \sqrt{N}\,\bar{a},$$

si ha

$$2|s\rangle\langle s|\psi\rangle = 2\bar{a} \sum_x |x\rangle.$$

Applicando D:

$$\begin{aligned} D|\psi\rangle &= (2|s\rangle\langle s| - I)\,|\psi\rangle \\ &= \sum_x (2\bar{a} - a_x)\,|x\rangle. \end{aligned}$$

Quindi ogni ampiezza viene trasformata secondo

$$\boxed{a_x \longmapsto 2\bar{a} - a_x.}$$

Questa trasformazione viene spesso descritta come *inversion about the mean*: ogni ampiezza viene riflessa rispetto alla media delle ampiezze.

Un'iterazione completa per $N = 4$

Il caso $N = 4$ è particolarmente utile perché una sola iterazione di Grover porta esattamente alla soluzione.

Partiamo dallo stato uniforme

$$|s\rangle = \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right)$$

e supponiamo ancora che

$$x^\star = 10.$$

Dopo l'*oracle*,

$$|\psi'\rangle = \frac{1}{2}\left(|00\rangle + |01\rangle - |10\rangle + |11\rangle\right).$$

Le ampiezze sono quindi

$$\left(\frac{1}{2}, \frac{1}{2}, -\frac{1}{2}, \frac{1}{2}\right).$$

La loro media è

$$\bar{a} = \frac{\frac{1}{2} + \frac{1}{2} - \frac{1}{2} + \frac{1}{2}}{4} = \frac{1}{4}.$$

L'operatore di diffusione produce

$$a'_x = 2\bar{a} - a_x.$$

Per i tre stati non marcati,

$$\frac{1}{2} \longmapsto 2 \cdot \frac{1}{4} - \frac{1}{2} = 0.$$

Per lo stato soluzione,

$$-\frac{1}{2} \longmapsto 2 \cdot \frac{1}{4} - \left(-\frac{1}{2}\right) = 1.$$

Lo stato finale diventa quindi

$$\boxed{|\psi_{\text{out}}\rangle = |10\rangle.}$$

La misura restituisce la soluzione con probabilità

$$1.$$

Esempio 8.5.4 (Grover con due qubit)**.** Consideriamo l'insieme

$$\{00, 01, 10, 11\}$$

e supponiamo che l'*oracle* marchi

$$01.$$

Lo stato uniforme iniziale è

$$|s\rangle = \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right).$$

Dopo l'*oracle*,

$$\frac{1}{2}\left(|00\rangle - |01\rangle + |10\rangle + |11\rangle\right).$$

La media delle ampiezze è ancora

$$\bar{a} = \frac{1}{4}.$$

Applicando l'inversione rispetto alla media,

$$\frac{1}{2} \mapsto 0$$

per gli stati non marcati e

$$-\frac{1}{2} \mapsto 1$$

per lo stato marcato.
Si ottiene

$$|01\rangle.$$

Con una sola iterazione, seguita dalla misura, la soluzione viene quindi trovata con certezza.

Esempio 8.5.5 (Amplificazione progressiva con $N = 10$)**.** Il caso $N = 4$ è molto elegante, ma è anche eccezionale: una sola iterazione porta esattamente allo stato soluzione. Per mostrare numericamente il carattere progressivo dell'amplificazione senza vincolarci alla dimensione di un registro completo di qubit, consideriamo un modello astratto con

$$N = 10$$

stati e supponiamo che esista una sola soluzione, per esempio l'elemento

$$x^\star = 7.$$

In un circuito basato su qubit questo caso richiederebbe un'immersione in uno spazio di dimensione 16 e un trattamento esplicito dei sei stati inutilizzati.

Nello stato uniforme iniziale ogni ampiezza vale

$$a_0 = \frac{1}{\sqrt{10}} \approx 0.31623.$$

La probabilità iniziale di misurare direttamente lo stato marcato è quindi soltanto

$$P_0 = \frac{1}{10} = 0.1.$$

Prima iterazione.
Dopo l'*oracle*, l'ampiezza dello stato marcato cambia segno:

$$a_m = -0.31623,$$

mentre le nove ampiezze non marcate restano

$$a_u = 0.31623.$$

La media delle dieci ampiezze è

$$\bar{a} = \frac{9(0.31623) - 0.31623}{10} \approx 0.25298.$$

Dopo l'operatore di diffusione,

$$a'_u = 2\bar{a} - a_u \approx 0.18974,$$

mentre per lo stato marcato

$$a'_m = 2\bar{a} - a_m \approx 0.82219.$$

La probabilità di ottenere la soluzione è quindi

$$P_1 = |a'_m|^2 \approx (0.82219)^2 = 0.676.$$

Siamo passati da

$$10\%$$

a circa

$$67.6\%$$

dopo una sola iterazione.

Seconda iterazione.
La seconda applicazione dell'*oracle* cambia nuovamente il segno dell'ampiezza marcata:

$$a_m = -0.82219, \qquad a_u = 0.18974.$$

La nuova media è

$$\bar{a} = \frac{9(0.18974) - 0.82219}{10} \approx 0.08855.$$

Dopo la diffusione,

$$a'_u \approx -0.01264,$$

mentre

$$a'_m \approx 0.99928.$$

La probabilità dello stato marcato diventa

$$\boxed{P_2 \approx (0.99928)^2 \approx 0.99856.}$$

Dopo soltanto due iterazioni, quindi, la probabilità di ottenere la soluzione è circa

$$99.856\%.$$

Questo esempio rende visibile il meccanismo di Grover:

$$\boxed{0.10 \longrightarrow 0.676 \longrightarrow 0.99856.}$$

L'algoritmo non crea immediatamente la soluzione. Trasferisce progressivamente ampiezza dagli stati non marcati allo stato marcato mediante la successione

$$\text{marcatura di fase} \longrightarrow \text{inversione rispetto alla media.}$$

Un primo confronto quantitativo

Prima di derivare geometricamente il numero ottimale di iterazioni, anticipiamo il risultato: quando esiste una sola soluzione e N è grande,

$$k_{\text{opt}} \approx \frac{\pi}{4}\sqrt{N}.$$

Esempio 8.5.6 (Confronto quantitativo per $N = 1000$)**.** Per

$$N = 1000$$

si ottiene

$$k_{\text{opt}} \approx \frac{\pi}{4}\sqrt{1000} \approx 24.8.$$

Possiamo quindi confrontare direttamente i numeri:

metodo	numero caratteristico di interrogazioni
ricerca classica sequenziale, valore medio	500.5
ricerca classica, caso peggiore	1000
Grover	circa 25

Il vantaggio non consiste in una riduzione di un semplice fattore costante. Al crescere di N, il costo classico cresce linearmente,

$$N,$$

mentre quello quantistico cresce come

$$\sqrt{N}.$$

La sezione geometrica seguente spiegherà da dove proviene precisamente questa legge di scala.

L'iterazione di Grover

Definiamo l'operatore

$$\boxed{G = DO_f.}$$

Un'applicazione di G costituisce un'iterazione di Grover:

$$\boxed{\textit{oracle} \text{ di fase} \longrightarrow \text{operatore di diffusione.}}$$

L'algoritmo completo può essere descritto schematicamente come segue:

1. si prepara
$$|0\rangle^{\otimes n};$$
2. si applica
$$H^{\otimes n}$$
per ottenere lo stato uniforme $|s\rangle$;
3. si applica più volte l'iterazione
$$G = DO_f;$$
4. si misura il registro nella base computazionale.

A differenza di Deutsch, Deutsch–Jozsa e Bernstein–Vazirani, in generale non è sufficiente una sola applicazione dell'*oracle*. Il vantaggio di Grover nasce dal fatto che il numero ottimale di iterazioni cresce come la radice quadrata del numero di elementi.

Interpretazione geometrica

Per capire perché l'algoritmo richieda circa $\sqrt{N}$ iterazioni, è utile osservare che l'intera evoluzione rimane in un sottospazio bidimensionale.

Indichiamo con

$$|w\rangle = |x^\star\rangle$$

lo stato soluzione e definiamo lo stato normalizzato che rappresenta la sovrapposizione uniforme degli elementi non soluzione:

$$|r\rangle = \frac{1}{\sqrt{N-1}} \sum_{x \neq x^\star} |x\rangle.$$

Lo stato uniforme iniziale può essere scritto come

$$|s\rangle = \frac{1}{\sqrt{N}}|w\rangle + \sqrt{\frac{N-1}{N}}|r\rangle.$$

Introduciamo un angolo θ tale che

$$\sin\theta = \frac{1}{\sqrt{N}}, \qquad \cos\theta = \sqrt{\frac{N-1}{N}}.$$

Allora

$$\boxed{|s\rangle = \sin\theta\,|w\rangle + \cos\theta\,|r\rangle.}$$

L'*oracle* e l'operatore di diffusione agiscono come due riflessioni nel piano generato da $|w\rangle$ e $|r\rangle$. La composizione di due riflessioni è una rotazione.

Una iterazione di Grover ruota quindi lo stato di un angolo

$$2\theta$$

verso lo stato soluzione.

Dopo k iterazioni,

$$\boxed{G^k|s\rangle = \sin\big((2k+1)\theta\big)|w\rangle + \cos\big((2k+1)\theta\big)|r\rangle.}$$

La probabilità di misurare la soluzione è pertanto

$$\boxed{P_k = \sin^2\big((2k+1)\theta\big).}$$

Quante iterazioni sono necessarie?

Vogliamo scegliere k in modo che

$$(2k+1)\theta$$

sia il più vicino possibile a

$$\frac{\pi}{2}.$$

Per N grande,

$$\sin\theta = \frac{1}{\sqrt{N}}$$

implica

$$\theta \simeq \frac{1}{\sqrt{N}}.$$

Di conseguenza,

$$k \simeq \frac{\pi}{4}\sqrt{N}.$$

Più precisamente, si sceglie l'intero più vicino al valore ottimale compatibile con la rotazione.

La complessità quantistica nel modello a *query* è quindi

$$\boxed{Q_{\mathrm{Q}} = \Theta(\sqrt{N}).}$$

Il confronto fondamentale è

$$\boxed{Q_{\mathrm{C}} = \Theta(N), \qquad Q_{\mathrm{Q}} = \Theta(\sqrt{N}).}$$

Grover fornisce quindi un vantaggio *quadratico*, non esponenziale.

Perché non bisogna iterare troppo

L'amplificazione non è monotona indefinitamente.

Dalla relazione

$$P_k = \sin^2((2k+1)\theta)$$

si vede che, dopo aver raggiunto un valore vicino a 1, ulteriori iterazioni fanno ruotare lo stato oltre la direzione $|w\rangle$ e la probabilità della soluzione torna a diminuire.

Non conviene quindi applicare l'iterazione di Grover "il maggior numero di volte possibile". Il numero di iterazioni deve essere scelto in funzione di N e del numero di soluzioni marcate.

Nel caso trattato qui abbiamo assunto, per semplicità,

$$M = 1.$$

Se esistono M soluzioni, la stessa analisi si generalizza sostituendo

$$\sin^2\theta = \frac{1}{N}$$

con

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

e il numero di iterazioni diventa dell'ordine di

$$\sqrt{\frac{N}{M}}.$$

Applicazione alla ricerca esaustiva di chiavi

Una delle conseguenze più immediate dell'accelerazione quadratica riguarda la ricerca esaustiva in uno spazio di chiavi.

Supponiamo che una chiave segreta sia rappresentata da n bit. Lo spazio delle possibili chiavi ha dimensione

$$N = 2^n.$$

Una ricerca esaustiva classica richiede un numero di tentativi dell'ordine di

$$2^n.$$

Se è disponibile un *oracle* quantistico capace di riconoscere la chiave corretta, Grover riduce il numero di *query* all'ordine di

$$\sqrt{2^n} = 2^{n/2}.$$

Nel modello ideale a *query* si ha quindi

$$\boxed{2^n \longrightarrow 2^{n/2}.}$$

Esempio 8.5.7 (Ricerca esaustiva e lunghezza delle chiavi)**.** Per AES-128,

$$\text{AES-128}: \qquad 2^{128} \longrightarrow 2^{64},$$

mentre per AES-256,

$$\text{AES-256}: \qquad 2^{256} \longrightarrow 2^{128}.$$

Questi numeri vanno interpretati correttamente. Non significano che un attacco quantistico reale richieda semplicemente 2^{64} o 2^{128} porte elementari. Una singola *query* deve implementare reversibilmente il controllo della chiave, e il circuito completo richiede inoltre diffusione, qubit di lavoro, correzione degli errori e controllo classico.

Il risultato di Grover riguarda anzitutto la legge di scala nel modello a *query*: una ricerca esaustiva su uno spazio di dimensione 2^n vede, idealmente, dimezzarsi l'esponente,

$$n \longrightarrow \frac{n}{2}.$$

Questo esempio mostra perché un vantaggio quadratico può comunque avere conseguenze rilevanti quando lo spazio di ricerca è esponenzialmente grande.

Dal numero di *query* al vantaggio computazionale

Il confronto

$$\Theta(N) \qquad \text{contro} \qquad \Theta(\sqrt{N})$$

dimostra una riduzione asintotica del numero di interrogazioni necessarie per la ricerca non strutturata.

È però importante distinguere il numero di *query* dal tempo fisico di esecuzione. Una *query* quantistica può essere molto più costosa di una semplice operazione classica, e l'operatore di diffusione richiede ulteriori porte.

Perciò Grover non implica che, per ogni valore finito di N, un dispositivo quantistico reale sia automaticamente più veloce di uno classico. Il vantaggio algoritmico emerge nella diversa legge di scala:

$$N \quad \text{contro} \quad \sqrt{N}.$$

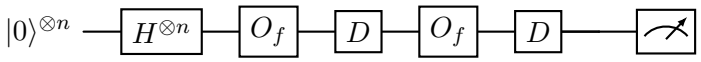
Questo è precisamente il tipo di confronto che motiva lo studio della complessità quantistica: non chiediamo soltanto se un problema sia calcolabile, ma come crescano le risorse necessarie quando aumenta la dimensione dell'input.

Circuito schematico

L'operatore di diffusione può essere riscritto come

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

Il circuito completo può quindi essere rappresentato schematicamente come



dove la coppia

$$O_f \longrightarrow D$$

viene ripetuta il numero di volte necessario.

La figura è schematica: sia l'*oracle* sia l'operatore di diffusione devono naturalmente essere decomposti in porte elementari quando si realizza concretamente il circuito.

Che cosa distingue Grover dagli algoritmi precedenti

Deutsch, Deutsch–Jozsa e Bernstein–Vazirani utilizzano una singola *query* per trasformare una struttura globale della funzione in un pattern di fasi che viene decodificato mediante Hadamard.

Grover utilizza invece una strategia iterativa:

$$\boxed{\text{marcatura di fase} \longrightarrow \text{diffusione} \longrightarrow \text{amplificazione dell'ampiezza.}}$$

L'*oracle* non rivela la soluzione. La marca soltanto con una fase. La diffusione converte ripetutamente questa differenza di fase in un aumento dell'ampiezza dello stato soluzione.

Il vantaggio quantistico nasce quindi da un uso controllato e ripetuto dell'interferenza, che concentra progressivamente la probabilità sul risultato desiderato.

8.6 L'algoritmo di Simon

L'algoritmo di Simon introduce un'idea diversa da quella utilizzata in Grover. Non si cerca un elemento marcato in uno spazio privo di struttura; si cerca invece di ricostruire una *struttura nascosta* della funzione.

Il problema è particolarmente importante dal punto di vista concettuale perché mostra una separazione esponenziale, nel modello a *query*, tra la procedura quantistica e quella classica. Inoltre introduce un tema che ritroveremo in forma più sofisticata nell'algoritmo di Shor: l'estrazione di una periodicità nascosta.

Il problema

Consideriamo una funzione

$$f : \{0,1\}^n \longrightarrow \{0,1\}^n.$$

Supponiamo che esista una stringa binaria sconosciuta

$$s \in \{0,1\}^n, \qquad s \neq 0^n,$$

tale che

$$\boxed{f(x) = f(y) \quad \Longleftrightarrow \quad y = x \oplus s.}$$

Qui $\oplus$ indica lo XOR bit a bit.

La condizione precedente è la *promise* del problema. Essa implica che ogni valore assunto da f possiede esattamente due controimmagini:

$$x \qquad \text{e} \qquad x \oplus s.$$

La funzione è quindi due-a-uno e gli input sono raggruppati in coppie separate sempre dalla stessa stringa nascosta s.

Il problema di Simon consiste nel determinare s.

Esempio 8.6.1 (Una funzione con periodo nascosto)**.** Prendiamo

$$n = 3, \qquad s = 101.$$

Le coppie di input collegate dalla trasformazione

$$x \longmapsto x \oplus 101$$

sono

$$000 \leftrightarrow 101,$$

$$001 \leftrightarrow 100,$$

$$010 \leftrightarrow 111,$$

$$011 \leftrightarrow 110.$$

Una possibile funzione compatibile con la *promise* è

x	$f(x)$
000	000
001	001
010	010
011	011
100	001
101	000
110	011
111	010

Per esempio,

$$f(001) = f(100),$$

e infatti

$$001 \oplus 100 = 101 = s.$$

Analogamente,

$$f(010) = f(111),$$

e

$$010 \oplus 111 = 101.$$

La stringa s descrive quindi la stessa relazione fra tutte le coppie di input che producono la

stessa uscita.

Come si risolverebbe classicamente

Se riusciamo a trovare due input distinti x e y tali che

$$f(x) = f(y),$$

la *promise* ci permette di ricavare immediatamente

$$\boxed{s = x \oplus y.}$$

Il problema classico consiste dunque nel *collision finding* (ricerca di collisioni), cioè nel trovare due input distinti che producano la stessa uscita.

Se interroghiamo l'*oracle* su pochi input, però, è molto probabile che i valori ottenuti siano tutti diversi. In assenza di una collisione non sappiamo ancora quali siano le coppie

$$\{x, x \oplus s\}.$$

Nel *collision finding*, per ottenere una collisione con probabilità significativa mediante interrogazioni casuali occorre, per il cosiddetto effetto del compleanno, un numero di *query* dell'ordine di

$$\boxed{2^{n/2}.}$$

Nel modello classico randomizzato la complessità in *query* è quindi

$$\boxed{Q_{\mathrm{C}} = \Theta\left(2^{n/2}\right).}$$

Un algoritmo deterministico che voglia essere certo di incontrare una coppia può richiedere invece un numero di interrogazioni dell'ordine di

$$2^n.$$

Esempio 8.6.2 (Perché serve una collisione)**.** Supponiamo ancora che

$$s = 101.$$

Se interroghiamo

$$x = 000$$

e otteniamo

$$f(000) = 000,$$

questo singolo risultato non ci dice quale sia il secondo input associato allo stesso valore. Se successivamente troviamo

$$f(101) = 000,$$

abbiamo una collisione e possiamo calcolare

$$s = 000 \oplus 101 = 101.$$

Classicamente l'informazione decisiva compare dunque nel momento in cui due interrogazioni raggiungono la stessa coppia nascosta.

L'idea quantistica

L'algoritmo quantistico non tenta di cercare direttamente una collisione.

Produce invece, a ogni esecuzione, una stringa

$$y \in \{0,1\}^n$$

che soddisfa una relazione lineare con la stringa nascosta:

$$\boxed{y \cdot s = 0 \pmod 2.}$$

Ripetendo l'algoritmo otteniamo più equazioni lineari modulo 2. Quando abbiamo raccolto un numero sufficiente di equazioni indipendenti, possiamo ricostruire s mediante algebra lineare classica.

La strategia è quindi

$$\boxed{\textit{oracle} \longrightarrow \text{interferenza} \longrightarrow \text{equazioni su } s \longrightarrow \text{soluzione classica del sistema.}}$$

È importante osservare che una singola esecuzione non restituisce direttamente s. Il vantaggio quantistico consiste nel produrre efficientemente vincoli lineari che sarebbero difficili da ottenere classicamente senza trovare collisioni.

Preparazione dello stato

Utilizziamo due registri di n qubit.

Il primo registro contiene l'input della funzione; il secondo conterrà il valore di $f(x)$.

Lo stato iniziale è

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

Applichiamo

$$H^{\otimes n}$$

al primo registro:

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle|0^n\rangle.$$

Il primo registro è ora una sovrapposizione uniforme di tutti i

$$2^n$$

possibili input.

Applicazione dell'*oracle*

L'*oracle* implementa reversibilmente la funzione:

$$U_f|x\rangle|z\rangle = |x\rangle|z \oplus f(x)\rangle.$$

Poiché il secondo registro parte da $|0^n\rangle$,

$$U_f|x\rangle|0^n\rangle = |x\rangle|f(x)\rangle.$$

Dopo una sola *query*,

$$\boxed{|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_x |x\rangle|f(x)\rangle.}$$

A causa della *promise*, ogni valore $f(x)$ compare esattamente due volte: una volta associato a x e una volta associato a $x \oplus s$.

Misura del secondo registro

Supponiamo di misurare il secondo registro e di ottenere un certo valore

$$f(x_0).$$

Non possiamo sapere se l'input corrispondente fosse

$$x_0$$

oppure

$$x_0 \oplus s,$$

perché entrambi producono la stessa uscita.

Dopo la misura, il primo registro collassa quindi nello stato

$$\boxed{|\psi_3\rangle = \frac{|x_0\rangle + |x_0 \oplus s\rangle}{\sqrt{2}}.}$$

Questo passaggio è fondamentale: la struttura nascosta s è ora contenuta nella relazione tra le due componenti della sovrapposizione.

Esempio 8.6.3 (Collasso su una coppia)**.** Nel caso

$$s = 101,$$

supponiamo che la misura del secondo registro restituisca

$$010.$$

Dalla tabella dell'esempio precedente sappiamo che

$$f(010) = f(111) = 010.$$

Il primo registro viene quindi proiettato nello stato

$$|\psi_3\rangle = \frac{|010\rangle + |111\rangle}{\sqrt{2}}.$$

Osserviamo che

$$010 \oplus 111 = 101 = s.$$

La misura non ci ha rivelato quale dei due input fosse presente: ha invece preparato una sovrapposizione coerente dei due input separati dal periodo nascosto.

Le Hadamard finali

Applichiamo ora

$$H^{\otimes n}$$

al primo registro.

Per una generica stringa x vale

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

dove

$$x \cdot y = x_1 y_1 \oplus x_2 y_2 \oplus \cdots \oplus x_n y_n$$

è il prodotto scalare binario.

Pertanto

$$\begin{aligned} H^{\otimes n}|\psi_3\rangle &= \frac{1}{\sqrt{2}} H^{\otimes n} \left(|x_0\rangle + |x_0 \oplus s\rangle\right) \\ &= \frac{1}{\sqrt{2^{n+1}}} \sum_y \left[(-1)^{x_0\cdot y} + (-1)^{(x_0\oplus s)\cdot y}\right] |y\rangle. \end{aligned}$$

Usando

$$(x_0 \oplus s) \cdot y = x_0 \cdot y \oplus s \cdot y,$$

possiamo raccogliere

$$(-1)^{x_0\cdot y}:$$

$$\boxed{H^{\otimes n}|\psi_3\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_y (-1)^{x_0\cdot y} \left[1 + (-1)^{s\cdot y}\right] |y\rangle.}$$

Ora compaiono due casi.

Se

$$s \cdot y = 0,$$

allora

$$1 + (-1)^{s \cdot y} = 1 + 1 = 2.$$

Se invece

$$s \cdot y = 1,$$

allora

$$1 + (-1)^{s \cdot y} = 1 - 1 = 0.$$

Per interferenza distruttiva scompaiono quindi tutte le componenti che non soddisfano

$$s \cdot y = 0.$$

La misura del primo registro può restituire soltanto stringhe tali che

$$\boxed{s \cdot y = 0 \pmod 2.}$$

Che cosa produce una singola esecuzione

Una singola esecuzione dell'algoritmo produce dunque un valore y scelto uniformemente tra le stringhe ortogonali a s nel senso del prodotto scalare modulo 2.

L'insieme

$$s^{\perp} = \{y \in \{0,1\}^n : s \cdot y = 0\}$$

contiene

$$2^{n-1}$$

stringhe.

Ogni esecuzione fornisce quindi un'equazione lineare

$$y_1 s_1 \oplus y_2 s_2 \oplus \cdots \oplus y_n s_n = 0.$$

Ripetendo il circuito raccogliamo equazioni diverse. Quando ne abbiamo ottenute

$$n - 1$$

linearmente indipendenti, il sistema omogeneo possiede uno spazio delle soluzioni unidimensionale:

$$\{0^n, s\}.$$

Poiché la *promise* esclude

$$s = 0^n,$$

rimane un'unica possibilità non nulla.

Esempio 8.6.4 (Ricostruzione di $s = 101$)**.** Per

$$s = 101$$

la condizione

$$s \cdot y = 0$$

diventa

$$y_1 \oplus y_3 = 0.$$

Le stringhe ammesse sono quindi

$$000, \qquad 010, \qquad 101, \qquad 111.$$

Supponiamo che due esecuzioni dell'algoritmo restituiscano

$$y^{(1)} = 010$$

e

$$y^{(2)} = 101.$$

La prima misura produce l'equazione

$$010 \cdot s = 0,$$

cioè

$$s_2 = 0.$$

La seconda produce

$$101 \cdot s = 0,$$

cioè

$$s_1 \oplus s_3 = 0.$$

Le soluzioni sono

$$s = 000$$

oppure

$$s = 101.$$

La *promise*

$$s \neq 000$$

elimina la prima possibilità e otteniamo

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

Quante *query* sono necessarie?

Ogni esecuzione del circuito utilizza una sola *query* all'*oracle* e produce una relazione del tipo

$$s \cdot y = 0.$$

Non tutte le relazioni ottenute sono necessariamente nuove: possiamo misurare nuovamente una stringa già incontrata oppure una combinazione lineare delle precedenti.

Tuttavia, dopo un numero di esecuzioni dell'ordine di

$$n,$$

si ottengono con alta probabilità

$$n-1$$

equazioni indipendenti.

La complessità quantistica in *query* è quindi

$$\boxed{Q_{\mathrm{Q}} = O(n).}$$

Il confronto con la strategia classica randomizzata è

$$\boxed{Q_{\mathrm{C}} = \Theta\left(2^{n/2}\right), \qquad Q_{\mathrm{Q}} = O(n).}$$

Si tratta di una separazione esponenziale nel modello a *query*.

Un confronto numerico

Per rendere visibile la differenza di scala, consideriamo

$$n = 20.$$

Il dominio contiene

$$N = 2^{20} = 1\,048\,576$$

input.

Classicamente, il *collision finding* richiede un numero di interrogazioni dell'ordine di

$$\sqrt{N} = 2^{10} = 1024.$$

L'algoritmo di Simon richiede invece un numero di esecuzioni dell'ordine di

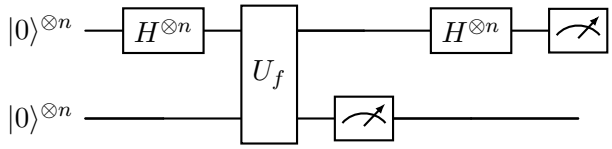
$$n = 20,$$

più il costo classico, polinomiale, necessario per risolvere il sistema lineare modulo 2.

Il confronto non va interpretato come una previsione diretta del tempo di esecuzione di dispositivi reali: riguarda la complessità nel modello a *query*. Mostra però che le due procedure possiedono leggi di scala radicalmente diverse.

Circuito schematico

Il circuito di una singola esecuzione può essere rappresentato schematicamente come



La misura del secondo registro seleziona una coppia

$$|x_0\rangle, \qquad |x_0 \oplus s\rangle,$$

mentre la Hadamard finale sul primo registro trasforma la relazione fra i due input in un vincolo lineare

$$s \cdot y = 0.$$

Il circuito deve essere ripetuto più volte per raccogliere un numero sufficiente di equazioni indipendenti.

Perché l'algoritmo funziona

Il punto centrale può essere riassunto nel passaggio

$$\frac{|x_0\rangle + |x_0 \oplus s\rangle}{\sqrt{2}} \quad \xrightarrow{H^{\otimes n}} \quad \text{solo stati } |y\rangle \text{ con } s \cdot y = 0.$$

La periodicità nascosta non viene misurata direttamente. Essa modifica il pattern delle fasi prodotto dalle Hadamard e determina quali ampiezze si cancellano per interferenza distruttiva.

Il computer quantistico non restituisce dunque "tutti i valori della funzione". Converte invece la simmetria

$$f(x) = f(x \oplus s)$$

in equazioni lineari sulla stringa sconosciuta.

Relazione con gli algoritmi precedenti e con Shor

Simon riprende alcuni ingredienti già incontrati:

- come in Deutsch–Jozsa e Bernstein–Vazirani, le Hadamard producono interferenza fra molte componenti della sovrapposizione;
- come in Bernstein–Vazirani, il risultato della misura fornisce informazione lineare su una stringa nascosta;
- diversamente da Grover, non si amplifica progressivamente una soluzione: si sfrutta una simmetria globale della funzione.

La novità fondamentale è la presenza di una periodicità nascosta:

$$x \longleftrightarrow x \oplus s.$$

Per questo Simon costituisce un ponte naturale verso Shor. Anche nell'algoritmo di Shor il problema computazionale viene trasformato nella ricerca di una periodicità; cambieranno però il gruppo matematico coinvolto e lo strumento usato per estrarre tale periodicità.

8.7 L'algoritmo di Shor

L'algoritmo di Shor è uno dei risultati più importanti del calcolo quantistico. Il suo obiettivo è fattorizzare un intero composto

$$N$$

in fattori non banali.

A differenza degli algoritmi precedenti, Shor non si limita a mostrare un vantaggio nel modello a *query*. Esso combina una parte classica e una subroutine quantistica per trasformare il problema della fattorizzazione in un problema di ricerca di periodicità.

La struttura generale è

$$\boxed{\text{fattorizzazione} \longrightarrow \text{ricerca dell'ordine} \longrightarrow \text{periodicità} \longrightarrow \text{QFT} \longrightarrow \text{fattori di } N.}$$

Il problema della fattorizzazione

Dato un intero composto

$$N,$$

vogliamo trovare due interi

$$p, q > 1$$

tali che

$$N = pq.$$

Esempio 8.7.1 (Semplici fattorizzazioni)**.** Alcuni casi elementari sono

$$15 = 3 \cdot 5,$$

$$21 = 3 \cdot 7,$$

$$35 = 5 \cdot 7.$$

Il problema è semplice da formulare, ma per numeri molto grandi non sono noti algoritmi classici generali che lo risolvano in tempo polinomiale nel numero di bit di N.

Shor non cerca direttamente i divisori. Introduce invece un problema intermedio: determinare l'ordine di un intero modulo N.

Dalla fattorizzazione alla ricerca dell'ordine

Scegliamo un intero

$$a$$

tale che

$$1 < a < N.$$

Calcoliamo

$$\gcd(a, N).$$

Se

$$\gcd(a, N) > 1,$$

abbiamo già trovato un fattore non banale di N e il problema è risolto.

Il caso interessante è quindi

$$\gcd(a, N) = 1.$$

Consideriamo allora la successione

$$a^0 \bmod N, \quad a^1 \bmod N, \quad a^2 \bmod N, \quad \dots$$

Poiché a è coprimo con N, questa successione è periodica.

Definiamo l'*ordine* di a modulo N come il più piccolo intero positivo

$$r$$

tale che

$$\boxed{a^r \equiv 1 \pmod{N}.}$$

Il problema di trovare r è detto *order finding* (ricerca dell'ordine).

Esempio 8.7.2 (Ordine di 2 modulo 15)**.** Prendiamo

$$N = 15, \qquad a = 2.$$

Calcoliamo:

$$2^0 \bmod 15 = 1,$$

$$2^1 \bmod 15 = 2,$$

$$2^2 \bmod 15 = 4,$$

$$2^3 \bmod 15 = 8,$$

$$2^4 \bmod 15 = 16 \bmod 15 = 1.$$

La sequenza ricomincia quindi dopo quattro passi:

$$1, 2, 4, 8, 1, 2, 4, 8, \dots$$

L'ordine è

$$\boxed{r = 4.}$$

Perché conoscere l'ordine permette di fattorizzare

Supponiamo di aver determinato l'ordine r di a modulo N.

Se r è pari, possiamo scrivere

$$a^r - 1 = \left(a^{r/2} - 1\right)\left(a^{r/2} + 1\right).$$

Poiché

$$a^r \equiv 1 \pmod{N},$$

si ha

$$N \mid a^r - 1.$$

Quindi

$$N \mid \left(a^{r/2} - 1\right)\left(a^{r/2} + 1\right).$$

Se inoltre

$$a^{r/2} \not\equiv -1 \pmod{N},$$

allora i due numeri

$$\gcd\left(a^{r/2} - 1, N\right),$$
$$\gcd\left(a^{r/2} + 1, N\right)$$

forniscono fattori non banali di N.

Esempio 8.7.3 (Dall'ordine ai fattori di 15)**.** Continuiamo con

$$N = 15, \qquad a = 2, \qquad r = 4.$$

Poiché r è pari,

$$\frac{r}{2} = 2.$$

Calcoliamo

$$a^{r/2} = 2^2 = 4.$$

Inoltre

$$4 \not\equiv -1 \pmod{15},$$

perché

$$-1 \equiv 14 \pmod{15}.$$

Possiamo quindi calcolare

$$\gcd(4 - 1, 15) = \gcd(3, 15) = 3,$$

e

$$\gcd(4 + 1, 15) = \gcd(5, 15) = 5.$$

Otteniamo così

$$\boxed{15 = 3 \cdot 5.}$$

La fattorizzazione è quindi ridotta alla determinazione efficiente di r.

Quando un tentativo non funziona

Non ogni scelta di a conduce immediatamente ai fattori.

Possono verificarsi due casi sfavorevoli:

1. l'ordine r è dispari;
2. r è pari, ma
$$a^{r/2} \equiv -1 \pmod{N}.$$

In questi casi si sceglie un altro valore di a e si ripete la procedura.

L'algoritmo di Shor è quindi probabilistico nel senso che può essere necessario ripetere la scelta di a e la subroutine di *order finding*. Tuttavia, per un intero composto non banale, la probabilità di ottenere una scelta utile è sufficientemente alta da mantenere polinomiale il costo complessivo.

La funzione periodica

Per determinare r consideriamo la funzione

$$f(x) = a^x \bmod N.$$

Per definizione dell'ordine,
$$a^{x+r} \equiv a^x a^r \equiv a^x \pmod{N},$$

quindi
$$\boxed{f(x+r) = f(x).}$$

La funzione possiede dunque periodo r.

Questa è la connessione con l'algoritmo di Simon: anche qui la quantità nascosta che vogliamo estrarre è una periodicità. La struttura matematica, tuttavia, è diversa. In Simon il periodo è una stringa rispetto allo XOR; in Shor il periodo riguarda l'aritmetica modulare sugli interi.

La parte quantistica: preparazione dei registri

La subroutine quantistica utilizza due registri.

Il primo deve poter rappresentare interi

$$x = 0, 1, \ldots, Q-1,$$

dove
$$Q = 2^t$$

è scelto come una potenza di due sufficientemente grande. Una scelta standard è

$$\boxed{N^2 \leq Q < 2N^2.}$$

Il secondo registro deve contenere i valori

$$a^x \bmod N.$$

Poniamo

$$m = \lceil \log_2 N \rceil ;$$

il secondo registro è quindi formato da m qubit ed è inizializzato nel valore binario 1, che indicheremo con $|1\rangle_m$.

Partiamo dallo stato

$$|\psi_0\rangle = |0\rangle^{\otimes t} |1\rangle_m.$$

Applicando

$$H^{\otimes t}$$

al primo registro otteniamo

$$\boxed{|\psi_1\rangle = \frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |x\rangle |1\rangle_m.}$$

Esponenziazione modulare quantistica

Applichiamo ora un circuito reversibile che calcola

$$a^x \bmod N.$$

L'operazione può essere rappresentata schematicamente come

$$U_a : |x\rangle|1\rangle_m \longmapsto |x\rangle|a^x \bmod N\rangle.$$

Dopo questa trasformazione:

$$\boxed{|\psi_2\rangle = \frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |x\rangle |a^x \bmod N\rangle.}$$

Il secondo registro contiene valori periodici con periodo r.

È importante non interpretare questo stato come una tabella che possiamo leggere integralmente. Una misura non restituirebbe tutti i valori di $a^x \bmod N$. Il vantaggio nasce dalla struttura coerente delle ampiezze del primo registro.

Misura del secondo registro

Per comprendere l'algoritmo è utile immaginare di misurare il secondo registro.

Supponiamo di ottenere un certo valore

$$a^{x_0} \bmod N.$$

Tutti gli esponenti

$$x_0, \quad x_0 + r, \quad x_0 + 2r, \quad \ldots$$

producono lo stesso valore della funzione, finché rimangono nell'intervallo

$$0, \dots, Q-1.$$

Il primo registro collassa quindi, a normalizzazione vicino, in uno stato del tipo

$$\boxed{|\psi_3\rangle = \frac{1}{\sqrt{L}} \sum_{k=0}^{L-1} |x_0 + kr\rangle,}$$

dove L è il numero di termini presenti.

Abbiamo quindi ottenuto una sovrapposizione di valori equidistanziati di passo r.

La periodicità è ora codificata direttamente nel primo registro.

Esempio 8.7.4 (Progressione periodica per $N = 15$)**.** Prendiamo ancora

$$N = 15, \qquad a = 2, \qquad r = 4.$$

Scegliamo

$$Q = 256.$$

Supponiamo che la misura del secondo registro selezioni il valore

$$2.$$

Poiché

$$2^1 \bmod 15 = 2,$$

e il periodo è 4, lo stesso valore compare per

$$x = 1, 5, 9, 13, \dots, 253.$$

Il primo registro diventa quindi

$$|\psi_3\rangle = \frac{1}{8} \sum_{k=0}^{63} |1 + 4k\rangle.$$

La distanza fra due termini consecutivi è precisamente

$$r = 4.$$

La trasformata di Fourier quantistica

Per estrarre la periodicità utilizziamo la *Quantum Fourier Transform* (trasformata di Fourier quantistica), abbreviata QFT.

Non la tratteremo qui come algoritmo autonomo; introduciamo soltanto la proprietà necessaria a Shor.

Su un registro di dimensione

$$Q$$

la QFT è definita da

$$\boxed{\mathrm{QFT}_Q|x\rangle = \frac{1}{\sqrt{Q}} \sum_{y=0}^{Q-1} e^{2\pi i x y/Q}|y\rangle.}$$

La QFT svolge, nel dominio delle ampiezze, il ruolo analogo alla trasformata discreta di Fourier: una struttura periodica nello spazio degli x produce picchi ben localizzati nello spazio degli indici y.

Se applichiamo la QFT allo stato

$$\frac{1}{\sqrt{L}} \sum_{k=0}^{L-1} |x_0 + kr\rangle,$$

l'ampiezza associata a un valore y contiene la somma

$$\sum_{k=0}^{L-1} e^{2\pi i k r y/Q}.$$

Questa è una somma geometrica. I contributi interferiscono costruttivamente quando

$$\frac{ry}{Q}$$

è vicino a un intero.

Di conseguenza, la probabilità di misura è concentrata vicino ai valori

$$\boxed{y \approx \frac{mQ}{r},}$$

con

$$m = 0, 1, \ldots, r-1.$$

La QFT converte quindi una periodicità nello spazio degli input in picchi nello spazio delle frequenze.

Il caso semplice in cui r divide Q

Se

$$r \mid Q,$$

la situazione è particolarmente trasparente.

Poniamo

$$Q = Lr.$$

Allora i picchi della distribuzione si trovano esattamente in

$$\boxed{y = m\frac{Q}{r}, \qquad m = 0, 1, \ldots, r-1.}$$

Misurando y otteniamo quindi

$$\frac{y}{Q} = \frac{m}{r}.$$

Se la frazione è ridotta ai minimi termini e m è coprimo con r, il denominatore rivela direttamente il periodo.

Esempio 8.7.5 (QFT per $N = 15$)**.** Nel nostro esempio

$$N = 15, \qquad a = 2, \qquad r = 4, \qquad Q = 256.$$

Poiché

$$4 \mid 256,$$

i picchi si trovano esattamente in

$$y = m\frac{256}{4} = 64m.$$

I possibili valori dominanti sono quindi

$$\boxed{0, \quad 64, \quad 128, \quad 192.}$$

Se la misura restituisce

$$y = 64,$$

otteniamo

$$\frac{y}{Q} = \frac{64}{256} = \frac{1}{4}.$$

Il denominatore ci fornisce

$$\boxed{r = 4.}$$

A questo punto la parte quantistica è terminata e possiamo tornare alla procedura classica per calcolare i fattori di 15.

Il caso generale: approssimazione razionale

In generale r non divide esattamente Q. La misura produce allora un valore y tale che

$$\frac{y}{Q}$$

è vicino a una frazione

$$\frac{m}{r}.$$

Il problema diventa quindi:

Data una frazione numerica y/Q, trovare una frazione semplice m/r che la approssimi sufficientemente bene.

Questo passaggio viene eseguito classicamente mediante lo sviluppo in *continued fractions* (frazioni continue).

L'espansione in frazione continua permette di ricostruire candidati razionali

$$\frac{m}{r}$$

con denominatore relativamente piccolo. Per ciascun candidato r si verifica poi classicamente se

$$a^r \equiv 1 \pmod{N}.$$

Se la verifica fallisce, si prova un altro convergente oppure si ripete la subroutine quantistica.

Schema completo dell'algoritmo

Possiamo ora riassumere Shor.

Dato un intero composto N:

1. si sceglie casualmente
$$1 < a < N;$$
2. si calcola
$$d = \gcd(a, N);$$
3. se
$$d > 1,$$
d è già un fattore non banale;
4. se
$$\gcd(a, N) = 1,$$
si usa la subroutine quantistica di *order finding* per stimare l'ordine r di a modulo N;
5. se r è dispari, si sceglie un nuovo a;
6. se
$$a^{r/2} \equiv -1 \pmod{N},$$
si sceglie un nuovo a;
7. altrimenti si calcolano
$$p = \gcd\left(a^{r/2} - 1, N\right),$$
$$q = \gcd\left(a^{r/2} + 1, N\right).$$

Quando il tentativo ha successo,

$$p$$

e

$$q$$

sono fattori non banali di N.

Esempio 8.7.6 (Esempio completo: fattorizzare 15)**.** Seguiamo l'intero algoritmo con

$$N = 15.$$

Scegliamo

$$a = 2.$$

Primo controllo:

$$\gcd(2, 15) = 1.$$

Dobbiamo quindi trovare l'ordine di 2 modulo 15.
La subroutine quantistica prepara

$$\frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |x\rangle |2^x \bmod 15\rangle,$$

con, per esempio,

$$Q = 256.$$

La funzione

$$2^x \bmod 15$$

ha periodo

$$r = 4.$$

Dopo la QFT, i risultati di misura sono concentrati sui valori

$$0, 64, 128, 192.$$

Supponiamo di misurare

$$64.$$

Allora

$$\frac{64}{256} = \frac{1}{4},$$

da cui ricaviamo il candidato

$$r = 4.$$

Verifichiamo:

$$2^4 = 16 \equiv 1 \pmod{15}.$$

L'ordine è quindi corretto.
Poiché

$$r = 4$$

è pari e

$$2^{r/2} = 2^2 = 4 \not\equiv -1 \pmod{15},$$

calcoliamo

$$\gcd(4 - 1, 15) = 3,$$

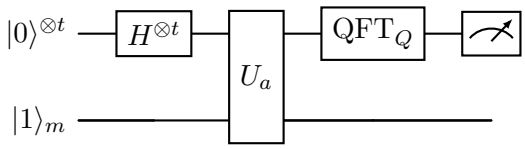
$$\gcd(4 + 1, 15) = 5.$$

Otteniamo infine

$$\boxed{15 = 3 \cdot 5.}$$

Circuito schematico della subroutine quantistica

La parte quantistica può essere rappresentata schematicamente come



dove

$$U_a : |x\rangle|1\rangle_m \longmapsto |x\rangle|a^x \bmod N\rangle.$$

In una realizzazione completa, l'esponenziazione modulare deve essere decomposta in porte reversibili elementari. Essa costituisce una parte rilevante del costo circuitale dell'algoritmo.

La misura del secondo registro, usata nella derivazione precedente per rendere trasparente la periodicità, non è concettualmente indispensabile: si può ottenere la stessa distribuzione finale sul primo registro senza leggerlo esplicitamente.

Perché la QFT è utile

La QFT non "calcola il periodo" in modo diretto.

Il suo ruolo è trasformare uno stato nel quale le ampiezze sono distribuite su posizioni periodicamente separate,

$$x_0, \quad x_0 + r, \quad x_0 + 2r, \quad \ldots,$$

in uno stato nel quale le probabilità sono concentrate vicino a frequenze

$$\frac{mQ}{r}.$$

In altre parole,

$$\boxed{\text{periodicità negli input} \longrightarrow \text{picchi nelle frequenze.}}$$

La misura fornisce quindi un'approssimazione razionale di

$$\frac{m}{r},$$

e la parte classica dell'algoritmo ricostruisce r mediante frazioni continue.

Questo è il punto nel quale la trasformata di Fourier quantistica entra realmente in Shor.

Complessità e vantaggio quantistico

Per un numero N di n bit,

$$n = \lceil \log_2 N \rceil .$$

L'algoritmo di Shor può essere implementato, nel modello ideale dei circuiti quantistici, con un numero di operazioni polinomiale in

$$n.$$

Il confronto con gli algoritmi classici deve essere formulato con cautela. Per la fattorizzazione generale non è noto un algoritmo classico polinomiale, ma non è neppure dimostrato che tale algoritmo non possa esistere.

I migliori algoritmi classici generali conosciuti hanno complessità subesponenziale, mentre Shor ha complessità polinomiale nel numero di bit.

Il vantaggio di Shor è quindi molto più forte del vantaggio quadratico di Grover.

Non si deve però concludere che un computer quantistico renda efficienti tutti i problemi classicamente difficili. Shor sfrutta una struttura aritmetica molto specifica: la periodicità dell'esponenziazione modulare.

Impatto sulla crittografia

La rilevanza applicativa di Shor deriva dal fatto che diversi sistemi crittografici a chiave pubblica basano la propria sicurezza pratica sulla difficoltà di problemi di teoria dei numeri.

La fattorizzazione di grandi interi è alla base della sicurezza di RSA.

Una versione dello stesso paradigma quantistico consente inoltre di risolvere efficientemente il problema del logaritmo discreto, con conseguenze per sistemi basati su gruppi moltiplicativi ed ellittici.

Questo non significa che ogni forma di crittografia venga annullata da Shor. In particolare, gli algoritmi crittografici simmetrici sono influenzati in modo differente: per una ricerca esaustiva, il riferimento quantistico è Grover, che produce una riduzione quadratica del costo.

Da Simon a Shor

Simon e Shor sono accomunati da un'idea profonda: una funzione apparentemente complessa nasconde una simmetria periodica e il circuito quantistico viene progettato per trasformare tale simmetria in un pattern di interferenza misurabile.

In Simon:

$$f(x) = f(x \oplus s),$$

e le Hadamard producono equazioni

$$y \cdot s = 0.$$

In Shor:

$$f(x) = a^x \bmod N,$$

con

$$f(x + r) = f(x),$$

e la QFT produce picchi vicini a

$$\frac{mQ}{r}.$$

Il passaggio concettuale può essere riassunto come

Simon	:	periodicità rispetto allo XOR,
Shor	:	periodicità dell'aritmetica modulare.

Shor rappresenta quindi il punto culminante della sequenza di algoritmi presentata in questo capitolo: sovrapposizione, interferenza, periodicità e post-elaborazione classica vengono combinate in un unico procedimento capace di modificare radicalmente il costo computazionale di un problema di grande rilevanza matematica e applicativa.

8.8 Confronto tra gli algoritmi studiati

Gli algoritmi analizzati finora mostrano diversi modi di utilizzare sovrapposizione, fasi e interferenza. Non esiste un unico "meccanismo di accelerazione quantistica": la struttura dell'algoritmo dipende dalla struttura del problema.

Per Deutsch, Deutsch–Jozsa, Bernstein–Vazirani, Grover e Simon è naturale esprimere il confronto nel modello a *query*. Shor richiede invece un confronto più generale di complessità computazionale, perché la parte decisiva è la subroutine quantistica di *order finding* implementata mediante esponenziazione modulare e QFT.

Algoritmo	**Problema**	***Query* classiche**	***Query* quantistiche**
Deutsch	Distinguere funzione costante e bilanciata su un bit	2	1
Deutsch–Jozsa	Distinguere funzione costante e bilanciata su n bit	$2^{n-1}+1$ esatte	1
Bernstein–Vazirani	Ricostruire la stringa nascosta s in $f_s(x) = s \cdot x$	n	1
Grover	Trovare un elemento marcato in una ricerca non strutturata di dimensione N	$\Theta(N)$	$\Theta(\sqrt{N})$
Simon	Ricostruire la stringa s tale che $f(x) = f(x \oplus s)$	$\Theta(2^{n/2})$ randomizzate	$O(n)$

Per Shor il confronto naturale è invece

procedura classica generale nota	subesponenziale
algoritmo di Shor	polinomiale in $\log N$

con la precisazione che non è dimostrata l'inesistenza di un algoritmo classico polinomiale per la fattorizzazione.

Dal punto di vista concettuale possiamo distinguere quattro schemi.

Per Deutsch, Deutsch–Jozsa e Bernstein–Vazirani:

$$\boxed{\text{sovrapposizione} \longrightarrow \text{codifica nelle fasi} \longrightarrow \text{interferenza} \longrightarrow \text{misura.}}$$

Per Grover:

$$\boxed{\text{marcatura di fase} \longrightarrow \text{diffusione} \longrightarrow \textit{amplitude amplification} \longrightarrow \text{misura.}}$$

Per Simon:

$$\boxed{\text{sovrapposizione} \longrightarrow \textit{oracle} \longrightarrow \text{interferenza} \longrightarrow \text{vincoli lineari} \longrightarrow \text{ricostruzione di } s.}$$

Per Shor:

$$\boxed{\text{esponenziazione modulare} \longrightarrow \text{periodicità} \longrightarrow \text{QFT} \longrightarrow \text{frazioni continue} \longrightarrow \text{fattori.}}$$

Il principio comune non è che il computer quantistico "provi contemporaneamente tutte le soluzioni". Una misura produce comunque un numero limitato di bit classici.

Il vantaggio nasce quando il circuito organizza le ampiezze in modo che la proprietà cercata determini un pattern di interferenza osservabile. Negli algoritmi successivi questa stessa idea sarà applicata a strutture periodiche più complesse.

Capitolo 9

Cenni su rumore, errori e strategie di protezione nel calcolo quantistico

I circuiti quantistici studiati nei capitoli precedenti sono stati descritti, per necessità didattica, come circuiti ideali. Si è supposto che i qubit potessero essere preparati esattamente nello stato desiderato, che ogni porta realizzasse perfettamente l'operazione prevista e che la misura restituisse senza ambiguità il risultato corretto.

Un computer quantistico reale è molto diverso. I qubit sono sistemi fisici che interagiscono inevitabilmente con l'ambiente; le porte vengono realizzate mediante segnali di controllo di durata e precisione finite; qubit vicini possono influenzarsi reciprocamente; la misura stessa può introdurre errori. Le proprietà del dispositivo, inoltre, non rimangono necessariamente costanti nel tempo e devono essere periodicamente caratterizzate e calibrate. Nei processori reali vengono infatti monitorate grandezze quali i tempi di rilassamento e coerenza T_1 e T_2, gli errori e la durata delle porte e gli errori di lettura [15, 16].

Nel calcolo classico associamo spontaneamente un errore al cambiamento di un bit, da 0 a 1 o viceversa. Nel caso quantistico la situazione è più complessa. Un errore può modificare le probabilità associate agli stati $|0\rangle$ e $|1\rangle$, ma può anche alterare le loro relazioni di fase, ridurre la coerenza di una sovrapposizione, compromettere l'entanglement o perfino portare il sistema fisico fuori dallo spazio a due livelli utilizzato per rappresentare il qubit.

Per affrontare il problema è utile distinguere quattro strategie:

1. **prevenire o sopprimere** il rumore, migliorando hardware, controllo e circuiti;
2. **mitigarne** gli effetti sul risultato;
3. **rilevare e correggere** gli errori codificando l'informazione quantistica;
4. progettare l'intera computazione in modo **tollerante ai guasti** (*fault tolerant*).

Queste strategie operano a livelli differenti e non devono essere confuse.

9.1 Dal circuito ideale al computer quantistico reale

Un algoritmo quantistico sfrutta proprietà quali sovrapposizione, interferenza ed entanglement, che dipendono dal mantenimento di precise relazioni tra le ampiezze degli stati. Una piccola variazione incontrollata della fase relativa può quindi modificare un processo di interferenza

anche senza trasformare direttamente $|0\rangle$ in $|1\rangle$. Se perturbazioni di questo tipo si accumulano durante molte operazioni, la distribuzione finale dei risultati può discostarsi sensibilmente da quella prevista dal circuito ideale.

È utile distinguere tre livelli.

- Il **livello fisico** comprende il sistema materiale che realizza i qubit, l'ambiente, i dispositivi di controllo e l'elettronica di lettura.
- Il **livello circuitale** comprende le effettive operazioni di preparazione, le porte, i tempi di attesa e le misure eseguite dalla macchina.
- Il **livello logico** è il livello astratto nel quale descriviamo l'algoritmo mediante qubit e porte ideali.

Un *errore fisico* riguarda uno o più componenti reali della macchina. Un *errore logico* è invece un errore che, dopo il processo di rilevazione e correzione previsto dal codice, induce una trasformazione non desiderata sull'informazione codificata. L'obiettivo di un computer quantistico tollerante ai guasti non è quindi costruire componenti perfetti, ma fare in modo che una certa quantità di errori fisici possa verificarsi senza trasformarsi in un errore logico.

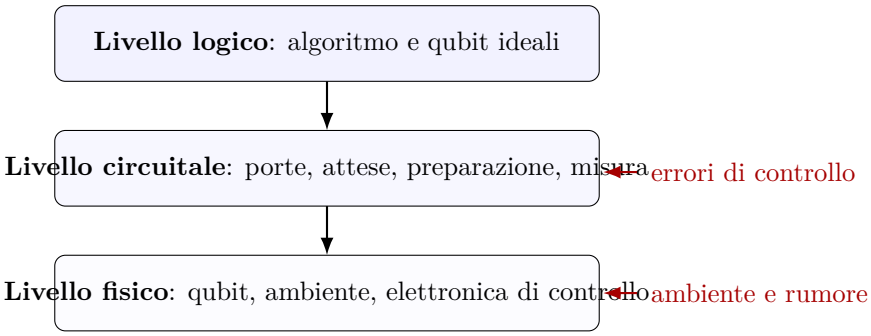


Figura 9.1: Dal livello logico al livello fisico. Le imperfezioni agiscono soprattutto ai livelli circuitale e fisico e possono propagarsi fino al risultato dell'algoritmo.

9.2 Le principali sorgenti di errore

Gli errori di un computer quantistico non hanno tutti la stessa origine. Alcuni intervengono mentre lo stato viene conservato, altri durante le porte o la misura; alcuni agiscono prevalentemente su un singolo qubit, altri possono essere correlati fra più qubit. Comprendere questa distinzione è importante perché sorgenti diverse richiedono strategie diverse [1, cap. 8].

Preparazione imperfetta dello stato

Molti circuiti richiedono che i qubit siano inizializzati nello stato $|0\rangle$. Nella macchina reale il processo di reset può non essere perfetto. Un qubit che dovrebbe iniziare in $|0\rangle$ può avere una piccola probabilità di trovarsi in $|1\rangle$ o, in alcuni sistemi fisici, in un livello energetico esterno allo spazio computazionale. L'errore è quindi presente prima ancora che il circuito abbia iniziato il calcolo.

Gli errori associati alla preparazione dello stato e alla misura vengono spesso raggruppati sotto l'acronimo *SPAM*, da *State Preparation And Measurement*. La loro caratterizzazione richiede

di preparare stati noti e verificare statisticamente con quale frequenza la misura restituisca un risultato differente [15, 16].

Rilassamento energetico e tempo T_1

Un qubit è un sistema fisico e può scambiare energia con l'ambiente. Se si trova nello stato eccitato $|1\rangle$, può perdere energia e decadere verso lo stato fondamentale:

$$|1\rangle \longrightarrow |0\rangle.$$

Il tempo caratteristico di questo processo viene indicato con T_1 e prende il nome di *tempo di rilassamento energetico*. T_1 non rappresenta un istante preciso nel quale tutti i qubit decadono, ma una scala temporale statistica. Nei dispositivi reali viene determinato preparando il qubit nello stato eccitato, attendendo per intervalli diversi e misurando la probabilità che esso sia ancora in $|1\rangle$ [15, 16]. Nel modello elementare di rilassamento verso lo stato fondamentale, a temperatura sufficientemente bassa, si scrive spesso

$$P_1(t) \simeq e^{-t/T_1}.$$

Questa è una descrizione introduttiva: a temperatura finita, o in presenza di eccitazione residua, la popolazione può tendere a un valore asintotico non nullo anziché a zero.

Un circuito che dura troppo a lungo rispetto ai tempi caratteristici dei qubit avrà quindi una probabilità crescente di essere compromesso dal rilassamento.

Perdita di coerenza, puro dephasing e tempo T_2

La perdita di energia non è l'unico problema. Consideriamo una sovrapposizione

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

Il funzionamento degli algoritmi quantistici dipende non soltanto dalle probabilità associate ai due termini, ma anche dalla loro fase relativa. Fluttuazioni dell'ambiente o del sistema di controllo possono rendere progressivamente incerta questa relazione. Questo fenomeno è detto *dephasing* o perdita di coerenza di fase; esso riprende la decoerenza discussa nel § 5.11.

È utile distinguere tre scale temporali. T_1 caratterizza il rilassamento energetico; T_ϕ caratterizza il *puro dephasing*, cioè la perdita di fase che rimarrebbe anche in assenza di rilassamento; T_2 descrive invece la perdita complessiva della coerenza trasversa. Nel semplice modello markoviano usuale vale

$$\frac{1}{T_2} = \frac{1}{2T_1} + \frac{1}{T_\phi}.$$

Questa relazione mostra che T_2 non è indipendente dal rilassamento descritto da T_1: il rilassamento energetico contribuisce già alla perdita di coerenza, mentre T_ϕ raccoglie il contributo aggiuntivo di puro dephasing. La documentazione IBM descrive infatti T_2 come una scala temporale che include sia rilassamento energetico sia pure dephasing [15]. Nel modello appena scritto segue inoltre il controllo di consistenza

$$T_2 \leq 2T_1.$$

La distinzione è essenziale: il rilassamento modifica le popolazioni degli stati, mentre il puro dephasing può compromettere l'interferenza anche quando le probabilità di trovare 0 o 1 non

sembrano cambiare in modo evidente.

Ramsey, spin echo e misura della coerenza. Una caratterizzazione elementare della coerenza prepara il qubit in una sovrapposizione, lascia trascorrere un intervallo τ, applica una seconda rotazione sensibile alla fase e misura. In una sequenza di Ramsey il decadimento del contrasto fornisce una scala spesso indicata con T_2^*, sensibile anche a detuning lento e disomogeneità quasi statiche. Inserendo un impulso di rifocalizzazione (*spin echo*) si può sopprimere parte di questi contributi lenti e ottenere una scala di coerenza più lunga. La distinzione fra T_2^*, una misura echo e il parametro T_ϕ dipende dal protocollo e dal modello di rumore; il punto operativo è che la coerenza deve essere caratterizzata mediante esperimenti specifici, così come il rilassamento viene caratterizzato mediante T_1.

Errori di controllo e delle porte

Una porta quantistica astratta è una trasformazione matematica perfettamente definita. La macchina deve però realizzarla mediante un'interazione fisica. Nei qubit controllati tramite impulsi elettromagnetici, per esempio, l'operazione dipende da parametri quali durata, frequenza, ampiezza e fase. Piccole imprecisioni possono produrre una trasformazione leggermente diversa da quella desiderata.

Se si vuole realizzare una rotazione

$$R_x(\pi)$$

ma l'impulso produce in realtà

$$R_x(\pi + \varepsilon),$$

si ha una sovrarotazione o una sottorotazione, a seconda del segno di ε. Una deviazione casuale può comportarsi come rumore. Una deviazione sistematica che si ripete sempre nello stesso modo può essere ancora più insidiosa, perché gli effetti di molte piccole rotazioni indesiderate possono accumularsi coerentemente. La calibrazione serve precisamente a ridurre questi scostamenti.

Errore coerente ed errore stocastico non sono equivalenti

Due dispositivi possono avere metriche medie simili e produrre effetti diversi su circuiti lunghi. Una sovrarotazione sistematica può accumularsi coerentemente da una porta alla successiva; un errore stocastico tende invece a produrre una dispersione statistica. Un singolo numero medio non descrive quindi, da solo, la struttura del rumore.

Errori delle porte a due qubit

Le operazioni a due qubit richiedono di attivare e controllare un'interazione fra sistemi fisici distinti. Questa operazione è in genere più complessa del controllo di un singolo qubit e costituisce spesso un elemento critico nella qualità complessiva di un circuito. Per questa ragione i dispositivi vengono caratterizzati separatamente anche attraverso metriche specifiche per le operazioni a due qubit e per ogni coppia fisicamente connessa [15].

Ciò ha una conseguenza importante per la compilazione: due circuiti logicamente equivalenti possono avere prestazioni molto differenti se uno richiede un numero maggiore di porte a due qubit o utilizza connessioni fisiche meno affidabili.

Crosstalk ed errori correlati

Nel modello più semplice si potrebbe immaginare che ogni qubit commetta errori indipendentemente dagli altri. Nelle macchine reali questa ipotesi non è sempre valida. Un segnale destinato a un qubit può avere un effetto indesiderato su un qubit vicino: questo fenomeno viene chiamato *crosstalk*. Possono inoltre esistere sorgenti ambientali comuni che producono errori contemporaneamente su più qubit.

Gli errori correlati sono particolarmente rilevanti per la correzione quantistica: un codice progettato per tollerare un singolo errore indipendente può essere meno efficace quando una stessa perturbazione colpisce simultaneamente più qubit. Anche nelle moderne implementazioni di quantum error correction gli eventi correlati rappresentano uno dei fenomeni che possono limitare le prestazioni a bassissimi tassi di errore logico [19].

Leakage: quando il qubit esce dallo spazio computazionale

La rappresentazione mediante due stati $|0\rangle$ e $|1\rangle$ è spesso un'approssimazione controllata. Il sistema fisico che realizza il qubit può infatti possedere ulteriori livelli energetici. Un impulso imperfetto può trasferire parte della popolazione verso uno di questi livelli. Si ha allora *leakage*: lo stato è uscito dallo spazio computazionale

$$\text{span}\{|0\rangle, |1\rangle\}.$$

Questo errore è qualitativamente diverso da un semplice bit flip: il sistema non si trova più nemmeno nel sottospazio utilizzato per rappresentare il qubit. Il problema è sufficientemente importante da richiedere specifiche procedure di rimozione del leakage anche in recenti implementazioni sperimentali del surface code [19].

Errori di misura

Infine, anche la misura è imperfetta. Un qubit che si trova in $|0\rangle$ può essere classificato come 1 e, viceversa, uno stato $|1\rangle$ può essere classificato come 0. Le due probabilità non sono necessariamente uguali:

$$P(1 \mid 0) \neq P(0 \mid 1).$$

I dispositivi reali possono quindi riportare separatamente queste probabilità e un valore complessivo di *readout error* [15].

È importante distinguere concettualmente due situazioni: il qubit può arrivare alla misura già nello stato sbagliato, oppure può arrivare nello stato corretto ma la catena di lettura può assegnare l'etichetta sbagliata. Nel primo caso l'errore è avvenuto durante la computazione; nel secondo nella fase finale di lettura.

9.3 Come si caratterizza un computer quantistico reale

Sapere che esistono numerose sorgenti di rumore non basta: occorre misurarne gli effetti. La caratterizzazione di un processore quantistico produce un insieme di indicatori che descrivono aspetti differenti del dispositivo. Non esiste un singolo numero capace di riassumere completamente la qualità di una macchina.

Fra le quantità più comuni troviamo:

- tempo di rilassamento T_1;
- tempo di coerenza trasversa T_2 e, quando caratterizzato separatamente, tempo di puro dephasing T_ϕ;
- errore delle porte a un qubit;
- errore delle porte a due qubit;
- durata delle porte;
- errore di lettura;
- misure ottenute mediante protocolli di benchmarking.

I backend di IBM Quantum, per esempio, espongono T_1, T_2, errori di lettura, errori e durate delle istruzioni e metriche ottenute mediante randomized benchmarking [15].

Randomized benchmarking

Un metodo molto utilizzato per caratterizzare gli errori delle porte è il *randomized benchmarking*. L'idea generale è costruire sequenze casuali di operazioni che, idealmente, dovrebbero produrre nel complesso un risultato noto. Ripetendo l'esperimento con sequenze progressivamente più lunghe si può stimare quanto rapidamente gli errori si accumulino [16].

Non è necessario, in questo contesto, entrare nella teoria matematica del protocollo. È invece importante comprenderne il ruolo: una singola esecuzione di una porta non permette di determinarne con precisione l'affidabilità; occorrono esperimenti statistici specificamente progettati. Il decadimento delle sequenze permette, sotto le ipotesi del protocollo, di stimare un errore medio delle porte con sensibilità relativamente ridotta agli errori SPAM; non ricostruisce però da solo la struttura completa del canale di rumore.

Calibrazione e variazioni nel tempo

Un'altra caratteristica fondamentale dei dispositivi reali è che alcuni parametri possono cambiare nel tempo. Le proprietà dinamiche di un QPU vengono aggiornate dopo le procedure di calibrazione; valori quali tempi caratteristici ed errori delle operazioni sono quindi associati anche al momento in cui sono stati misurati [15].

Questo fenomeno, indicato genericamente come *drift*, ha una conseguenza pratica: un circuito ottimizzato utilizzando una certa caratterizzazione del dispositivo può non essere altrettanto efficace dopo una variazione delle prestazioni dei qubit.

Fenomeno	Effetto principale	Indicatore tipico
Rilassamento	perdita della popolazione di $\|1\rangle$	T_1
Puro dephasing	perdita di fase non dovuta al rilassamento	T_ϕ
Coerenza trasversa	rilassamento + puro dephasing	T_2
Errore di porta	trasformazione imprecisa	gate error / fidelity
Errore a due qubit	interazione imperfetta	2-qubit gate error
Errore di lettura	confusione fra 0 e 1	readout error
SPAM	preparazione o misura errata	SPAM error
Leakage	uscita da span$\{\|0\rangle, \|1\rangle\}$	leakage rate
Drift	variazione temporale delle prestazioni	calibrazioni successive

Due processori con valori medi apparentemente simili possono quindi comportarsi diversamente su uno stesso algoritmo. Un errore coerente, un errore casuale e un errore fortemente correlato non hanno necessariamente le stesse conseguenze anche quando qualche misura media della loro intensità appare simile.

9.4 Quattro modi di affrontare gli errori

Le strategie utilizzate per contrastare il rumore possono essere organizzate in quattro categorie.

Soppressione. Si cerca di evitare che l'errore si produca o di ridurne fisicamente l'intensità: miglioramento dei materiali, isolamento dall'ambiente, calibrazione, ottimizzazione degli impulsi, riduzione della durata del circuito, dynamical decoupling.

Mitigazione. L'errore viene accettato, ma si cerca di stimare in modo più accurato quale sarebbe stato il risultato in assenza di rumore. La mitigazione agisce quindi principalmente sul risultato statistico e non costruisce necessariamente uno stato quantistico protetto.

Correzione. L'informazione viene codificata in modo ridondante su più sistemi fisici. Si raccolgono informazioni sull'errore e si recupera lo stato logico.

Fault tolerance. La protezione viene estesa all'intera computazione. Anche le porte, le misure, la preparazione degli ausiliari e lo stesso procedimento di correzione devono essere progettati in modo da non propagare un singolo guasto in un errore incontrollabile.

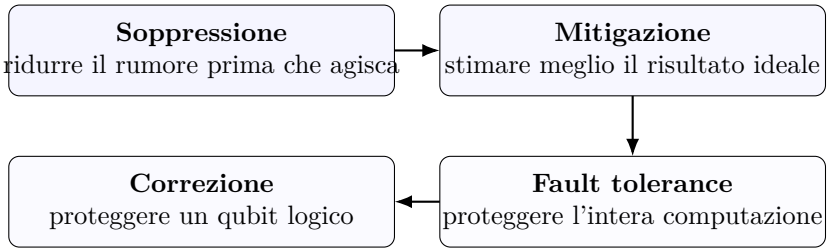


Figura 9.2: Le quattro strategie principali contro il rumore agiscono a livelli differenti e possono essere combinate nella stessa architettura.

9.5 Prevenire e sopprimere gli errori

La prima strategia consiste nel migliorare il comportamento fisico del dispositivo prima di ricorrere alla ridondanza.

Migliorare l'hardware

Una parte essenziale della ricerca sui computer quantistici riguarda qualità dei materiali, riduzione dei difetti, isolamento elettromagnetico, controllo della temperatura, progettazione delle connessioni e stabilità dei sistemi elettronici. Gli interventi dipendono fortemente dalla tecnologia utilizzata per realizzare i qubit.

Migliorare la calibrazione e il controllo

La qualità di una porta dipende non soltanto dal qubit, ma anche dai segnali che lo controllano. Una calibrazione accurata cerca di determinare quali impulsi producano nel sistema reale le trasformazioni richieste. Poiché il dispositivo può subire variazioni nel tempo, la calibrazione non è necessariamente un'operazione effettuata una volta per tutte [15].

Rendere il circuito meno vulnerabile

Anche il software può ridurre l'esposizione al rumore. Il compilatore può ridurre il numero totale delle porte, minimizzare le operazioni a due qubit, utilizzare le connessioni fisiche più favorevoli, scegliere qubit con prestazioni migliori e ridurre la profondità del circuito. Mapping e scheduling possono inoltre utilizzare la caratterizzazione corrente del backend per evitare connessioni o qubit particolarmente rumorosi.

È importante distinguere il numero di porte dalla durata fisica. In presenza di parallelismo non si otteniene il tempo del circuito sommando la durata di tutte le porte: la quantità rilevante è il *makespan* dello schedule, cioè l'intervallo fra inizio e fine della computazione determinato dal cammino temporale critico, inclusi gli idle lungo tale schedule. Gate count, depth e durata sono quindi quantità correlate ma non equivalenti. Si tratta delle stesse risorse circuitali introdotte nel § 6.10, che qui acquistano un significato direttamente fisico. Il circuito matematicamente più elegante non è quindi necessariamente quello più affidabile sulla macchina reale.

Dynamical decoupling

Durante alcuni intervalli del circuito un qubit può rimanere inattivo. Questo non significa che la sua evoluzione sia perfettamente congelata: l'ambiente continua ad agire. Il *dynamical decoupling* inserisce opportunamente sequenze di impulsi durante questi intervalli per ridurre l'effetto di interazioni indesiderate [17]. Una scelta elementare è la sequenza XX: idealmente $X^2 = \mathbb{I}$, quindi l'operazione logica complessiva resta l'identità mentre la presenza fisica degli impulsi può sopprimere parte dell'evoluzione indesiderata.

Non si tratta però di una soluzione universale. Nei circuiti con pochi intervalli inattivi può essere poco utile e gli stessi impulsi aggiuntivi possono introdurre errori. La gestione del rumore è quindi sempre un bilancio fra protezione ottenuta e nuove operazioni introdotte [17].

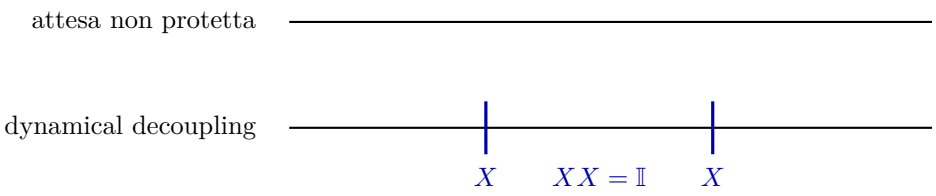


Figura 9.3: Schema concettuale di dynamical decoupling con sequenza XX. I due impulsi soddisfano idealmente $X^2 = \mathbb{I}$: l'operazione logica complessiva è quindi l'identità, mentre la presenza fisica degli impulsi può ridurre l'effetto di alcune interazioni indesiderate.

Noise tailoring: Pauli twirling e randomized compiling

Un'altra strategia consiste nel modificare il modo in cui il circuito sperimenta il rumore senza cambiarne l'azione ideale. Le tecniche di *twirling* introducono trasformazioni casuali scelte in modo che il circuito ideale resti equivalente, ma che errori coerenti strutturati vengano trasformati, nel canale medio twirlato, in un rumore più simile a un processo stocastico. Nel *Pauli twirling* il canale medio effettivo assume forma di Pauli ed è per questo utile come tecnica di *noise tailoring* [17]. Questa affermazione riguarda il canale mediato sulla randomizzazione: non implica che ogni singola realizzazione fisica del rumore sia letteralmente un errore di Pauli, né che l'infedeltà media venga automaticamente ridotta.

Il *randomized compiling* distribuisce trasformazioni casuali fra circuiti logicamente equivalenti per rendere, sotto le ipotesi del protocollo, il rumore effettivo più stocastico, meno coerente e più trattabile statisticamente. Non elimina però il drift né garantisce la stazionarietà dell'hardware.

La terminologia non è del tutto uniforme: in alcuni workflow twirling e randomized compiling vengono combinati con tecniche chiamate complessivamente *error mitigation*. In questa dispensa li classifichiamo tra soppressione e *noise tailoring* perché modificano preventivamente il circuito e il canale di rumore, anziché correggere soltanto la stima statistica finale.

9.6 Mitigare gli errori

La *quantum error mitigation* utilizza informazioni sul rumore e più esecuzioni del circuito per ottenere una stima migliore del risultato ideale. A differenza della quantum error correction, non richiede necessariamente la costruzione di un qubit logico protetto [14].

Mitigazione degli errori di lettura

Supponiamo che la misura abbia una certa probabilità di classificare $|0\rangle$ come 1 e $|1\rangle$ come 0. Possiamo caratterizzare sperimentalmente queste probabilità preparando stati noti e misurandoli molte volte. Le informazioni ottenute possono essere utilizzate per correggere statisticamente le frequenze osservate. Il procedimento non rende perfetta la misura fisica: cerca di ricostruire una distribuzione più vicina a quella che sarebbe stata osservata con un sistema di lettura ideale [18]. Se la matrice di risposta della misura viene invertita per ricostruire la distribuzione ideale, un problema mal condizionato può amplificare le fluttuazioni statistiche. Procedure pratiche impongono quindi spesso vincoli fisici o usano forme di regolarizzazione anziché una semplice inversione numerica.

Zero-noise extrapolation

Un'idea particolarmente intuitiva è la *zero-noise extrapolation* (ZNE). Si esegue il circuito più volte a differenti livelli effettivi di rumore e si osserva come varia una quantità di interesse. Il rumore può essere scalato, per esempio, mediante *gate folding* o mediante una modifica della durata/forma degli impulsi che preservi idealmente l'unitaria desiderata. Dai diversi risultati si tenta quindi di estrapolare il valore che quella quantità avrebbe nel limite di rumore nullo [14].

In forma schematica, risultati ottenuti a livelli di rumore r_1, r_2 e r_3 vengono utilizzati per stimare il valore corrispondente a

$$r \longrightarrow 0.$$

Il rumore non viene eliminato dalla macchina: viene utilizzata la sua dipendenza per correggere la stima finale. L'affidabilità dell'estrapolazione dipende però dal modello assunto e dal modo in cui il rumore viene scalato; una dipendenza non compatibile con il modello di fit può introdurre un bias residuo.

Verifica di simmetrie

Alcuni problemi possiedono proprietà che il risultato ideale deve necessariamente rispettare. Se, per esempio, un circuito dovrebbe conservare una certa quantità, un risultato che viola questa condizione può essere identificato come incompatibile con l'evoluzione ideale. Queste informazioni possono essere utilizzate per scartare alcuni risultati o pesarli diversamente. Lo scarto riduce però il campione effettivo disponibile e può quindi aumentare il costo in shot e l'incertezza statistica.

Il prezzo della mitigazione

La mitigazione non è gratuita. Richiede in genere più esecuzioni del circuito, procedure di calibrazione, elaborazione classica aggiuntiva e assunzioni sul comportamento del rumore. Inoltre, aumentando la correzione statistica può aumentare anche la varianza della stima.

Per questo motivo la distinzione deve essere mantenuta netta:

La mitigazione migliora la stima del risultato; la correzione cerca di proteggere direttamente l'informazione quantistica.

9.7 Perché la correzione quantistica sembra impossibile

La correzione classica degli errori utilizza la ridondanza. Possiamo, per esempio, rappresentare

$$0 \longrightarrow 000, \qquad 1 \longrightarrow 111.$$

Se un solo bit viene invertito, il voto di maggioranza permette di ricostruire il valore originario. Nel caso quantistico sembrano però comparire tre ostacoli fondamentali [2, cap. 10].

Primo ostacolo: non possiamo copiare un qubit sconosciuto

Come discusso nel § 5.6, il no-cloning theorem impedisce di costruire una procedura universale che trasformi

$$|\psi\rangle|0\rangle$$

in due copie indipendenti

$$|\psi\rangle|\psi\rangle$$

per ogni possibile stato $|\psi\rangle$. La ridondanza quantistica non può quindi essere ottenuta semplicemente "facendo delle copie".

Secondo ostacolo: non possiamo controllare lo stato misurandolo direttamente

Se misurassimo i qubit codificati per scoprire quale valore possiedono, potremmo distruggere proprio la sovrapposizione che vogliamo proteggere. Bisogna quindi raccogliere informazione sull'errore senza raccogliere informazione sullo stato logico.

Terzo ostacolo: gli errori sembrano continui

Un bit classico può essere corretto considerando essenzialmente la possibilità di un'inversione. Un qubit, invece, può subire una rotazione di qualsiasi angolo. La teoria della quantum error correction mostra che non è necessario misurare continuamente "quanto" il qubit si sia spostato: una misura della sindrome permette di ricondurre l'errore a classi discrete correggibili [1, cap. 10].

La discretizzazione può essere vista anche attraverso la base di Pauli. Ogni operatore lineare che agisce su un singolo qubit — per esempio ciascun operatore di Kraus E_k in una descrizione a canale — può essere espanso come

$$E_k = a_k\mathbb{I} + b_kX + c_kY + d_kZ.$$

Un processo rumoroso generale può richiedere più operatori E_k: non viene quindi descritto, in generale, da un unico operatore di errore. I codici vengono progettati in modo che la correzione delle componenti discrete rilevanti si estenda per linearità agli errori che esse generano.

La soluzione ai tre ostacoli può quindi essere sintetizzata in tre idee:

- non copiare lo stato, ma **distribuirne l'informazione**;
- non misurare il qubit logico, ma una **sindrome**;
- non stimare continuamente l'errore, ma identificarne una **classe correggibile**.

9.8 Qubit fisico, qubit logico e sindrome

Consideriamo un qubit nello stato

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

Aggiungiamo due qubit ausiliari inizializzati in $|0\rangle$ e realizziamo una trasformazione che produce

$$|\psi_L\rangle = \alpha|000\rangle + \beta|111\rangle.$$

Questo stato rappresenta un qubit logico mediante tre qubit fisici.

È fondamentale notare che

$$\alpha|000\rangle + \beta|111\rangle \neq \big(\alpha|0\rangle + \beta|1\rangle\big)^{\otimes 3}.$$

L'informazione quantistica è distribuita nelle correlazioni fra i tre sistemi. Questa distinzione permette alla ridondanza quantistica di non violare il no-cloning theorem.

Supponiamo ora che uno dei qubit subisca un bit flip. Non misuriamo direttamente i tre qubit per determinare se essi rappresentino 000 o 111. Misuriamo invece alcune relazioni di parità fra di essi. Il risultato di queste misure viene chiamato *sindrome di errore*.

La sindrome fornisce informazione sull'errore sufficiente a determinare un recupero, senza rivelare lo stato logico. In generale errori fisici differenti possono produrre la stessa sindrome: il decoder deve allora scegliere un'operazione di recupero compatibile con i dati osservati. È questa separazione fra informazione logica e informazione sull'errore che rende possibile la quantum error correction.

9.9 Il codice a tre qubit contro il bit flip

Il codice più semplice da utilizzare per comprendere il meccanismo è il codice a tre qubit contro il bit flip [1, cap. 10]. Definiamo

$$|0_L\rangle = |000\rangle, \qquad |1_L\rangle = |111\rangle.$$

Di conseguenza

$$\alpha|0\rangle + \beta|1\rangle \longrightarrow \alpha|000\rangle + \beta|111\rangle.$$

Supponiamo esplicitamente che al massimo uno dei tre qubit subisca un errore X. Misuriamo le due osservabili di parità

$$Z_1Z_2, \qquad Z_2Z_3.$$

Nel codice ideale i loro autovalori ± 1 indicano se i qubit confrontati hanno parità concorde o discorde. Sotto l'ipotesi di al massimo un singolo bit flip, le quattro possibilità sono:

Errore	$\boldsymbol{Z_1Z_2}$	$\boldsymbol{Z_2Z_3}$	**Recupero**
nessun errore	$+1$	$+1$	nessuna operazione
X_1	-1	$+1$	applicare X_1
X_2	-1	-1	applicare X_2
X_3	$+1$	-1	applicare X_3

Stabilizzatori e code space: definizione minima

Nel codice a tre qubit gli operatori

$$S_1 = Z_1 Z_2, \qquad S_2 = Z_2 Z_3$$

stabilizzano lo spazio del codice: per ogni stato logico valido $|\psi_L\rangle$ vale $S_1|\psi_L\rangle = S_2|\psi_L\rangle = |\psi_L\rangle$. Il *code space* è quindi il sottospazio comune di autovalore $+1$ per questi controlli. Un errore correggibile modifica uno o più autovalori; il pattern risultante costituisce la sindrome.

Supponiamo, per esempio, che il secondo qubit venga invertito:

$$\alpha|000\rangle + \beta|111\rangle \longrightarrow \alpha|010\rangle + \beta|101\rangle.$$

In questo modello a singolo bit flip, la sindrome identifica il secondo qubit come quello alterato. Applicando nuovamente X al secondo qubit si recupera

$$X_2^2|\psi_L\rangle = |\psi_L\rangle, \qquad X_2^2 = \mathbb{I}.$$

Il punto importante è ciò che non è stato misurato: durante il procedimento non abbiamo scoperto i coefficienti α e β.

Qubit ausiliari

Nella pratica le parità possono essere trasferite a qubit ausiliari, detti anche *ancilla*, che vengono successivamente misurati. Il principio è

qubit di dati $\longrightarrow$ interazione con ancilla $\longrightarrow$ misura degli ancilla $\longrightarrow$ sindrome $\longrightarrow$ recupero.

Questo schema, in forme molto più sofisticate, resta alla base delle moderne tecniche di correzione quantistica [2, cap. 10].

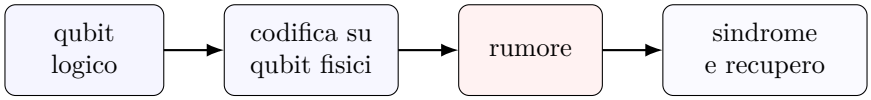


Figura 9.4: Schema concettuale della correzione: l'informazione logica viene distribuita su più qubit fisici; dopo il rumore si estrae una sindrome e si determina il recupero senza misurare direttamente lo stato logico.

Quando il codice fallisce

Il codice appena descritto corregge un singolo bit flip. Se vengono invertiti due qubit, la sindrome può essere interpretata come se fosse avvenuto un errore sul terzo; il recupero produce allora un errore logico. La ridondanza non rende quindi il sistema invulnerabile: stabilisce con precisione quale insieme di errori può essere corretto.

9.10 E gli errori di fase?

Il codice precedente protegge contro l'inversione fra $|0\rangle$ e $|1\rangle$, ma non contro un errore di fase. L'idea può però essere trasferita nella base

$$\{|+\rangle, |-\rangle\}.$$

In questa base un phase flip scambia il ruolo dei due stati:

$$Z|+\rangle = |-\rangle, \qquad Z|-\rangle = |+\rangle.$$

Un errore di fase può quindi essere trasformato concettualmente in un problema analogo al bit flip mediante un cambiamento di base con porte di Hadamard [6].

Il codice di Shor: perché è importante

Nel 1995 Peter Shor mostrò che era possibile combinare la protezione contro bit flip e phase flip in un unico codice a nove qubit [9]. Il risultato è importante soprattutto per ciò che dimostra: nonostante il no-cloning theorem, il problema della misura e il carattere continuo delle perturbazioni, un errore arbitrario che colpisca un singolo qubit può essere corretto. Il codice di Shor ha quindi un valore fondativo, anche se non rappresenta l'unica né necessariamente la più economica soluzione per un'architettura concreta. Il *codice di Shor* non va confuso con l'*algoritmo di Shor* per la fattorizzazione studiato nel § 8.7: i due risultati condividono l'autore, ma risolvono problemi completamente diversi.

9.11 Dalla correzione alla tolleranza ai guasti

Finora abbiamo implicitamente assunto che il circuito utilizzato per estrarre la sindrome e correggere l'errore funzioni correttamente. Ma anche quei circuiti sono realizzati mediante qubit e porte fisiche rumorose. Nasce così un problema apparentemente paradossale: come possiamo utilizzare componenti imperfetti per correggere l'imperfezione degli stessi componenti?

La semplice disponibilità di un codice di correzione non basta. La procedura deve essere progettata in modo tale che un singolo guasto non produca una cascata di errori superiore alla capacità di correzione del codice. Questo è il principio della *fault tolerance*. Le operazioni vengono eseguite direttamente sugli stati codificati, con circuiti progettati per limitare la propagazione degli errori [1, cap. 10].

Propagazione degli errori

Un errore presente su un qubit può propagarsi attraverso una porta che coinvolge più qubit. Questo rende particolarmente delicata l'estrazione della sindrome: un ancilla difettoso non deve essere messo in condizione di contaminare molti qubit di dati. Per questo le architetture fault tolerant utilizzano circuiti progettati per limitare la propagazione, ancilla preparati e verificati, misure ripetute ed elaborazione classica delle sindromi.

Per la CNOT la propagazione dei Pauli si vede in modo trasparente mediante coniugazione.

Indicando con c il controllo e con t il target,

$$\begin{aligned} \mathrm{CNOT}\, X_c\, \mathrm{CNOT} &= X_c X_t, & \mathrm{CNOT}\, Z_c\, \mathrm{CNOT} &= Z_c, \\ \mathrm{CNOT}\, X_t\, \mathrm{CNOT} &= X_t, & \mathrm{CNOT}\, Z_t\, \mathrm{CNOT} &= Z_c Z_t. \end{aligned}$$

Le due trasformazioni non banali mostrano perché un singolo fault può propagarsi fra qubit e perché l'architettura di estrazione della sindrome deve essere progettata con attenzione.

Il concetto di soglia

Consideriamo un modello molto semplificato nel quale ogni operazione fisica fallisce con probabilità p e un *gadget* fault tolerant è progettato per tollerare un singolo fault. Perché il gadget produca un errore logico devono allora verificarsi almeno due guasti opportunamente combinati. In prima approssimazione possiamo aspettarci un comportamento del tipo

$$p_L \simeq C p^2,$$

dove p_L è il tasso di errore logico e C dipende dalla struttura del codice e del circuito. Questa espressione non è una legge universale: serve soltanto a illustrare l'idea essenziale.

Se

$$C p^2 < p,$$

cioè, in questo modello giocattolo, se

$$p < \frac{1}{C},$$

il gadget codificato ha migliorato l'affidabilità rispetto alla corrispondente operazione fisica. Questa semplice disuguaglianza serve soltanto a motivare il concetto di *soglia di errore* e non deve essere confusa con il teorema generale della soglia. Se il tasso di errore fisico è sufficientemente basso e valgono determinate ipotesi sul rumore, l'aumento controllato della ridondanza permette di ridurre progressivamente il tasso di errore logico [1, cap. 10].

Essere "sotto soglia" non significa quindi avere eliminato gli errori. Significa che, aumentando opportunamente la protezione, l'errore logico diminuisce invece di aumentare.

9.12 Il surface code

Fra le numerose famiglie di codici quantistici, il *surface code* ha ricevuto particolare attenzione perché combina protezione dagli errori con una struttura basata prevalentemente su interazioni locali in una griglia bidimensionale [13, 6].

L'idea generale può essere compresa senza entrare nel formalismo completo degli stabilizzatori. Un surface code utilizza due tipi principali di qubit:

- **qubit di dati**, sui quali è distribuita l'informazione logica;
- **qubit di misura**, utilizzati ripetutamente per raccogliere informazioni sulle sindromi.

I qubit di misura estraggono localmente i valori di controlli, o *stabilizzatori*, di tipo X e di tipo Z. I primi sono sensibili soprattutto agli errori di fase, i secondi agli errori di bit. Le misure vengono ripetute ciclicamente perché non soltanto i qubit di dati possono commettere errori: anche una misura della sindrome può essere sbagliata. Per questo il decoder utilizza in particolare

i *cambiamenti* dei valori di sindrome fra cicli successivi e analizza un pattern di eventi nello spazio e nel tempo, non semplicemente una singola misura.

Il procedimento può essere schematizzato come

$$\text{qubit di dati} \longrightarrow \text{misure locali ripetute} \longrightarrow \text{storia delle sindromi}$$
$$\longrightarrow \text{decoder classico} \longrightarrow \text{interpretazione dell'errore.}$$

La parte classica è quindi essenziale: una macchina fault tolerant non è soltanto un processore quantistico, ma un sistema nel quale elaborazione quantistica e decodifica classica devono procedere in stretto coordinamento.

Proteggere una memoria logica, tuttavia, non basta a costruire un computer universale. Occorrono anche preparazioni, misure e operazioni logiche fault tolerant. Nel surface code queste operazioni possono essere realizzate mediante deformazioni del codice e protocolli come la *lattice surgery*; per ottenere universalità bisogna inoltre gestire operazioni non-Clifford, spesso introducendo risorse speciali come gli *magic states* [13]. Il dettaglio architetturale va oltre lo scopo di questa dispensa, ma il punto concettuale è essenziale: una memoria protetta è soltanto uno dei componenti di una computazione fault tolerant.

Distanza del codice

Più precisamente, la distanza d di un codice è il peso minimo di un operatore logico non banale: il numero minimo di qubit fisici sui quali deve agire un'operazione capace di modificare l'informazione logica senza essere equivalente a uno stabilizzatore. Nel modello ideale standard, un codice di distanza d può correggere fino a

$$\left\lfloor \frac{d-1}{2} \right\rfloor$$

errori. Nel surface code questa nozione può essere visualizzata come la lunghezza minima di una catena di errori capace di realizzare un operatore logico. Aumentare la distanza aumenta quindi la protezione, ma richiede anche più qubit fisici, più misure e più elaborazione classica.

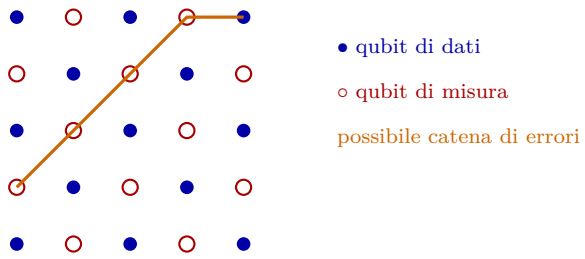


Figura 9.5: Schema puramente concettuale di una griglia di surface code. I qubit di dati e di misura cooperano nell'estrazione ripetuta delle sindromi; il decoder interpreta i pattern prodotti dagli errori.

Dalla teoria all'esperimento

La condizione "sotto soglia" è stata osservata sperimentalmente in sistemi recenti. Google Quantum AI ha realizzato memorie surface-code di distanza 5 e 7 integrate con un decoder in tempo reale. Nel codice di distanza 7, realizzato con 101 qubit, l'aumento della distanza di due unità ha ridotto il tasso di errore logico di un fattore 2.14 ± 0.02 [19].

Il significato del risultato è il principio, non il singolo numero: aumentando la dimensione del codice, il qubit logico può diventare più affidabile, purché la qualità dell'hardware sia già sufficientemente elevata. Lo stesso lavoro mostra inoltre che, a livelli di errore molto bassi, possono emergere eventi rari e correlati che diventano a loro volta un limite importante [19].

9.13 Più shot non significano meno rumore

La distinzione fra rumore fisico e incertezza statistica è particolarmente importante e riprende la discussione del § 6.9. Un circuito quantistico viene normalmente ripetuto molte volte. Ogni ripetizione è detta *shot*. Se un evento ha probabilità p, tale probabilità viene stimata contando con quale frequenza esso compare nei diversi shot.

Per shot indipendenti e per una distribuzione stazionaria, se $\hat{p}$ è la frequenza osservata, l'incertezza statistica binomiale è

$$\sigma_{\hat{p}} = \sqrt{\frac{p(1-p)}{N}},$$

e quindi scala come $1/\sqrt{N}$. L'ipotesi di distribuzione stazionaria è importante: il drift discusso nel § 9.3 può rendere meno appropriato un modello nel quale tutti gli shot sono estratti dalla stessa distribuzione.

Più shot permettono dunque di stimare con maggiore precisione la distribuzione prodotta dal dispositivo. Non significa però che quella distribuzione diventi più vicina alla distribuzione ideale.

Supponiamo che, idealmente, il risultato debba essere sempre 0, ma che il sistema di lettura assegni erroneamente il valore 1 nell'8% dei casi. Con 100 shot ci attendiamo circa otto risultati pari a 1; con 10 000 shot circa ottocento. Aumentando il numero di shot, la frequenza osservata si avvicina con maggiore precisione a 0.08, non a zero.

Abbiamo quindi migliorato la precisione con cui conosciamo il risultato rumoroso, non abbiamo corretto il rumore. Per avvicinarci al risultato ideale dobbiamo caratterizzare e mitigare l'errore di lettura oppure utilizzare una forma appropriata di protezione e correzione.

più shot $\Rightarrow$ meno incertezza statistica, più shot $\not\Rightarrow$ meno errore sistematico.

9.14 Dal dispositivo rumoroso al risultato affidabile

Nessuna delle tecniche descritte in questo capitolo costituisce, isolatamente, una soluzione generale. In una macchina reale occorre combinare più livelli di intervento. La tassonomia concettuale del § 9.4 resta basata su quattro categorie — soppressione, mitigazione, correzione e fault tolerance — mentre il flusso operativo seguente è deliberatamente più dettagliato e include anche le fasi preparatorie di caratterizzazione, compilazione e validazione.

1. **Caratterizzare**: misurare T_1, T_2, errori delle porte, readout error, crosstalk, leakage e variazioni temporali.
2. **Compilare**: adattare il circuito all'hardware disponibile, scegliendo qubit, connettività e decomposizione delle porte in modo da ridurre profondità e operazioni rumorose.

3. **Sopprimere**: ridurre fisicamente gli errori attraverso hardware, calibrazione, controllo degli impulsi e tecniche come il dynamical decoupling.
4. **Mitigare**: utilizzare, quando appropriato, più esecuzioni rumorose per ottenere una migliore stima della quantità ideale.
5. **Rilevare e correggere**: codificare l'informazione in qubit logici, estrarre sindromi e applicare procedure di recupero.
6. **Rendere il calcolo fault tolerant**: estendere la protezione alle porte logiche, agli ancilla, alle misure della sindrome e al controllo classico.
7. **Validare**: confrontare i risultati con casi risolvibili classicamente, circuiti di riferimento, simmetrie e stime dell'incertezza.

Strategia	**Su che cosa agisce**	**Obiettivo**	**Costo**
Soppressione / noise tailoring	hardware, controllo e circuito	ridurre o rendere più gestibile il rumore prima della lettura	tecnologia e controllo
Mitigazione	risultato statistico	ridurre il bias della stima	più shot e calcolo classico
Correzione quantistica	informazione codificata, sindromi e recupero	mantenere affidabile il qubit logico	ridondanza e ancilla
Fault tolerance	intera computazione	impedire che pochi fault si propaghino in errori logici	elevato overhead

9.15 Errori concettuali frequenti

1. **"Un errore quantistico significa sempre che** 0 **è diventato** 1**."** No. Possono cambiare fase, coerenza, correlazioni o perfino il sottospazio fisico occupato dal sistema.
2. **"**T_1 **e** T_2 **sono due meccanismi indipendenti."** No. Sono tempi caratteristici: T_1 descrive il rilassamento energetico, T_ϕ il puro dephasing e T_2 la perdita complessiva di coerenza trasversa, alla quale contribuiscono entrambi. Nel semplice modello markoviano vale inoltre $T_2 \leq 2T_1$.
3. **"Se aumento gli shot, elimino il rumore."** No. Aumento la precisione statistica con cui stimo il risultato prodotto dal dispositivo.
4. **"Mitigazione e correzione quantistica sono sinonimi."** No. La mitigazione cerca di ricostruire una migliore stima del risultato ideale; la correzione protegge direttamente informazione codificata.
5. **"Codificare un qubit in tre qubit significa produrne tre copie."** No. Lo stato logico è distribuito nelle correlazioni fra i qubit fisici.
6. **"Per correggere un errore bisogna misurare il qubit logico."** No. Si misurano sindromi che contengono informazione sull'errore senza rivelare lo stato logico.
7. **"Un codice di correzione rende perfetto il computer."** No. Ogni codice corregge soltanto determinati insiemi di errori e introduce un costo di risorse.
8. **"Essere sotto soglia significa non avere più errori."** No. Significa che, aumentando opportunamente la protezione, il tasso di errore logico diminuisce.

9. **“La quantum error correction rende inutile migliorare l’hardware.”** No. La correzione scalabile funziona soltanto quando i componenti fisici sono già sufficientemente affidabili.

9.16 Sintesi

Il passaggio dal circuito quantistico ideale a un computer reale introduce numerose sorgenti di errore. Il rilassamento trasferisce energia dal qubit all’ambiente ed è caratterizzato da T_1; il puro dephasing è caratterizzato da T_ϕ; la coerenza trasversa complessiva è descritta da T_2 e risente di entrambi i processi. Imperfezioni di controllo producono errori delle porte; crosstalk ed effetti ambientali possono generare errori correlati; il leakage porta lo stato fuori dal sottospazio computazionale; infine la misura può confondere gli esiti.

Nessun singolo parametro descrive completamente la qualità di una macchina. Per questo vengono utilizzati tempi di coerenza, errori delle porte, errori di lettura, benchmarking e procedure periodiche di calibrazione [15, 16].

Il rumore può essere affrontato a livelli differenti. La soppressione cerca di evitare che gli errori si producano. La mitigazione utilizza informazioni sul rumore e più esecuzioni del circuito per ottenere stime migliori. La quantum error correction codifica invece l’informazione in uno spazio fisico più grande, produce sindromi dalle quali il decoder determina un recupero compatibile.

Il codice a tre qubit mostra l’idea fondamentale: uno stato

$$\alpha|0\rangle + \beta|1\rangle$$

può essere codificato come

$$\alpha|000\rangle + \beta|111\rangle$$

senza creare tre copie indipendenti del qubit. Le misure di parità producono una sindrome che, nel modello a singolo bit flip, consente di individuare il qubit colpito senza rivelare i coefficienti dello stato logico. Nei codici generali la sindrome identifica invece una classe di errori compatibili, dalla quale il decoder determina un recupero.

La fault tolerance estende questo principio a tutto il calcolo. Anche la codifica, le porte, gli ancilla e le misure della sindrome sono imperfetti e devono essere organizzati in modo che un singolo guasto non produca una cascata di errori. La nozione di soglia stabilisce il criterio fondamentale: se l’hardware possiede un livello di errore sufficientemente basso, aumentando la protezione è possibile diminuire il tasso di errore logico.

I surface code rappresentano uno degli esempi più importanti di questa strategia. Esperimenti recenti hanno mostrato direttamente il regime nel quale l’aumento della distanza del codice riduce l’errore logico [19]. La conclusione fondamentale è quindi che rumore e correzione non costituiscono un problema separato dagli algoritmi quantistici: stabilire se un algoritmo possa essere eseguito utilmente richiede di considerare insieme circuiti, hardware, tempi di coerenza, errori delle porte, strategie di mitigazione, qubit logici e costo della fault tolerance.

Appendice A

Reversibilità e calcolo quantistico

Nel **calcolo quantistico la reversibilità non è un requisito accessorio**: discende direttamente dalla struttura matematica della meccanica quantistica.

Se uno stato quantistico iniziale è rappresentato da un vettore

$$|\psi\rangle,$$

la sua evoluzione, finché non viene effettuata una misura, è descritta da un operatore unitario U:

$$|\psi'\rangle = U|\psi\rangle.$$

Un operatore unitario soddisfa

$$U^\dagger U = UU^\dagger = I,$$

e quindi possiede sempre un inverso:

$$U^{-1} = U^\dagger.$$

Perciò, conoscendo lo stato finale,

$$|\psi'\rangle,$$

è sempre possibile ricostruire lo stato iniziale:

$$|\psi\rangle = U^\dagger|\psi'\rangle.$$

Ogni trasformazione quantistica unitaria è dunque necessariamente reversibile.

Consideriamo una normale porta logica classica AND:

$$(a, b) \longrightarrow a \wedge b.$$

Essa associa, per esempio,

$$(0,0) \to 0, \qquad (0,1) \to 0, \qquad (1,0) \to 0.$$

Se conosciamo soltanto il risultato 0, non possiamo sapere quale fosse l'ingresso.

La trasformazione ha quindi **perso informazione**.

Una trasformazione di questo tipo non può essere rappresentata direttamente da un operatore unitario, perché un operatore unitario deve essere biunivoco: stati iniziali distinti devono produrre stati finali distinti.

In termini matematici,

$$U|\psi_1\rangle = U|\psi_2\rangle$$

implicherebbe, applicando U^{-1},

$$|\psi_1\rangle = |\psi_2\rangle.$$

Quindi un'evoluzione unitaria **non può fondere due stati distinti nello stesso stato finale**.

Questo punto è particolarmente importante perché uno stato quantistico non contiene soltanto valori 0 e 1, ma anche **ampiezze e fasi relative**.

Per esempio,

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$$

e

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

hanno le stesse probabilità di produrre 0 o 1 se misurati nella base computazionale, ma sono stati quantistici differenti perché differiscono nella fase relativa.

Una trasformazione quantistica deve preservare questa informazione in modo coerente. In particolare, gli operatori unitari preservano i prodotti scalari:

$$\langle\psi|\phi\rangle = \langle U\psi|U\phi\rangle.$$

Di conseguenza preservano anche:

- le norme degli stati;
- i prodotti interni e le sovrapposizioni tra stati;
- le probabilità di transizione tra stati trasformati dalla stessa unitaria;
- la distinguibilità tra stati.

Una trasformazione unitaria può invece modificare sia le probabilità di misura in una base fissata sia le fasi relative: è proprio questo controllo coerente a rendere possibile l'interferenza quantistica.

Questa conservazione dell'informazione è ciò che rende possibile l'**interferenza quantistica**, sulla quale si basano molti algoritmi quantistici.

Le porte quantistiche devono quindi essere reversibili

Una porta classica come NOT è già reversibile:

$$0 \to 1, \qquad 1 \to 0.$$

Infatti applicandola una seconda volta si recupera il valore iniziale:

$$\mathrm{NOT}(\mathrm{NOT}(x)) = x.$$

La sua controparte quantistica, la porta Pauli-X, soddisfa infatti

$$X^2 = I.$$

Anche la porta Hadamard è reversibile:

$$H^2 = I.$$

E una delle porte fondamentali nei sistemi a più qubit, la **CNOT**, realizza

$$|a, b\rangle \longrightarrow |a, a \oplus b\rangle.$$

Questa operazione è reversibile perché

$$|a, a \oplus b\rangle \longrightarrow |a, b\rangle$$

applicando nuovamente la stessa CNOT.

Supponiamo di voler implementare una funzione classica

$$f(x).$$

La trasformazione

$$|x\rangle \longrightarrow |f(x)\rangle$$

non è necessariamente reversibile, perché valori diversi di x potrebbero avere lo stesso $f(x)$.

Nel calcolo quantistico si usa invece una trasformazione del tipo

$$\boxed{|x\rangle|y\rangle \longrightarrow |x\rangle|y \oplus f(x)\rangle}$$

dove il valore originale x viene conservato.

Se inizialmente $y = 0$,

$$|x\rangle|0\rangle \longrightarrow |x\rangle|f(x)\rangle.$$

La trasformazione complessiva rimane reversibile perché dall'uscita possiamo recuperare l'ingresso.

Questo è il modo standard con cui una funzione classica viene incorporata in un circuito quantistico.

C'è anche una ragione computazionale più profonda.

Un algoritmo quantistico crea normalmente una sovrapposizione di molti stati:

$$|\psi\rangle = \sum_x \alpha_x |x\rangle.$$

Una porta quantistica agisce linearmente:

$$U|\psi\rangle = \sum_x \alpha_x U|x\rangle.$$

Le varie componenti della sovrapposizione devono quindi continuare a mantenere le proprie ampiezze e fasi, perché successivamente possano **interferire**.

Un algoritmo quantistico cerca infatti di organizzare l'interferenza in modo che:

- le ampiezze associate alle risposte sbagliate si cancellino;
- le ampiezze associate alle risposte desiderate si rafforzino.

Se durante il calcolo si distruggesse irreversibilmente informazione sulle diverse componenti della sovrapposizione, si perderebbe anche la possibilità di controllarne coerentemente l'interferenza.

Il ruolo dell'*uncomputation*

La reversibilità porta anche a una tecnica tipicamente quantistica chiamata *uncomputation*.

Durante un calcolo possono essere creati qubit ausiliari, detti *ancilla*:

$$|x\rangle|0\rangle \to |x\rangle|g(x)\rangle.$$

Dopo aver utilizzato $g(x)$, non è generalmente possibile semplicemente cancellare il secondo registro ponendolo a zero:

$$|g(x)\rangle \to |0\rangle.$$

Questa sarebbe infatti un'operazione irreversibile.

Si applica invece il circuito inverso:

$$U_g^{-1},$$

ottenendo

$$|x\rangle|g(x)\rangle \longrightarrow |x\rangle|0\rangle.$$

Questo procedimento elimina l'informazione temporanea senza distruggerla arbitrariamente ed evita che i qubit ausiliari rimangano entangled con il risultato del calcolo.

La misura costituisce l'importante eccezione.

Se abbiamo

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

una misura può produrre

$$|0\rangle$$

oppure

$$|1\rangle.$$

Dal risultato della misura non è possibile ricostruire in generale i coefficienti originali α e β.

La misura è quindi **irreversibile** nel modello standard della meccanica quantistica.

Per questo un circuito quantistico viene normalmente pensato come:

$$\boxed{\text{preparazione dello stato} \longrightarrow \text{evoluzione unitaria reversibile} \longrightarrow \text{misura}}$$

Durante la parte computazionale l'evoluzione è reversibile; l'irreversibilità compare nel momento in cui si estrae informazione classica mediante la misura.

La reversibilità è fondamentale nel calcolo quantistico perché **l'evoluzione isolata di un sistema quantistico è descritta da operatori unitari**, e ogni operatore unitario è invertibile. Questo impedisce che durante il calcolo stati distinti vengano confusi o che l'informazione contenuta nelle ampiezze e nelle fasi venga distrutta. La conservazione coerente di tale informazione rende possibile l'interferenza quantistica e, quindi, il funzionamento stesso degli algoritmi quantistici.

In sintesi:

> **Il calcolo quantistico è reversibile perché l'evoluzione di un sistema quantistico chiuso è unitaria. Una trasformazione unitaria è invertibile e non può distruggere informazione: stati iniziali distinti devono rimanere distinguibili durante l'evoluzione. Questa proprietà preserva ampiezze e fasi e rende possibile l'interferenza quantistica. L'irreversibilità compare invece nella misura, attraverso la quale il risultato quantistico viene convertito in informazione classica.**

Bibliografia

[1] M. A. Nielsen e I. L. Chuang, *Quantum Computation and Quantum Information*, 10th Anniversary Edition, Cambridge University Press, 2010.

[2] P. Kaye, R. Laflamme e M. Mosca, *An Introduction to Quantum Computing*, Oxford University Press, 2007.

[3] R. de Wolf, *Quantum Computing: Lecture Notes*, arXiv:1907.09415.

[4] A. Di Pierro, *Quantum Computing: Appunti delle lezioni*.

[5] J. Hellenkamp, S. Stump e D. Unruh, *Introduction to Quantum Computing: Lecture Notes*, RWTH Aachen, 2025.

[6] A. F. Kockum et al., *Lecture Notes on Quantum Computing*, arXiv:2311.08445.

[7] L. Castellani, *Lecture Notes on Quantum Computation*, Università del Piemonte Orientale, 2021.

[8] J. Preskill, *Lecture Notes for Physics 219: Quantum Computation*, California Institute of Technology, 2004.

[9] P. W. Shor, *Scheme for Reducing Decoherence in Quantum Computer Memory*, Physical Review A **52**, R2493–R2496 (1995), DOI: https://doi.org/10.1103/PhysRevA.52.R2493.

[10] A. M. Steane, *Error Correcting Quantum Code*, Physical Review Letters **77**, 793–797 (1996), DOI: https://doi.org/10.1103/PhysRevLett.77.793.

[11] E. Knill e R. Laflamme, *Theory of Quantum Error-Correcting Codes*, Physical Review A **55**, 900–911 (1997), DOI: https://doi.org/10.1103/PhysRevA.55.900.

[12] D. Gottesman, *Stabilizer Codes and Quantum Error Correction*, PhD thesis, California Institute of Technology, 1997, https://arxiv.org/abs/quant-ph/9705052.

[13] A. G. Fowler, M. Mariantoni, J. M. Martinis e A. N. Cleland, *Surface Codes: Towards Practical Large-Scale Quantum Computation*, Physical Review A **86**, 032324 (2012), DOI: https://doi.org/10.1103/PhysRevA.86.032324.

[14] K. Temme, S. Bravyi e J. M. Gambetta, *Error Mitigation for Short-Depth Quantum Circuits*, Physical Review Letters **119**, 180509 (2017), DOI: https://doi.org/10.1103/PhysRevLett.119.180509.

[15] IBM Quantum, *View backend details*, IBM Quantum Documentation, https://quantum.cloud.ibm.com/docs/en/guides/qpu-information, consultato il 18 agosto 2026.

[16] IBM Quantum, *Real-time benchmarking for qubit selection*, IBM Quantum Documentation, https://quantum.cloud.ibm.com/docs/en/tutorials/real-time-benchmarking-for-qubit-selection, consultato il 18 agosto 2026.

[17] IBM Quantum, *Error mitigation and suppression techniques*, IBM Quantum Documentation, https://quantum.cloud.ibm.com/docs/en/guides/error-mitigation-and-suppression-techniques, consultato il 18 agosto 2026.

[18] S. Bravyi, S. Sheldon, A. Kandala, D. C. McKay e J. M. Gambetta, *Mitigating Measurement Errors in Multiqubit Experiments*, Physical Review A **103**, 042605 (2021), DOI: https://doi.org/10.1103/PhysRevA.103.042605.

[19] Google Quantum AI and Collaborators, *Quantum error correction below the surface code threshold*, Nature **638**, 920–926 (2025), DOI: https://doi.org/10.1038/s41586-024-08449-y.

[20] D. Deutsch, *Quantum Theory, the Church–Turing Principle and the Universal Quantum Computer*, Proceedings of the Royal Society of London A **400**, 97–117 (1985), DOI: https://doi.org/10.1098/rspa.1985.0070.

[21] D. Deutsch e R. Jozsa, *Rapid Solution of Problems by Quantum Computation*, Proceedings of the Royal Society of London A **439**, 553–558 (1992), DOI: https://doi.org/10.1098/rspa.1992.0167.

[22] E. Bernstein e U. Vazirani, *Quantum Complexity Theory*, Proceedings of the 25th Annual ACM Symposium on Theory of Computing (STOC), 11–20 (1993), DOI: https://doi.org/10.1145/167088.167097.

[23] L. K. Grover, *A Fast Quantum Mechanical Algorithm for Database Search*, Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC), 212–219 (1996), DOI: https://doi.org/10.1145/237814.237866.

[24] D. R. Simon, *On the Power of Quantum Computation*, Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 116–123 (1994), DOI: https://doi.org/10.1109/SFCS.1994.365701.

[25] P. W. Shor, *Algorithms for Quantum Computation: Discrete Logarithms and Factoring*, Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 124–134 (1994), DOI: https://doi.org/10.1109/SFCS.1994.365700.

[26] G. Strang, *Linear Algebra*, MIT OpenCourseWare, 18.06SC, https://ocw.mit.edu/courses/18-06sc-linear-algebra-fall-2011/.

[27] G. Strang, *Linear Algebra*, MIT OpenCourseWare, 18.06, https://ocw.mit.edu/courses/18-06-linear-algebra-spring-2010/.

[28] S. Boyd, *ENGR108: Introduction to Matrix Methods*, Stanford University, https://web.stanford.edu/class/engr108/.

[29] Stanford University, *STATS305C – Normal Theory: Kronecker product*, https://web.stanford.edu/class/stats305c/lectures/Normal_Theory.html.

[30] W. K. Wootters e W. H. Zurek, *A Single Quantum Cannot Be Cloned*, Nature **299**, 802–803 (1982).

[31] IBM Quantum Learning, *Quantum information: single systems*, https://quantum.cloud.ibm.com/learning/en/courses/basics-of-quantum-information/single-systems/quantum-information.

[32] IBM Quantum Learning, *Density matrix basics*, https://quantum.cloud.ibm.com/learning/en/courses/general-formulation-of-quantum-information/density-matrices/density-matrix-basics.

[33] IBM Quantum Learning, *Multiple systems and reduced states*, https://quantum.cloud.ibm.com/learning/en/courses/general-formulation-of-quantum-information/density-matrices/multiple-systems.

[34] IBM Quantum Learning, *Quantum circuits: no-cloning theorem*, https://learning.quantum.ibm.com/course/basics-of-quantum-information/quantum-circuits.