Università degli Studi di Milano - Bicocca

Teoria della informazione e della computazione quantistica

Raccolta di appunti, dispense e libri

Anno accademico 2021/2022

Marco Gobbo e Gabriele Morandi

`https://github.com/marcogobbo/tecnologie-quantistiche`

6 novembre 2021

Indice

Capitolo 1

Meccanica quantistica

Lezione 1 - 04/10/2021

1.1 Stati e qubit

Il **bit** è il concetto fondamentale su cui si basa la teoria dell'informazione e computazione classica. Similmente, la teoria dell'informazione e computazione quantistica si basa sul concetto analogo di **quantum bit** o **qubit**. Che cos'è un qubit? Dal punto di vista della meccanica quantistica un qubit è un qualsiasi sistema a due stati (livelli). Ad esempio, si può creare utilizzando le due differenti polarizzazioni del fotone, utilizzando l'allineamento dello spin di un nucleo immerso in un campo magnetico uniforme oppure anche usando i due stati di un elettrone che orbita attorno ad un singolo atomo o molecola (ammonia-based quantum computer[i]). Così come il bit classico possiede uno **stato**, il quale è identificato da uno 0 o 1, anche il qubit ha uno stato i cui livelli sono solitamente indicati con $|0\rangle$ e $|1\rangle$ (utilizzeremo durante tutto il corso la *notazione di Dirac*[ii]). L'idea è quella di considerare i qubit come portatori di informazioni esattamente come lo sono i bit nei computer classici. La differenza fondamentale tra bit e qubit è che gli stati quantistici possono esistere in configurazioni differenti dai soli $|0\rangle$ e $|1\rangle$ poiché è possibile formare *combinazioni lineari* o *sovrapposizioni* di stati:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \quad \text{dove} \quad \alpha, \beta \in \mathbb{C}\,.$$

Vedremo più avanti che formalmente uno stato $|\psi\rangle$ non è altro che un vettore di un opportuno spazio vettoriale: tale vettore può essere decomposto sugli elementi della base ortonormale $\{|0\rangle\,, |1\rangle\}$, chiamata anche **base computazionale**.
Per capire il significato di questo scrittura si ricordi che la QM (Quantum Mechanics) è probabilistica e ci permette di estrarre solamente un'informazione ben precisa: quando si ha solamente lo stato $|0\rangle$ (o $|1\rangle$) si ha la certezza che il sistema si trovi in $|0\rangle$ (o $|1\rangle$) (la probabilità è 1), tuttavia quando si ha la sovrapposizione precedente la frazione di volte

[i] Si veda ad esempio *Ferguson, A., Cain, P., Williams, D., & Briggs, G. (2002). Ammonia-based quantum computer. Phys. Rev. A, 65, 034303.* Solitamente si impiegano degli atomi i cui spettri presentano due livelli energetici molto vicini tra loro e al tempo stesso molto lontani da tutti gli altri livelli.

[ii] Anche conosciuta come la notazione bra-ket, si tratta di un formalismo introdotto da Paul Dirac per indicare uno stato quantistico e tutte le operazioni ad esso collegate. Il nome deriva dal fatto che il prodotto scalare di uno stato $|\psi\rangle$ (ket) con uno stato duale $\langle\phi|$ (bra) viene indicato con una parentesi $\langle\phi|\psi\rangle$ (bra-ket).

che una misura dà come risultato $|0\rangle$ o $|1\rangle$ dipende direttamente dai coefficienti α e β. In altre parole, un sistema in una sovrapposizione di stati ha una ben precisa probabilità (non certezza) che la misura produca $|0\rangle$ o $|1\rangle$. Dalle leggi della QM la suddetta probabilità è data da $P(|0\rangle) = |\alpha|^2$ e $P(|1\rangle) = |\beta|^2$ rispettivamente. La corretta normalizzazione di $|\psi\rangle$ impone che

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

Si noti che un'eventuale fase globale in $|\psi\rangle$ è irrilevante perché scompare sempre in qualsiasi calcolo fisico (moduli quadri, valori di aspettazione, ecc.). Sono solamente due i numeri indipendenti che possono essere impiegati per la determinazione univoca di un qubit: per tale ragione sembrerebbe che a differenza di un bit classico, un qubit possa memorizzare un'infinità di informazioni. Il problema è che tale conclusione non è del tutto vera per lo "strano" comportamento della QM: l'unico modo di estrarre informazioni dagli stati è effettuare una **misura**, ma è impossibile estrarre da una singola misurazione sia α sia β a causa del collasso dello stato. Ad esempio, supponiamo che il sistema si trovi in $\alpha\,|0\rangle + \beta\,|1\rangle$ e supponiamo che una misura sperimentale dia come risultato 0 (singolo bit di informazione): in seguito alla misura lo stato collassa in $|0\rangle$ e d'ora in avanti qualsiasi misura effettuata su questo sistema produrrà sempre 0 con probabilità 1.
Per determinare univocamente α e β si necessiterebbero di un'infinità di esperimenti su un'infinità di stati tutti preparati nel medesimo qubit, ma, come vedremo, ciò non è auspicabile a causa delle "bizzarre" leggi della QM. Nonostante ciò, questo non significa che non sia possibile impiegare i qubit per estrarre e contenere informazioni nei computer, perché questo "strano" comportamento fa sì che solamente particolari operazioni siano predittive: uno degli scopi del corso è proprio quello di cercare di studiare e comprendere come si possono ricavare informazioni, quali informazioni possono essere estratte e in che modo lo si può fare.
Per comprendere al meglio i concetti che introdurremo cominciamo con un riassunto dei principi generali della QM:

- **I Postulato** (**Stato**): Che cos'è uno stato? Utilizziamo la notazione di Dirac per rappresentare un vettore $|\psi\rangle$ di uno spazio di Hilbert $\mathcal{H}$ (molto spesso uno spazio vettoriale finito dimensionale) e diremo che $|\psi\rangle \in \mathcal{H}$. Uno stato è un **raggio** tale che $\||\psi\rangle\| = 1$ (per la conservazione della probabilità) e $|\psi\rangle \cong e^{i\alpha}\,|\psi\rangle$ con[iii] $\alpha \in \mathbb{R}$. Dato che la fase globale è irrilevante, quando due stati differiscono per una fase hanno il medesimo effetto fisco.

Consideriamo due stati $|\psi_1\rangle\,, |\psi_2\rangle \in \mathcal{H}$: la loro combinazione lineare $|\psi\rangle \equiv \alpha_1\,|\psi_1\rangle + \alpha_2\,|\psi_2\rangle \in \mathcal{H}$. Per mantenere la conservazione della probabilità, si può sempre normalizzare lo stato: $|\psi\rangle \to \frac{|\psi\rangle}{\||\psi\rangle\|}$.

Definizione 1.1 (**Prodotto scalare**). *Sia $|\psi\rangle$ uno stato ("vettore", "ket") e $\langle\phi|$ uno stato duale ("vettore duale", "bra"). Definiamo **prodotto scalare** la seguente azione del bra sul ket:*

$$\langle\phi|:\ |\psi\rangle \longrightarrow \langle\phi|\psi\rangle \in \mathbb{C}\,.$$

[iii] La notazione $\cong$ significa "equivalente a".

Nel caso dei qubit consideriamo $\mathcal{H} = \mathbb{C}^2$, quindi un generico vettore viene indicato con $\begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$ dove $z_1, z_2 \in \mathbb{C}$. Come detto in precedenza, possiamo considerare la base ortonormale

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}; \tag{1.1.1}$$

in questo modo il generico vettore di $\mathbb{C}^2$ può essere scritto come combinazione lineare dei vettori di base:

$$\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = z_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + z_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = z_1 |0\rangle + z_2 |1\rangle .$$

Usando questa notazione vettoriale possiamo anche riscrivere il prodotto scalare di due stati generici:

$$|\phi\rangle = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}, \quad |\psi\rangle = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}, \quad \Rightarrow \quad \langle\psi|\phi\rangle = w_1^* z_1 + w_2^* z_2 ;$$

chiaramente i vettori di base sono ortonormali: $\langle 0|0\rangle = \langle 1|1\rangle$ e $\langle 0|1\rangle = \langle 1|0\rangle = 1$. Dato un ket (vettore) come costruiamo il bra (vettore duale)? Prendendo l'aggiunto, ossia il trasposto complesso coniugato:

$$|\phi\rangle = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}, \quad \Rightarrow \quad \langle\phi| \equiv |\phi\rangle^\dagger = \begin{pmatrix} z_1^* & z_2^* \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}^\dagger = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}^{t,*} ;$$

in questo modo il prodotto scalare è direttamente il prodotto matriciale riga $\times$ colonna:

$$\langle\psi|\phi\rangle = \begin{pmatrix} w_1^* & w_2^* \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = w_1^* z_1 + w_2^* z_2 .$$

Riassumendo, come possiamo scrivere un generico qubit? Possiamo pensarlo come un vettore di $\mathbb{C}^2$ decomposto sulla base computazionale

$$|\psi\rangle = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = z_1 |0\rangle + z_2 |1\rangle ,$$

che soddisfa i seguenti due vincoli

- **Conservazione della probabilità**: $|z_1|^2 + |z_2|^2 = 1$.
- **Invarianza di fase**: $\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \cong e^{i\alpha} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \Rightarrow \begin{matrix} z_1 \cong e^{i\alpha} z_1 \\ z_2 \cong e^{i\alpha} z_2 \end{matrix}$.

Per implementare il primo vincolo possiamo facilmente scrivere

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right) e^{i\phi_1} |0\rangle + \sin\left(\frac{\theta}{2}\right) e^{i\phi_2} |1\rangle ;$$

mentre per implementare l'invarianza di fase dobbiamo ricordarci che ϕ_1 e ϕ_2 sono fasi arbitrarie perciò abbiamo della libertà e possiamo moltiplicare lo stato precedente per $e^{i\alpha}$:

$$e^{i\alpha} |\psi\rangle \cong |\psi\rangle = \left[\cos\left(\frac{\theta}{2}\right) e^{i\phi_1} |0\rangle + \sin\left(\frac{\theta}{2}\right) e^{i\phi_2} |1\rangle\right] e^{i\alpha} ;$$

la scelta standard è quella di porre $\alpha = -\phi_1$ in maniera tale che

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + \sin\left(\frac{\theta}{2}\right)e^{i\phi}|1\rangle \ . \tag{1.1.2}$$

La relazione (1.1.2) rappresenta la parametrizzazione standard di un generico qubit mediante due numeri reali θ e ϕ. Si noti che come sottolineato in precedenza la fase globale scompare in qualsiasi conto fisico, tuttavia $e^{i\phi}$ è fondamentale perché origina i fenomeni di interferenza.

1.1.1 Sfera di Bloch

È possibile visualizzare lo stato generico di un qubit mediante un espediente grafico: si introduce il seguente versore unitario in 3 dimensioni $\vec{n} = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta)$ e si disegna la sfera $S^2 \in \mathbb{R}^3$; in questo modo il generico qubit (1.1.2) può essere disegnato identificando il punto sulla sfera individuato dagli angoli θ e ϕ. Tale sfera prende il nome di **Sfera di Bloch**: esiste una corrispondenza uno a uno fra tutti i possibili qubit scrivibili mediante (1.1.2) e i punti sulla sfera di Bloch.

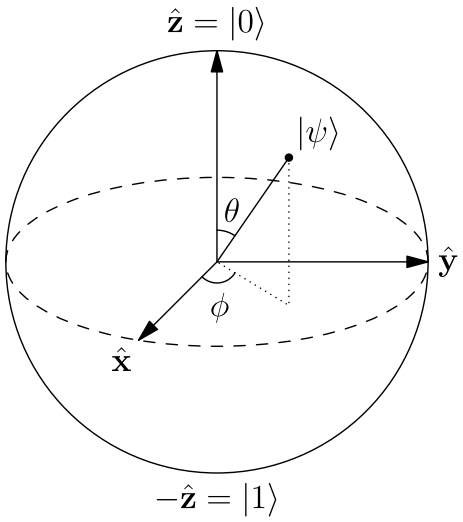


Figura 1.1: Rappresentazione generale di un qubit $|\psi\rangle$ sulla sfera di Bloch. Si noti dalla (1.1.2) come per $\theta = 0$ si abbia $|\psi\rangle = |0\rangle$ (polo Nord) e invece per $\theta = \pi$ risulta $|\psi\rangle = |1\rangle$ (polo Sud).

È fondamentale evidenziare che la rappresentazione dei qubit tramite sfera di Bloch è solo un espediente grafico poiché il prodotto scalare tra qubit è diverso dal classico prodotto scalare di $\mathbb{R}^3$. In $\mathbb{C}^2$ gli stati $|0\rangle$ e $|1\rangle$ sono ovviamente ortogonali, mentre, come evidente dalla figura, su S^2 si ha $\langle 0|1\rangle = -1$. Questo fatto è dovuto ad una sorta di doppio conteggio in (1.1.2) per la presenza di $\theta/2$.

Lezione 2 - 08/10/2021

1.2 Osservabili

- **II Postulato** (**Osservabili**): Che cosa si può misurare in QM? Vengono misurate le **osservabili**, ossia **operatori autoaggiunti** (o **hermitiani**) $\hat{A}$ tali che

$$\hat{A} : \mathcal{H} \to \mathcal{H} \text{ con } \hat{A}^\dagger = \hat{A} \,,$$

  dove più precisamente $\hat{A}^\dagger \equiv (\hat{A}^t)^*$. Dal punto di vista degli elementi di matrice, calcolare l'aggiunto di A_{ij} significa $A^\dagger_{ij} = A^*_{ji}$. Dunque le matrici autoaggiunte (hermitiane) sono tali che $A^\dagger \equiv (A^t)^* = A$.

In base a ciò che abbiamo visto sulla notazione braket ($\langle\phi| = |\phi\rangle^\dagger$) abbiamo necessariamente che

$$|\psi\rangle = B\,|\phi\rangle \;, \quad \Rightarrow \quad \langle\psi| = \langle\phi|\,B^\dagger \,.$$

Focalizzando la nostra attenzione sugli operatori autoaggiunti, richiamiamo un importante teorema dell'algebra lineare:

Teorema 1.1 (**Teorema Spettrale**)**.** *Sia $\hat{A}$ un operatore autoaggiunto su uno spazio di Hilbert $\mathcal{H}$ (reale o complesso). Allora esiste una base ortonormale di $\mathcal{H}$ composta da autovettori di $\hat{A}$, ossia $\exists\ \{|n\rangle\} \in \mathcal{H}$ tale che $\hat{A}\,|n\rangle = a_n\,|n\rangle$ dove gli autovalori $a_n \in \mathbb{R}$.*

Si noti dal teorema che $\langle n|m\rangle = \delta_{nm}$ dove $n, m = 1, \ldots, N$ con $N \equiv \dim \mathcal{H}$. Trattandosi di una base, qualsiasi vettore dello spazio di Hilbert può essere scritto come combinazione lineare di tali vettori:

$$|\psi\rangle = \sum_{n=1}^{N} \alpha_n\,|n\rangle \;, \text{ dove } \alpha_n \equiv \langle n|\psi\rangle \in \mathbb{C}\,.$$

Ritornando al nostro caso del sistema a due livelli, lo spazio di Hilbert in esame è $\mathbb{C}^2$, dove consideriamo la **base canonica** (o **base computazionale**) data dagli stati $|0\rangle$ e $|1\rangle$ (si veda (1.1.1)). In questo spazio vettoriale gli operatori sono rappresentati da matrici 2×2. La più generale matrice 2×2 hermitiana contenente 4 parametri reali è

$$A = \begin{pmatrix} a+b & c-id \\ c+id & a-b \end{pmatrix} \,,$$

dove $a, b, c, d \in \mathbb{R}$. Si noti che sulla diagonale le entrate sono puramente reali. Così come abbiamo decomposto uno stato generico $|\psi\rangle$ mediante combinazione lineare di autovettori $|n\rangle$, possiamo decomporre il generico operatore hermitiano di $\mathbb{C}^2$ come

$$A = a\mathbb{I} + c\sigma_1 + d\sigma_2 + b\sigma_3 \,,$$

dove $\mathbb{I}$ è la matrice **identità** e $\sigma_1, \sigma_2, \sigma_3$ sono le **matrici Pauli**:

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \,, \qquad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \,, \qquad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \,. \tag{1.2.1}$$

Matrice di Pauli	Autovettori		Autovalori
σ_1	$\|+\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$,	$\|-\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$	$\{1,-1\}$
σ_2	$\|i\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\i\end{pmatrix}$,	$\|-i\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-i\end{pmatrix}$	$\{1,-1\}$
σ_3	$\|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$,	$\|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$	$\{1,-1\}$

Tabella 1.1: Autovettori e autovalori delle matrici di Pauli.

Si ricordi che le matrici di Pauli sono i generatori del momento angolare in QM e sono infatti utilizzate per descrivere lo spin come $\hat{\vec{S}} = \frac{\hbar}{2}\hat{\vec{\sigma}}$. I relativi autovalori e autovettori sono mostrati nella tabella 1.1.
Dato che in futuro ci tornerà utile, osserviamo che gli autovettori di σ_1 e σ_2 possono essere espressi mediante base computazionale come

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}, \quad |i\rangle = \frac{|0\rangle + i\,|1\rangle}{\sqrt{2}}, \quad |-i\rangle = \frac{|0\rangle - i\,|1\rangle}{\sqrt{2}}, \tag{1.2.2}$$

Utilizzando la rappresentazione dei qubit tramite sfera di Bloch, questi autovettori sono mostrati in figura 1.2.

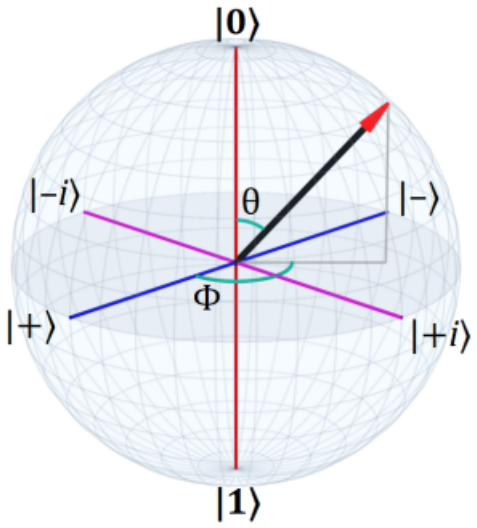


Figura 1.2: Rappresentazione degli autovettori delle matrici di Pauli sulla sfera di Bloch. Il punto indicato dalla freccia rossa indica un generico qubit.

Come detto in precedenza, le 3 matrici di Pauli parametrizzano lo spin e i 3 assi della sfera di Bloch possono essere associati allo spin. Considerando lo stato generico $|\psi\rangle$ della (1.1.2), possiamo definire lo spin lungo una direzione generica $\vec{\sigma}\cdot\vec{n}$ dove $\vec{n} = (\cos\phi\sin\theta, \sin\phi\sin\theta, \cos\theta)$:

$$\vec{\sigma}\cdot\vec{n} = \cos\phi\sin\theta\sigma_1 + \sin\phi\sin\theta\sigma_2 + \cos\theta_3 \ ;$$

così facendo è un semplice esercizio di QM dimostrare che $|\psi\rangle$ è autostato di $\vec{\sigma}\cdot\vec{n}$, ossia $\vec{\sigma}\cdot\vec{n}\,|\psi\rangle = |\psi\rangle$. Questo significa che dato uno stato sulla sfera di Bloch, allora è autostato

dello spin nella direzione individuata da tale qubit: l'idea fisica alla base della sfera di Bloch è che la direzione arbitraria scelta non è altro che la direzione della quantizzazione dello spin.

1.3 Misurazioni

- **III Postulato** (**Regola di Born**):

  1. **Misurazione**: sia $\hat{A}$ un osservabile con autostati $|n\rangle$, ossia $\hat{A}|n\rangle = a_n|n\rangle$. Prendiamo per semplicità $a_n \neq a_m \ \forall n \neq m$ (osservabile con autovalori distinti). Consideriamo uno stato generico espanso sugli autostati precedenti: $|\psi\rangle = \sum_n \alpha_n |n\rangle$. Allora una misura dell'osservabile $\hat{A}$ produce il valore a_n con probabilità data da $|\alpha_n|^2$ (assumendo lo stato correttamente normalizzato).
  2. **Collasso dello stato**: cosa succede allo stato del sistema dopo la misurazione? Istantaneamente lo stato $|\psi\rangle$ collassa sull'autostato associato all'autovalore risultante dalla misura. Ad esempio se misurando otteniamo a_n allora $|\psi\rangle \to |n\rangle$. Effettuando delle misure successive sullo stato si ottiene sempre $|n\rangle$ con probabilità esattamente uguale a 1.

Esempio 1.1. *Consideriamo per esempio il generico qubit* (1.1.2) *e immaginiamo di voler effettuare delle misurazioni in differenti basi. Supponiamo di voler misurare lo spin lungo z (base* $\{|0\rangle, |1\rangle\}$ *di* σ_3*) e lungo x (base* $\{|+\rangle, |-\rangle\}$ *di* σ_1*). Essendo il qubit già decomposto sulla base computazionale, una misurazione lungo z produrrà*

$$P(|0\rangle) = \left|\cos\left(\frac{\theta}{2}\right)\right|^2, \qquad P(|1\rangle) = \left|\sin\left(\frac{\theta}{2}\right)\right|^2.$$

Per capire il risultato della misurazione lungo x, invece, dobbiamo espandere $|\psi\rangle$ *sulla base* $\{|+\rangle, |-\rangle\}$*: usando le* (1.2.2) *per esprimere* $\{|0\rangle, |1\rangle\}$ *in termini di* $\{|+\rangle, |-\rangle\}$ *ricaviamo*

$$P(|+\rangle) = \frac{1}{2}\left|\cos\left(\frac{\theta}{2}\right) + e^{i\phi}\sin\left(\frac{\theta}{2}\right)\right|^2, \qquad P(|-\rangle) = \frac{1}{2}\left|\cos\left(\frac{\theta}{2}\right) - e^{i\phi}\sin\left(\frac{\theta}{2}\right)\right|^2.$$

Si noti come in entrambe le situazioni la probabilità risulta correttamente normalizzata: $P(|0\rangle) + P(|1\rangle) = P(|+\rangle) + P(|-\rangle) = 1$.

Esempio 1.2. *Consideriamo lo stato* $|+\rangle$ *delle* (1.2.2)*. Qual è l'interpretazione fisica di tale stato? Supponiamo che rappresenti lo spin di una particella: quando lo spin si trova in* $|+\rangle$ *allora sappiamo con certezza che punta lungo la direzione x, ossia* $P(|+\rangle) = 1$*. Al contrario, per una misurazione lungo z sappiamo che* $P(|0\rangle) = 1/2$ *e* $P(|1\rangle) = 1/2$*: abbiamo la certezza del risultato lungo le x, ma lungo le z si ha totale incertezza. Questo fenomeno è dovuto alla non commutatività degli operatori di spin nelle 3 direzioni:*

$$\left[\hat{S}_i, \hat{S}_j\right] = i\hbar\varepsilon_{ijk}\hat{S}_k\,.$$

Se consideriamo infatti il sistema preparato in $|+\rangle$ *e supponiamo di effettuare una misura lungo z ottenendo* $|0\rangle$ *allora lo stato collasserà in* $|0\rangle$ *e, d'ora in avanti, qualsiasi misurazione lungo z produrrà sempre* $|0\rangle$ *con* $P(|0\rangle) = 1$*. Nonostante ciò, il fatto che* $\hat{S}_z$ *non commuti con* $\hat{S}_x$ *fa sì che una misura successiva lungo x "rigeneri" dell'incertezza:* $P(|+\rangle) = 1/2$ *e* $P(|-\rangle) = 1/2$ *(si veda* $|0\rangle$ *espresso in termini di* $\{|+\rangle, |-\rangle\}$ *dalle* (1.2.2)*).*

Discutiamo la generalizzazione del III postulato nel caso in cui alcuni autovalori associati ad autostati differenti siano uguali, ovvero in presenza di **degenerazione**. Per esempio supponiamo il caso $N = \dim \mathcal{H} = 6$:

$$|\psi\rangle = \alpha_1 |1\rangle + \alpha_2 |2\rangle + \alpha_3 |3\rangle + \alpha_4 |4\rangle + \alpha_5 |5\rangle \alpha_6 |6\rangle \ ,$$

dove supponiamo la degenerazione su $a_1 = a_2$ e $a_4 = a_5 = a_6$. Introduciamo gli operatori $\hat{P}_{a_i}$ che considerano solamente la parte di $|\psi\rangle$ corrispondente all'autospazio associato ad a_i:

$$|\psi\rangle = \underbrace{\alpha_1 |1\rangle + \alpha_2 |2\rangle}_{\hat{P}_{a_1}|\psi\rangle} + \underbrace{\alpha_3 |3\rangle}_{\hat{P}_{a_3}|\psi\rangle} + \underbrace{\alpha_4 |4\rangle + \alpha_5 |5\rangle + \alpha_6 |6\rangle}_{\hat{P}_{a_4|\psi\rangle}} ;$$

tali operatori prendono il nome di **proiettori** e soddisfano le proprietà seguenti:

1. $\hat{P}^\dagger_{a_i} = \hat{P}_{a_i}$.
2. $\hat{P}^2_{a_i} = \hat{P}_{a_i}$.
3. $\sum_i \hat{P}_{a_i} = \mathbb{I}$.

I proiettori sono utili per scrivere la **regola di Born** (III postulato) nel caso generale: dato uno stato $|\psi\rangle$ con degenerazione sugli autovalori a_i, la probabilità di ottenere il risultato a_n è

$$P(a_n) = \left\| \hat{P}_{a_n} |\psi\rangle \right\|^2 ;$$

dopo la misura la funzione d'onda collassa sul seguente stato normalizzato:

$$|\psi\rangle \to \frac{\hat{P}_{a_n} |\psi\rangle}{\left\| \hat{P}_{a_n} |\psi\rangle \right\|} .$$

Ad esempio, nel caso dello stato sopra scritto, la probabilità di ottenere $a_1 = a_2$ non è altro che

$$P(a_1) = \left\| \hat{P}_{a_1} |\psi\rangle \right\|^2 = \| \alpha_1 |1\rangle + \alpha_2 |2\rangle \|^2 = |\alpha_1|^2 + |\alpha_2|^2 ,$$

e lo stato collassa su

$$|\psi\rangle \to \frac{\alpha_1 |1\rangle + \alpha_2 |2\rangle}{\sqrt{|\alpha_1|^2 + |\alpha_2|^2}} ;$$

ora si ha incertezza su quale stato si trova $|\psi\rangle$, ma con un esperimento successivo siamo in grado di risolvere la degenerazione e ottenere $|1\rangle$ o $|2\rangle$.

1.4 Evoluzione temporale

Il postulato successivo riguarda l'evoluzione temporale degli stati:

- **IV Postulato** (**Evoluzione temporale**): L'evoluzione temporale di uno stato generico $|\psi(0)\rangle$ è descritta dall'equazione di Schrödinger:

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

  dove $\hat{H}$ è l'operatore (hermitiano) **hamiltoniana** del sistema. L'equazione di Schrödinger conserva le probabilità: $\langle\psi(t)|\psi(t)\rangle = \langle\psi(0)|\psi(0)\rangle = 1$.

Solitamente si risolve questa equazione introducendo l'**operatore di evoluzione temporale** $\hat{U}(t)$:

$$|\psi(t)\rangle = \hat{U}(t)|\psi(0)\rangle \ ;$$

quando l'hamiltoniana è indipendente dal tempo $\hat{U}(t)$ diventa semplicemente

$$\hat{U}(t) = e^{-\frac{i}{\hbar}\hat{H}t} \ ;$$

se invece $\hat{H} = \hat{H}(t)$, è necessario distinguere i casi di hamiltoniane commutanti o non commutanti a tempi differenti.
Come detto sopra, l'evoluzione temporale preserva le probabilità e ciò è una diretta conseguenza del fatto che $\hat{U}(t)$ sia **unitario**:

- $\hat{U}\hat{U}^\dagger = \hat{U}^\dagger U = \mathbb{I} \quad \Rightarrow \quad \hat{U}^\dagger = \hat{U}^{-1}$.
- Il prodotto scalare è conservato: $\left\langle \hat{U}\phi \middle| \hat{U}\psi \right\rangle = \left\langle \phi \middle| \hat{U}^\dagger\hat{U}\psi \right\rangle = \langle\phi|\psi\rangle$.

Notiamo che $\hat{U}(t)$ per hamiltoniane indipendenti da t è effettivamente unitario:

$$\hat{U}^\dagger\hat{U} = \left(e^{-\frac{i}{\hbar}\hat{H}t}\right)^\dagger e^{-\frac{i}{\hbar}\hat{H}t} = e^{\frac{i}{\hbar}\hat{H}t}e^{-\frac{i}{\hbar}\hat{H}t} = \mathbb{I} \, .$$

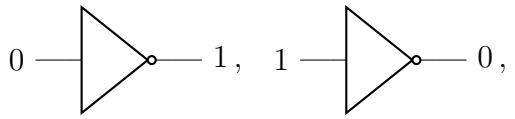
1.5 Gate

Definizione 1.2 (**Porte quantistiche**). *L'analogo quantistico delle porte (o gate) logiche classiche sono le* ***porte quantistiche*** *(o* ***gate quantistici****). Un gate quantistico è un operatore unitario che cambia lo stato del sistema.*

Notiamo che una delle principali differenze che rendono di difficile implementazione le porte quantistiche risiede nel fatto che non possiamo implementare direttamente le più semplici operazioni classiche come `AND`, `OR` o `XOR`.

Definizione 1.3 (**Circuito Quantistico**). *Un* ***circuito quantistico*** *è un modello di computazione quantistica in cui una sequenza ordinata di gate quantistici è applicata ai qubit.*

In un circuito classico l'uso dei gate logici è banale. Supponiamo di considerare un bit che si trova in 0 o 1: un gate costituisce l'implementazione di un agente esterno che cambia lo stato del bit. Si pensi ad esempio al gate `NOT` per il quale $a \to$ `NOT` a:



Nel caso invece di un qubit i circuiti funzionano diversamente perché le porte agiscono su sistemi a due livelli. Immaginiamo che a causa di un agente esterno il qubit $|\psi\rangle$ subisca un evoluzione temporale $\hat{U}$: rappresentiamo questo fatto mediante il circuito seguente

$$|\psi\rangle \;\text{——}\boxed{\hat{U}}\text{——}\; \hat{U}\,|\psi\rangle \ ,$$

Si ricordi che $\hat{U}$ è sempre un operatore unitario: ad esempio per un'hamiltoniana indipendente dal tempo si ha semplicemente $\hat{U}(t) = e^{-\frac{i}{\hbar}\hat{H}t}$.
Consideriamo le matrici di Pauli: sappiamo che sono hermitiane ($\sigma_i^\dagger = \sigma_i$) e che soddisfano la proprietà $\sigma_i^2 = \mathbb{I}$, ma questo significa che sono anche matrici unitarie. Questo fatto ci permette di costruire[iv] dei gate in cui $\hat{U} = \hat{\sigma}_i$. Ad esempio è possibile implementare dei gate come $\mathbb{I}$, $\sigma_1 \equiv X$, $\sigma_2 \equiv Y$ e $\sigma_2 \equiv Z$. Ricordando che $\sigma_i\sigma_j = 2i\varepsilon_{ijk}\sigma_k$, notiamo che $XZ = -iY$ e inoltre anche la matrice $-iY$ è unitaria. Per tale ragione molto spesso, al posto di considerare i gate $\{\mathbb{I}, X, Y, Z\}$ si sceglie la base $\{\mathbb{I}, X, Z, XZ\}$: questo significa che possiamo implementare i gate seguenti

$$\text{——}\boxed{X}\text{——}\ , \qquad \text{——}\boxed{Z}\text{——}\ , \qquad \text{——}\boxed{XZ}\text{——}\ ,$$

Consideriamo l'`X-gate`: dalle (1.2.1) è evidente che X rappresenta una sorta di "quantum" `NOT` perché inverte semplicemente lo stato della base computazionale:

$$\begin{aligned} |0\rangle \;\text{——}\boxed{X}\text{——}\; |1\rangle \ , \\ |1\rangle \;\text{——}\boxed{X}\text{——}\; |0\rangle \ , \end{aligned}$$

Consideriamo ora lo `Z-gate`: gli stati della base computazionale sono autovettori con autovalori 0 e 1 di σ_3, quindi questo gate inverte semplicemente il segno

$$\begin{aligned} |0\rangle \;\text{——}\boxed{Z}\text{——}\; |0\rangle \ , \\ |1\rangle \;\text{——}\boxed{Z}\text{——}\; -|1\rangle \ , \end{aligned}$$

L'azione dello `Z-gate` su un generico qubit risulterà quindi in

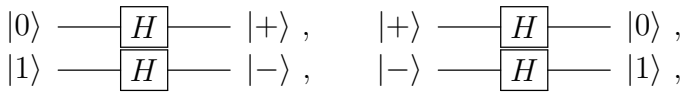
$$a\,|0\rangle + b\,|1\rangle \;\text{——}\boxed{Z}\text{——}\; a\,|0\rangle - b\,|1\rangle \ ,$$

e questo significa che Z aggiunge semplicemente una fase $e^{i\pi} = -1$ allo stato. Ricapitolando: l'`X-gate` implementa un'interferenza dall'esterno che inverte lo stato (ad esempio cambia segno dello spin lungo x) e lo `Z-gate` implementa l'introduzione di una fase.
Una matrice particolarmente importante per i nostri scopi è

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \ , \tag{1.5.1}$$

chiamata **matrice di Hadamard**. Notiamo che è unitaria in quanto $H^\dagger H = \mathbb{I}$. Essa può essere implementata nel cosiddetto `H-gate` o **gate di Hadamard**: si tratta di un gate particolarmente importante (lo useremo largamente durante tutto il corso) in quanto permette di cambiare base $\{|0\rangle, |1\rangle\} \leftrightarrow \{|+\rangle, |-\rangle\}$

[iv] Un tale sistema in natura è abbastanza semplice da implementare poiché, essendo $\hat{H} = \vec{\sigma}\cdot\vec{B}$ l'accoppiamento tra spin e campo magnetico, è facile costruire una tale evoluzione temporale.



Possiamo introdurre anche le matrici seguenti (ci serviranno più avanti)

$$S \equiv \sqrt{Z} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} , \qquad T \equiv \sqrt{S} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix} , \tag{1.5.2}$$

Le matrici introdotte in precedenza costituiscono gli oggetti base con cui andremo a implementare diversi gate e circuiti durante tutto il corso. Per costruire il gate più generale possiamo esponenziare scrivendo $U = e^{-\frac{i}{\hbar}Ht}$ dove $H = a\mathbb{I} + b_i\sigma_i$ e $a, b_i \in \mathbb{R}$ con $i = 1, 2, 3$. In particolare esiste una particolare classe di operatori che utilizzeremo molto

$$R_{\vec{n}} = e^{-i\frac{\lambda}{2}(\vec{n}\cdot\vec{\sigma})} ;$$

si tratta di un caso particolare dell'esponenziazione precedente in cui $a = 0$ e i coefficienti b_i sono scelti lungo un particolare versore $\vec{n}$. Questo operatore unitario implementa una rotazione di angolo λ in una direzione particolare individuata da $\vec{n}$:

$$R_{\vec{n}} = e^{-i\frac{\lambda}{2}(\vec{n}\cdot\vec{\sigma})} = \cos\left(\frac{\lambda}{2}\right)\mathbb{I} - i\sin\left(\frac{\lambda}{2}\right)\vec{\sigma}\cdot\vec{n} ; \tag{1.5.3}$$

(si espanda il LHS con la serie di Taylor dell'esponenziale e si usi $(\vec{\sigma}\cdot\vec{n})^2 = \mathbb{I}$ per dimostrare l'uguaglianza con il RHS). È possibile dimostrare, inoltre, che qualsiasi matrice unitaria 2×2 può essere scritta nella forma seguente

$$U = e^{i\alpha}\begin{pmatrix} e^{-i\frac{\beta}{2}} & 0 \\ 0 & e^{i\frac{\beta}{2}} \end{pmatrix}\begin{pmatrix} \cos\frac{\gamma}{2} & -\sin\frac{\gamma}{2} \\ \sin\frac{\gamma}{2} & \cos\frac{\gamma}{2} \end{pmatrix}\begin{pmatrix} e^{-i\frac{\delta}{2}} & 0 \\ 0 & e^{i\frac{\delta}{2}} \end{pmatrix} = e^{i\alpha}R_z(\beta)R_x(\gamma)R_z(\delta) ; \tag{1.5.4}$$

perciò il più generale operatore unitario presenta 4 parametri reali $\alpha, \beta, \gamma, \delta \in \mathbb{R}$ e può implementare un possibile gate in un computer quantistico. Appare subito evidente come la scelta di 4 possibili parametri reali sia molto maggiore che nel caso dei gate logici classici.

Lezione 3 - 11/10/2021

1.6 Sistemi a più qubit

Enunciamo l'ultimo postulato della QM riguardante i sistemi composti da diversi sottosistemi:

- **V Postulato** (**Sistemi multi-partiti**): Consideriamo un sistema quantistico composto da 2 sottosistemi A e B con spazi di Hilbert $\mathcal{H}_A$ e $\mathcal{H}_B$ rispettivamente. Lo spazio di Hilbert del sistema totale è dato da[v] $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$.

Esempio 1.3 (**Sistema di 2 qubit**). *Immaginiamo un sistema quantistico costituito da 2 qubit tale che ogni qubit possa esistere in uno stato differente, ossia* $|0\rangle$ *o* $|1\rangle$*. Ovviamente ci sono 4 possibili scelte per il sistema totale:* $|00\rangle$*,* $|01\rangle$*,* $|10\rangle$ *e* $|11\rangle$*, dove la notazione*

[v] Il simbolo "$\otimes$" indica il **prodotto tensoriale** tra spazi.

implica che la prima entrata si riferisca al primo qubit e la seconda al secondo qubit. Il generico stato del sistema è dato da una combinazione lineare di questi ultimi:

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

con la condizione di normalizzazione $\sum_{i,j=0}^{1} |\alpha_{ij}|^2 = 1$. *La probabilità che il primo qubit si trovi in* $|i\rangle$ *e al contempo il secondo si trovi in* $|j\rangle$ *è data da* $P(|ij\rangle) = |\alpha_{ij}|^2$. *Se introduciamo i proiettori* $\hat{P}_0$ *e* $\hat{P}_1$ *sul risultato del primo qubit, possiamo allora scrivere che*

$$|\psi\rangle = \underbrace{\alpha_{00}|00\rangle + \alpha_{01}|01\rangle}_{\hat{P}_0|\psi\rangle} + \underbrace{\alpha_{10}|10\rangle + \alpha_{11}|11\rangle}_{\hat{P}_1|\psi\rangle} = \hat{P}_0|\psi\rangle + \hat{P}_1|\psi\rangle \ ;$$

dalla regola di Born generalizzata avremo che

$$P_1(|0\rangle) = \left\|\hat{P}_0|\psi\rangle\right\|^2 = |\alpha_{00}|^2 + |\alpha_{01}|^2 \,, \quad \Rightarrow \quad |\psi\rangle \to \frac{\hat{P}_0|\psi\rangle}{\left\|\hat{P}_0|\psi\rangle\right\|} \,,$$

$$P_1(|1\rangle) = \left\|\hat{P}_1|\psi\rangle\right\|^2 = |\alpha_{10}|^2 + |\alpha_{11}|^2 \,, \quad \Rightarrow \quad |\psi\rangle \to \frac{\hat{P}_1|\psi\rangle}{\left\|\hat{P}_1|\psi\rangle\right\|} \,.$$

Più in generale, consideriamo il caso di due spazi di Hilbert generici $\mathcal{H}_A$ e $\mathcal{H}_B$ con basi $|n\rangle_A$ e $|n\rangle_B$ rispettivamente. Possiamo scrivere lo spazio di Hilbert totale come

$$\mathcal{H}_A \otimes \mathcal{H}_B = \left\{ \sum_{n,m} \alpha_{nm} |nm\rangle \right\} \,,$$

dove $\dim(\mathcal{H}_A \otimes \mathcal{H}_B) = (\dim \mathcal{H}_A) \times (\dim \mathcal{H}_B)$. Questo spazio possiede la seguente operazione

Definizione 1.4 (Prodotto Tensoriale). *Siano* $\{|n\rangle_A\}$ *e* $\{|n\rangle_B\}$ *i due set di basi di due spazi di Hilbert* $\mathcal{H}_A$ *e* $\mathcal{H}_B$. *Definiamo* ***prodotto tensoriale*** *la funzione* $\otimes : \mathcal{H}_A \times \mathcal{H}_B \to \mathcal{H}_A \otimes \mathcal{H}_B$ *con la regola*

$$\{|n\rangle_A, |n\rangle_B\} \to |n\rangle_A \otimes |n\rangle_B \equiv |nm\rangle \ .$$

Questa operazione può essere estesa per linearità su tutto lo spazio di Hilbert: consideriamo due stati $|\psi\rangle_A \in \mathcal{H}_A$ e $|\phi\rangle_B \in \mathcal{H}_B$, allora

$$|\psi\rangle_A \otimes |\phi\rangle_B = \left(\sum_n \alpha_n |n\rangle_A \right) \otimes \left(\sum_m \beta_m |m\rangle_B \right) = \sum_{n,m} \underbrace{\alpha_n \beta_m}_{\alpha_{nm}} |nm\rangle \ . \tag{1.6.1}$$

Vettori di $\mathcal{H}_A \otimes \mathcal{H}_B$ che possono essere scritti come la decomposizione in (1.6.1) sono detti **separabili**. Solamente alcuni vettori di $\mathcal{H}_A \otimes \mathcal{H}_B$ sono separabili perché α_{nm} è una matrice particolare, risultante dal prodotto dei due vettori contenenti i coefficienti α_n e β_m. È possibile dimostrare che tali matrici che si scrivono come $A_{nm} = \alpha_n \beta_m$ hanno $\det A_{nm} = 0$ (sono di rango 1). Gli stati particolarmente interessanti che tratteremo a lungo durante il corso sono quelli che non soddisfano la decomposizione (1.6.1), detti stati **entangled**.

Esempio 1.4. *Il seguente stato è separabile:*

$$\frac{1}{\sqrt{2}}(|00\rangle + |01\rangle) \equiv \frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |0\rangle \otimes |1\rangle) = |0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \equiv |\psi\rangle_A \otimes |\phi\rangle_B \ .$$

Al contrario lo stato $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ *è entangled poiché non può essere decomposto come in* (1.6.1).

Analizziamo ciò che abbiamo appena visto nel contesto di qubit e circuiti. In qualità di portatori di informazioni denoteremo due qubit con due linee in questo modo

$$\begin{array}{l} |0\rangle, |1\rangle \ \text{————} \\ |0\rangle, |1\rangle \ \text{————} \end{array}$$

mentre il risulto di tali qubit è la combinazione lineare $\sum_{n,m} \alpha_{nm} |nm\rangle$. Dato che $\dim(\mathcal{H}_A \otimes \mathcal{H}_B) = 2 \times 2 = 4$ allora $\mathcal{H}_A \otimes \mathcal{H}_B \simeq \mathbb{C}^4$ e su tale spazio di Hilbert gli operatori unitari corrispondenti ai gate quantistici sono le matrici unitarie 4×4. Alcune di queste matrici derivano dalle operazioni sui qubit singoli: ad esempio se

$$\begin{array}{l} |\psi\rangle \ \text{——}\boxed{A}\text{——} \ A|\psi\rangle \ , \\ |\phi\rangle \ \text{——}\boxed{B}\text{——} \ B|\phi\rangle \ , \end{array}$$

la matrice risultante agente se entrambi i qubit è $U = A \otimes B$, la quale agisce naturalmente come $U(|\psi\rangle \otimes |\phi\rangle) = A|\psi\rangle \otimes B|\phi\rangle$. Queste tipologie di matrici sono molto speciali e non sono sicuramente le più generali.
Come facciamo per passare dai vettori a 2 componenti di ciascun qubit ad un vettore a 4 componenti del sistema a 2 qubit? Supponiamo che ciascun qubit sia un vettore

$$\alpha_0 |0\rangle + \alpha_1 |1\rangle \equiv \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \in \mathbb{C}^2 ,$$

e consideriamo il vettore risultante dal sistema di due qubit

$$\alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle \equiv \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix} \in \mathbb{C}^4 .$$

Per passare dall'uno all'altro possiamo utilizzare il prodotto seguente:

Definizione 1.5 (**Prodotto di Kronecker**)**.** *Siano A e B due matrici. Assumendo che A sia una matrice* $q \times p$ *allora definiamo il prodotto di Kronecker* $A \otimes B$ *come*

$$A \otimes B = \begin{pmatrix} A_{11}B & A_{12}B & \cdots & A_{1p}B \\ \vdots & & \ddots & \\ A_{q1}B & A_{q2}B & \cdots & A_{qp}B \end{pmatrix}$$

Esempio 1.5 (Vettore sistema di due qubit)**.** *Utilizzando il prodotto di Kronecker il vettore risultante di un sistema di due qubit può essere scritto come*

$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \otimes \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} = \begin{pmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_0 \\ \alpha_1\beta_1 \end{pmatrix} ;$$

notiamo infatti che questo vettore di coefficienti è lo stesso risultante dal prodotto tensoriale dei due qubit

$$(\alpha_0 |0\rangle + \alpha_1 |1\rangle) \otimes (\beta_0 |0\rangle + \beta_1 |1\rangle) = \alpha_0\beta_0 |00\rangle + \alpha_0\beta_1 |01\rangle + \alpha_1\beta_0 |10\rangle + \alpha_1\beta_1 |11\rangle \ .$$

Esempio 1.6. *Chiaramente il prodotto di Kronecker funziona anche per un sistema di due qubit in cui agiscono un* `X-gate` *e un* `Z-gate`

$$\begin{array}{l} |\psi\rangle \ \text{——}\boxed{X}\text{——} \ X|\psi\rangle \ , \\ |\phi\rangle \ \text{——}\boxed{Z}\text{——} \ Z|\phi\rangle \ , \end{array}$$

la matrice 4×4 *risultante sarà*

$$X \otimes Z = \sigma_1 \otimes \sigma_3 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}$$

Così come sottolineato in precedenza per gli stati separabili, queste matrici non costituiscono il caso generale perché essendo frutto di un prodotto tensoriale in realtà agiscono separatamente su un singolo qubit alla volta: quello di cui abbiamo bisogno in QC ("quantum computing") è di utilizzare matrici generiche che rendono i qubit entangled.
Cominciamo a vedere qualche esempio nel caso del CC ("classical computing"). Oltre al gate logico `NOT`, vediamo l'azione di altri gate

$$x \ \texttt{AND} \ y = \begin{cases} 1\,, \ \text{se entrambi } x = y = 1 \\ 0\,, \ \text{altrimenti} \end{cases} \ ,$$

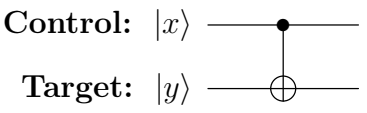
$$x \ \texttt{OR} \ y = \begin{cases} 1\,, \ \text{se } x = 1 \text{ oppure } y = 1 \\ 0\,, \ \text{altrimenti} \end{cases} \ ,$$

$$x \ \texttt{XOR} \ y = (x + y) \bmod 2 \equiv x \oplus y = \begin{cases} 0 \oplus 0 = 0 \\ 0 \oplus 1 = 1 \\ 1 \oplus 0 = 1 \\ 1 \oplus 1 = 0 \end{cases} \ ,$$

dove l'operazione $(x + y)$mod 2 indica il resto della divisione per 2 della somma $x + y$. Tutte queste operazioni non possono essere fattorizzate perché non sono il prodotto di operazioni sui singoli bit.
Consideriamo ora il caso quantistico. Una delle operazioni più importanti è il **controlled NOT** o `CNOT-gate`, il quale è una sorta di generalizzazione al caso quantistico del classico `XOR`. Si tratta di un gate U_{CN} (unitario) che agisce su un sistema di 2 qubit (sistema di dimensione 4) come segue:

$$\begin{array}{lll} U_{\text{CN}}: & |00\rangle \to |00\rangle \ , & |01\rangle \to |01\rangle \ , \\ & |10\rangle \to |11\rangle \ , & |11\rangle \to |10\rangle \ , \end{array}$$

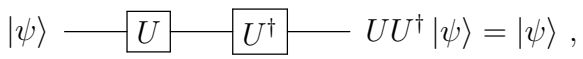
Come evidente, il primo qubit è utilizzato come **target**, mentre il secondo funge da qubit di **controllo**: a seconda del valore del primo qubit si svolge o meno un'azione sul secondo. In particolare quando il primo qubit è in $|0\rangle$, il secondo non viene toccato; quando invece il primo si trova in $|1\rangle$, il secondo viene scambiato. Dal punto di vista grafico indicheremo il `CNOT-gate` come



È chiaro che per $x = 0$ si ha $y \to 0 \oplus y = y$ mentre per $x = 1$ si ha $y \to 1 \oplus y$, il quale risulta 1 per $y = 0$ e 0 per $y = 1$. In forma matriciale questo gate non è altro che

$$U_{\text{CN}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} ,$$

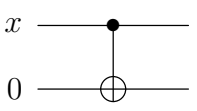
Si tratta di un esempio di gate (matrice 4×4) che agisce in maniera non banale sui qubit e che non può essere fattorizzato come azione sui singoli qubit.
Un punto fondamentale che differenzia i gate quantistici da quelli classici è che, essendo unitari, sono sempre **reversibili**. Infatti



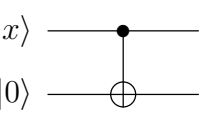
In generale il CC non è reversibile[vi]: ad esempio se si ha che x `AND` $y = 0$ non possiamo dire nulla su x e y separatamente.

1.7 Teorema di no-cloning

Il teorema di no-cloning è un risultato molto importante in QM perché stabilisce che cosa è permesso fare o meno in un computer quantistico. Discutiamolo in dettaglio. Supponiamo di considerare lo stato $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$: anche se, a differenza del caso classico, i coefficienti α e β sono arbitrari, il problema risiede nell'estrarre informazioni. Per trovare in quale stato preciso si trova il sistema bisogna considerare un numero infinito di qubit tutti preparati nel medesimo stato iniziale, ma in un computer quantistico tipicamente si ha solamente 1 singolo qubit! Il punto è che c'è moltissima informazione in $|\psi\rangle$ ma non sappiamo come estrarla: infatti l'arte del quantum computing consiste nell'estrarre *particolari* informazioni riguardanti α e β usando solamente una sola misurazione.
Un problema come questo può essere risolto se potessimo duplicare gli stati: il teorema di no-cloning stabilisce proprio il fatto che ciò non è possibile. Nonostante ciò la questione è sottile quindi vediamo di analizzarla in dettaglio.
Nel CC è possibile clonare un bit utilizzando un `CNOT-gate` classico: per $y = 0$ infatti (indichiamo nel circuito seguente i singoli bit e non gli stati perché è un circuito classico)



siamo quindi riusciti ad ottenere due copie esatte del bit x. Nell'analogo caso quantistico abbiamo



[vi] In realtà esiste un modo per calcolare solamente operazioni reversibili mediante gate reversibili in un computer classico. Lo vedremo tra qualche lezione. Si tratta di convertire le operazioni base dell'aritmetica in calcoli step by step reversibili.

perciò sembrerebbe che siamo riusciti a clonare anche lo stato quantistico. Ad esempio per $x = 0$ si ha $|00\rangle \overset{\texttt{CNOT}}{\to} |00\rangle$ e per $x = 1$ si ha $|10\rangle \overset{\texttt{CNOT}}{\to} |11\rangle$. Da questi esempi si potrebbe ingenuamente concludere che sia possibile clonare uno stato anche nel mondo quantistico. In realtà la questione è più sottile perché è possibile clonare uno stato che appartiene ad una base, come $\{|0\rangle, |1\rangle\}$, ma è molto semplice vedere che non si può clonare uno stato generico. Ad esempio, prendiamo il caso in cui vogliamo clonare il generico qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Applichiamo un `CNOT-gate` con un target dato da $|0\rangle$:

$$(\alpha|0\rangle + \beta|1\rangle) \otimes |0\rangle = \alpha|00\rangle + \beta|10\rangle \overset{\texttt{CNOT}}{\to} \alpha|00\rangle + \beta|11\rangle ,$$

questo risultato è chiaramente differente da

$$|\psi\rangle \otimes |\psi\rangle = \alpha^2|00\rangle + \alpha\beta|01\rangle + \beta\alpha|10\rangle + \beta^2|11\rangle ,$$

nel caso generale in cui $\alpha \neq 0$ e $\beta \neq 0$. In generale è possibile clonare solamente stati ortogonali tra loro. Enunciamo il teorema:

Teorema 1.2 (**No-cloning**). *Dato uno stato generico $|\psi\rangle$ normalizzato, non esiste alcun operatore unitario U tale che*

$$|\psi\rangle \otimes |\phi\rangle \to U(|\psi\rangle \otimes |\phi\rangle) = |\psi\rangle \otimes |\psi\rangle , \quad \forall\, |\psi\rangle . \tag{1.7.1}$$

Dimostrazione. Supponiamo per assurdo che esista un operatore unitario U che realizza la trasformazione (1.7.1). Supponiamo di aver clonato due generici stati $|\psi_1\rangle$ e $|\psi_2\rangle$:

$$|\psi_1\rangle \otimes |\phi\rangle \overset{U}{\to} |\psi_1\rangle \otimes |\psi_1\rangle ,$$
$$|\psi_2\rangle \otimes |\phi\rangle \overset{U}{\to} |\psi_2\rangle \otimes |\psi_2\rangle ,$$

ma dato che l'operatore è unitario deve preservare il prodotto scalare:

$$(\langle\psi_2| \otimes \langle\phi|)(|\psi_1\rangle \otimes |\phi\rangle) \overset{?}{=} (\langle\psi_2| \otimes \langle\psi_2|)(|\psi_1\rangle \otimes |\psi_1\rangle) ,$$
$$\Rightarrow \quad \langle\psi_2|\psi_1\rangle = \langle\psi_2|\psi_1\rangle^2 ,$$

dove abbiamo assunto $\langle\phi|\phi\rangle = 1$. Questo significa che $\langle\psi_2|\psi_1\rangle(1 - \langle\psi_2|\psi_1\rangle) = 0$, quindi $\langle\psi_2|\psi_1\rangle = 0$ (i due stati sono ortogonali) oppure $\langle\psi_2|\psi_1\rangle = 1$, il che significa che $|\psi_1\rangle = e^{i\alpha}|\psi_2\rangle$, quindi i due stati sono proporzionali per una fase e quindi di fatto si tratta dello stesso stato. In definitiva non è possibile clonare stati generici, ma solamente stati particolari. □

Dobbiamo capire come superare queste limitazioni in un computer quantistico. In realtà vedremo che per la maggior parte delle situazioni fisicamente rilevanti è già abbastanza clonare solamente degli stati che appartengono ad una base, quindi non sarà necessario duplicare stati generici.

Capitolo 2

Entanglement

In questo capitolo affronteremo molti argomenti riguardanti il concetto di *teletrasporto*, *crittografia quantistica* e *disuguaglianze di Bell*. Tutti questi temi sono legati dal fenomeno dell'**entanglement**. Innanzitutto ricordiamo che uno stato è definito **entangled** se **non** può essere scritto come stato separabile $|\psi\rangle = |\psi\rangle_A \otimes |\psi\rangle_B$, ovvero non è frutto del prodotto tensoriale di stati appartenenti a differenti spazi di Hilbert.

Esempio 2.1. *Per un sistema in cui è presente una particella dotata di spin, lo stato di singoletto*[i] $|00\rangle = \frac{1}{\sqrt{2}}(|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle)$ *(momento angolare totale nullo) è uno stato entangled.*

Gli stati entangled danno origine a numerosi paradossi che sono stati studiati a partire dall'inizio del '900. Il paradosso più famoso è probabilmente il **paradosso EPR** (dai nomi Einstein-Podolski-Rosen).

Esempio 2.2 (**Paradosso EPR**)**.** *Consideriamo un sistema costituito da 2 qubit (o due particelle dotate di spin) che si trova nello stato entangled* $|\psi\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$ *e immaginiamo di voler effettuare una misurazione sul primo qubit (spin). Secondo le regole della QM, semplicemente avremo che*

$$P(|0\rangle_1) = \frac{1}{2}\,, \qquad P(|1\rangle_1) = \frac{1}{2}\,,$$
$$P(|0\rangle_2) = \frac{1}{2}\,, \qquad P(|1\rangle_2) = \frac{1}{2}\,,$$

dove il pedice numerico sui ket indica il qubit (spin) che è stato misurato. Dalla forma dello stato $|\psi\rangle$ *notiamo che le misure sono correlate perché una misura sul sistema causa il collasso dello stato in* $|01\rangle$ *o* $|10\rangle$ *e quindi il risultato della misura sull'altro qubit viene direttamente influenzato. La correlazione è sottile perché se supponiamo di separare le due particelle (due qubit) in due città differenti mantenendo lo stato totale entangled allora è possibile determinare il risultato di entrambe le misure effettuando una singola misurazione! Ad esempio, se una misura sul primo qubit produce* $|0\rangle_1$ *allora* $|\psi\rangle$ *collassa istantaneamente in* $|01\rangle$ *(ora lo stato non è più entangled): d'ora in avanti il secondo sperimentatore che si trova nell'altra città trova sempre* $P(|1\rangle_2) = 1$*, sebbene prima del collasso vedeva le probabilità equiprobabili.*

Chiaramente ciò che accade nel paradosso EPR è molto "strano": il fatto che lo stato sia entangled suggerisce una sorta di azione istantanea a distanza. È importante evidenziare

[i] In generale non è difficile realizzare un tale stato in natura. Si vedano ad esempio l'ortoelio e il paraelio.

che questo paradosso non ha nulla a che fare con la violazione della relatività speciale perché nessuno dei due sperimentatori ha inviato informazioni istantanee. Uno dei due aspetti che il paradosso vuole sottolineare è la violazione del **principio di località**: una misura effettuata in una regione non può influenzare istantaneamente una misura che viene effettuata in un'altra regione casualmente disconnessa alla precedente. Ma come l'evidenza sperimentale mostra, il risultato è l'esatto contrario.

Lezione 4 - 15/10/2021

Continuiamo la discussione riguardante il concetto di entanglement. Dato che questo fenomeno si manifesta in sistemi con almeno due qubit, possiamo utilizzare due differenti basi:

- **Base computazionale** (o **standard**): formata dagli stati $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ (la due entrate indicano gli stati del primo e secondo qubit rispettivamente).
- **Base di Bell** (o **EPR**): costituita dagli stati

$$|\beta_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \qquad |\beta_{01}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle),$$
$$|\beta_{10}\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle), \qquad |\beta_{11}\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle);$$

  notiamo che si trattano di stati entangled costruiti a partire da combinazioni lineari indipendenti degli stati della base computazionale.

Chiaramente, trattandosi di una base, possiamo espandere qualsiasi stato $|\psi\rangle$ nella base EPR, scrivendo

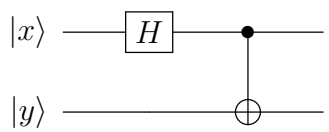
$$|\psi\rangle = \sum_{n,m=0}^{1} \alpha_{nm} |\beta_{nm}\rangle .$$

Conseguentemente, se si cerca la probabilità di trovarsi in $|\beta_{nm}\rangle$ si può effettuare una misurazione nella base di Bell e ottenere $P(|\beta_{nm}\rangle) = |\alpha_{nm}|^2$.
Gli stati della base di Bell non sono difficili da costruire utilizzando i gate che abbiamo visto nelle lezioni precedenti. Supponiamo di poter utilizzare un computer quantistico i cui qubit si trovano nella base standard $\{|0\rangle, |1\rangle\}$. Utilizziamo l'`H-gate` e il `CNOT-gate`:

- Ricordiamo che $H|0\rangle = |+\rangle$ e $H|1\rangle = |-\rangle$, quindi il gate di Hadamard permette di passare da un qubit nella base computazionale ad un qubit in una combinazione lineare di elementi di questa base (si ricordi la matrice (1.5.1) e le (1.2.2)).
- Il `CNOT-gate`, invece, scambia il secondo qubit solamente se il primo si trova in $|1\rangle$:

$$|00\rangle \xrightarrow{\texttt{CNOT}} |00\rangle , \qquad |01\rangle \xrightarrow{\texttt{CNOT}} |01\rangle ,$$
$$|10\rangle \xrightarrow{\texttt{CNOT}} |11\rangle , \qquad |11\rangle \xrightarrow{\texttt{CNOT}} |10\rangle .$$

Utilizzando questi due gate possiamo facilmente implementare il circuito seguente



i cui output sono esattamente gli stati $|\beta_{xy}\rangle$ della base di Bell. Verifichiamolo:

$$|00\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \xrightarrow{\texttt{CNOT}} \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \equiv |\beta_{00}\rangle \ ,$$
$$|01\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |1\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |11\rangle) \xrightarrow{\texttt{CNOT}} \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \equiv |\beta_{01}\rangle \ ,$$
$$|10\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |10\rangle) \xrightarrow{\texttt{CNOT}} \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \equiv |\beta_{10}\rangle \ ,$$
$$|11\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \otimes |1\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |11\rangle) \xrightarrow{\texttt{CNOT}} \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \equiv |\beta_{11}\rangle \ ;$$

si noti come l'azione di un gate (in questo caso l'`H-gate`) su un singolo qubit non basti per produrre uno stato entangled. Al contrario il `CNOT-gate`, invece, crea stati entangled poiché agisce su coppie di qubit.
Vediamo ora due esplicite applicazioni dell'entanglement.

2.1 Superdense coding

Si tratta del primo esempio esplicito delle potenzialità dei metodi quantistici contro i metodi classici. Il problema riguarda il come inviare informazioni di due bit classici (00, 01, 10, 11) ad un generico sperimentatore. Consideriamo la sperimentatrice Alice e supponiamo che abbia due bit di informazione (due numeri xy) e che voglia inviarli allo sperimentatore Bob. Dal punto di vista classico, Alice semplicemente utilizza un canale classico (un telefono ad esempio) per comunicare direttamente a Bob quale coppia di numeri possiede. Nel caso in cui Alice possieda due qubit, invece, può inviare solamente uno dei due sfruttando il fatto che siano entangled. Supponiamo che Alice e Bob condividano due qubit entangled, ad esempio

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = |\beta_{00}\rangle \ ,$$

dove Alice possiede il primo qubit (prima entrata del ket) e Bob il secondo. Cosa deve fare Alice per inviare solamente un singolo "pezzo" di informazione ? Ad esempio Alice può effettuare una qualche operazione sul suo qubit e, sfruttando l'entanglement, Bob sarà in grado di leggere l'informazione desiderata (la coppia di numeri xy che Alice vuole inviare) a seguito del collasso dello stato. Più precisamente, supponiamo che Alice voglia inviare delle informazioni effettuando le seguenti operazioni sul proprio qubit di $|\psi\rangle$:

$$\text{Alice invia: } \begin{cases} 00\,, & \text{non fa niente} \\ 10\,, & \text{applica } Z \\ 01\,, & \text{applica } X \\ 11\,, & \text{applica } ZX \end{cases} .$$

Cosa succede allo stato condiviso quando applica queste operazioni ?

- Quando vuole inviare 00 non effettua alcuna operazione quindi $|\psi\rangle \to |\psi\rangle = |\beta_{00}\rangle$.
- Nel caso in cui decide di inviare 10 applica Z:

$$|\psi\rangle \to Z\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) = |\beta_{10}\rangle \ .$$

- Quando invece vuole inviare 01 applica X:

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

- Infine se vuole inviare 11 applica ZX:

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

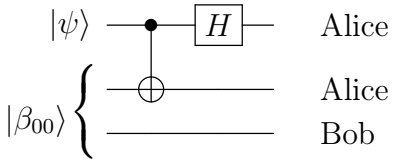
Effettuando queste operazioni, Alice è in grado di spedire ciò che vuole: $xy \to |\beta_{xy}\rangle$, quindi Bob può effettuare una misura nella base di Bell e stabilire quale dei 4 stati possiede. Qui risiede l'idea di **non-località** della QM: sebbene Bob possa trovarsi molto lontano da Alice, il suo qubit è cambiato a seguito delle operazioni di lei e può facilmente leggere il bit corrispondente xy con una singola misurazione.
Questo esempio, nonostante sia un po' accademico, risulta particolarmente interessante perché mette in risalto come, grazie all'entanglement, sia possibile ridurre il numero di operazioni necessarie per inviare un'informazione rispetto al caso classico.

2.2 Teleportation

Innanzitutto che cosa intendiamo con il termine "teletrasporto"? In questo contesto viene inteso con il significato di ricostruire un qubit molto lontano da dove si trovava in origine: il qubit originale sparisce e una sua nuova copia viene creata altrove. L'idea è quella di effettuare questa particolare ricostruzione usando solamente operazioni classiche sui qubit.
Supponiamo che Alice (d'ora in avanti chiameremo sempre in questo modo i nostri due sperimentatori) abbia un generico qubit $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ e che voglia inviarlo a Bob senza utilizzare alcun canale quantistico. Dalle leggi della QM sappiamo che non possiamo estrarre sia α sia β con una singola misura e inoltre non è possibile clonare questo stato generico. Inoltre, se volesse inviare direttamente questo stato con assoluta precisione (assumiamo $\alpha, \beta \in \mathbb{R}$) mediante un canale classico, allora necessiterebbe due stringhe infinite di bit e quindi del tempo infinito per inviarle. Come nel caso precedente, assumiamo che Alice e Bob condividano lo stato entangled $|\beta_{00}\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right)$, dove il primo qubit è di Alice e il secondo di Bob. Notiamo che Alice possiede due qubit: il qubit generico $|\psi\rangle$ che vuole teletrasportare e il qubit entangled con quello di Bob. Lo stato iniziale non è altro che

$$\left(\alpha\,|0\rangle + \beta\,|1\rangle\right)|\beta_{00}\rangle = \frac{\alpha}{\sqrt{2}}\,|0\rangle\left(|00\rangle + |11\rangle\right) + \frac{\beta}{\sqrt{2}}\,|1\rangle\left(|00\rangle + |11\rangle\right) \ . \tag{2.2.1}$$

Alice sottopone gli stati in suo possesso al seguente circuito:



dove si è indicato in output a chi appartiene quel determinato qubit. Esplicitamente, si applica `CNOT-gate` ai due qubit di Alice in (2.2.1):

$$\frac{\alpha}{\sqrt{2}}|0\rangle(|00\rangle+|11\rangle)+\frac{\beta}{\sqrt{2}}|1\rangle(|10\rangle+|01\rangle)\ ;$$

dopodiché viene applicato `H-gate` al primo qubit di Alice:

$$\frac{\alpha}{2}(|0\rangle+|1\rangle)(|00\rangle+|11\rangle)+\frac{\beta}{2}(|0\rangle-|1\rangle)(|00\rangle+|11\rangle)\ ;$$

infine possiamo riscrivere l'espressione nel modo seguente

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

dove in questo ultimo passaggio abbiamo svolto i conti e riordinato l'espressione focalizzandoci su ciò che è posseduto da Alice (i due qubit di fronte alle 4 parentesi tonde) e da Bob (stato nella parentesi tonda). Consideriamo ora la tabella 2.1: Alice può effettuare una misura nella base computazionale e dire a Bob (mediante un canale classico) ciò che ha ottenuto; dopo la misura lo stato collassa e Bob, a seconda del risultato, può effettuare o meno un'opportuna operazione sul proprio stato per ricostruire precisamente ciò che si voleva teletrasportare.

Alice misura	Bob trova	Bob applica
$\|00\rangle$	$\alpha\|0\rangle+\beta\|1\rangle$	Nulla
$\|01\rangle$	$\alpha\|1\rangle+\beta\|0\rangle$	X
$\|10\rangle$	$\alpha\|0\rangle-\beta\|1\rangle$	Z
$\|11\rangle$	$\alpha\|1\rangle-\beta\|0\rangle$	ZX

Tabella 2.1: Una volta che Alice effettua la propria misura nella base computazionale e dice a Bob ciò che ha ottenuto, quest'ultimo può applicare una precisa operazione per ricostruire lo stato $|\psi\rangle$ che Alice voleva teletrasportare. Si noti che, in tutti e quattro i casi, lo stato finale che ha Bob è sempre $|\psi\rangle$ indipendentemente dal risultato di Alice.

Un fatto fondamentale da evidenziare è che solamente informazioni classiche sono state trasferite tra Bob e Alice poiché tutto il resto (misurazioni e operazioni sugli stati) viene svolto localmente nel laboratorio: non c'è né violazione della relatività speciale in quanto non avviene alcun trasferimento di informazioni più veloce della luce, né violazione del teorema di no-cloning perché, una volta che Bob ottiene $|\psi\rangle$, Alice non possiede più lo stato che voleva teletrasportare. Si tratta solamente di un modo ingegnoso per sfruttare l'entanglement.

2.3 Disuguaglianze di Bell

L'argomento delle disuguaglianze di Bell è un tema molto vasto che comprende moltissime disuguaglianze testabili sperimentalmente: ciò che accomuna tutte le misurazioni è la profonda differenza tra il concetto di probabilità "classica" e "quantistica". Nella prima metà del '900, dopo la nascita della QM e i conseguenti trionfi che tale teoria

era in grado di riportare, molti fisici, tra cui lo stesso Einstein, erano profondamente insoddisfatti del concetto intrinseco ed inevitabile di probabilità che permea tale teoria. In particolare, coloro che non accettavano la QM come teoria completa, credevano che il suo comportamento fosse in realtà dovuto alla nostra ignoranza su teorie ancora più fondamentali. Questo gruppo di persone credevano che le **osservabili**, in fisica, dovessero sempre soddisfare 2 requisiti base:

- **Realismo**: un'osservabile deve avere un valore definito anche prima che la misura sia effettuata.

- **Località**: un esperimento effettuato in un ben preciso luogo ha solamente un effetto locale perché non può in alcun modo modificare risultati e comportamenti di altri esperimenti effettuati in regioni causalmente disconnesse. L'entanglement, ad esempio, è in profondo contrasto con il concetto di località.

Nel corso di quegli anni furono svolti numerosi tentativi di riscrivere la QM in maniera tale che soddisfacesse i requisiti precedenti. Ad esempio, furono utilizzate le cosiddette **teorie delle variabili nascoste**. Tali teorie si basano sull'assunto secondo cui quando si misura un valore a di un'osservabile A, in realtà il risultato della misurazione è incompleto perché l'intera teoria prevede l'esistenza di un'altra variabile λ *nascosta* ed inaccessibile. Se si potesse conoscere λ allora si potrebbe predire qualsiasi cosa in maniera del tutto deterministica. Nonostante ciò, il concetto di probabilità in QM vìola queste regole e, come vedremo tra poco, utilizzando le disuguaglianze di Bell è possibile rilevare sperimentalmente tale violazione.

Esempio 2.3 (Singolo qubit)**.** *Consideriamo nuovamente il caso di un qubit e immaginiamo di trovarci nello stato* $|0\rangle$*. Nella sezione 1.2 abbiamo visto che il più generale operatore hermitiano che agisce su un singolo qubit è dato da una combinazione lineare di matrici di Pauli (non consideriamo l'identità), ossia* $\vec{\sigma}\cdot\vec{n}$ *dove* $|\vec{n}| = 1$*. Supponiamo che a seguito di una misurazione possiamo ottenere* $\vec{\sigma}\cdot\vec{n}\,|\vec{n}\rangle = |\vec{n}\rangle$ *e* $\vec{\sigma}\cdot\vec{n}\,|-\vec{n}\rangle = -\,|-\vec{n}\rangle$*, quindi il risultato è* ± 1*. Perciò, ricordando la decomposizione* $|0\rangle = c_1\,|\vec{n}\rangle + c_2\,|-\vec{n}\rangle$*, la probabilità di misurare 1 è* $P(\vec{\sigma}\cdot\vec{n} = 1) = |c_1|^2 = |\langle 0|\vec{n}\rangle|^2$*. Al tempo stesso sappiamo anche che il generico qubit si scrive come in* (1.1.2)*, quindi* $\vec{n}$ *può essere specificato scegliendo gli angoli* θ *e* ϕ*: dato che avevamo sottolineato che* $\vec{\sigma}\cdot\vec{n}\,|\psi\rangle = |\psi\rangle$ *allora la soluzione che cerchiamo è* $|\vec{n}\rangle = |\psi\rangle$*, quindi*

$$P(\vec{\sigma}\cdot\vec{n} = 1) = |\langle 0|\vec{n}\rangle|^2 = |\langle 0|\psi\rangle|^2 = \cos^2\left(\frac{\theta}{2}\right) .$$

Vediamo se riusciamo a riprodurre la medesima distribuzione di probabilità utilizzando una teoria classica basata sulle variabili nascoste. Supponiamo che, oltre allo spin, la particella sia in realtà descritta da un'extra variabile λ*: tutte le particelle sono specificate dalla coppia fissata* $(a = \pm 1, \lambda)$*, ossia hanno spin* $a = \pm 1$ *e un preciso valore di* λ *persino prima di effettuare la misurazione. Per semplicità assumiamo* $\lambda \in [0, 1]$*. Supponiamo di voler effettuare una misurazione dello spin in una particolare direzione* $|n\rangle$*: la misurazione, in questa teoria, corrisponde a particolari valori di spin e* λ *con l'idea che una misura effettuata con angolo* θ *abbia risultato dipendente dal valore assunto da* λ *nell'intervallo* $[0, 1]$*. Più precisamente, assumendo di non poter rilevare il valore* λ *e richiedendo che le particelle abbiano dei valori di tale variabile uniformemente distribuiti, un esperimento*

di questo tipo produce

$$\begin{cases} spin\ |\uparrow\rangle\ , & 0 \leqslant \lambda \leqslant \cos^2\theta/2 \\ spin\ |\downarrow\rangle\ , & \cos^2\theta/2 \leqslant \lambda \leqslant 1 \end{cases}, \quad \Rightarrow \quad P(a=1) = \cos^2\left(\frac{\theta}{2}\right) .$$

Nell'esempio precedente è stato analizzato il caso di un singolo qubit, tuttavia prendiamo in esame il sistema di 2 qubit, dove sappiamo che l'entanglement gioca un ruolo centrale. Esistono diversi modi per scrivere delle disuguaglianze che testino sperimentalmente la profonda differenza tra probabilità "classica" e "quantistica". Uno dei più famosi è il seguente

2.3.1 Disuguaglianza CHSH

Si tratta di una semplice generalizzazione delle disuguaglianze di Bell utilizzata per testarle sperimentalmente. Ancora una volta, consideriamo i due sperimentatori Alice e Bob situati in città differenti. Supponiamo che entrambi abbiano a disposizione un apparato identico su cui possano effettuare misure e che ne possiedano infinite copie sulle quali possono compiere dei test. Alice misura le osservabili a, a', mentre Bob controlla b, b', dove $a, a', b, b' = \pm 1$. Entrambi scelgono di fare misurazioni simultanee di una di queste osservabili.
In un'ipotetica teoria basata sulle variabili nascoste, dato questo sistema è impossibile stabilire immediatamente quale sia il risultato di una misura poiché ci sarà necessariamente una distribuzione di probabilità classica, che chiamiamo $P(a, a', b, b')$, associata alla nostra ignoranza. Consideriamo ora l'osservabile $C = (a + a')b + (a - a')b'$; per costruzione sappiamo che

$$\begin{cases} a + a' = 0\,,\ a - a' = \pm 2\,, & \text{se } a \neq a' \\ a + a' = \pm 2\,,\ a - a' = 0\,, & \text{se } a = a' \end{cases},$$

ma questo significa allora che per qualsiasi valore delle 4 osservabili in gioco si ha sempre $C = \pm 2$. Dalla teoria della probabilità classica sappiamo che $|\langle C\rangle| \leqslant \langle |C|\rangle$ dato che $|\sum_c cp(c)| \leqslant \sum_c |c|p(c)$. Siccome assumiamo che C esista, applichiamo questa disuguaglianza: in tutte le possibili configurazioni $|C| = 2$ quindi $\langle |C|\rangle = 2$ e allora

$$|\langle C\rangle| \leqslant 2\,. \tag{2.3.1}$$

La disuguaglianza precedente prende il nome di **disuguaglianza CHSH**[ii]. Notiamo che si tratta di un risultato classico derivante dalla teoria della probabilità.
In QM è facile trovare un esempio nel quale questa disuguaglianza è violata. Supponiamo che Alice e Bob condividano lo stato entangled $|\psi\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$ e che entrambi decidano di misurare qualcosa che in QM abbia 2 possibili valori. In particolare misurano

$$a = \vec{\sigma}\cdot\hat{a} = \pm 1\,, \qquad a' = \vec{\sigma}\cdot\hat{a}' = \pm 1\,,$$
$$b = \vec{\sigma}\cdot\hat{b} = \pm 1\,, \qquad b' = \vec{\sigma}\cdot\hat{b}' = \pm 1\,,$$

dove il simbolo "ˆ" indica un vettore di modulo unitario. È possibile dimostrare in QM che

$$\langle\psi|(\vec{\sigma}\cdot\hat{c})\otimes(\vec{\sigma}\cdot\hat{d})|\psi\rangle = -\hat{c}\cdot\hat{d} = -\cos\theta\,, \tag{2.3.2}$$

[ii] Clauser, J., Horne, M., Shimony, A., & Holt, R. (1969). Proposed Experiment to Test Local Hidden-Variable Theories. Phys. Rev. Lett., 23, 880–884.

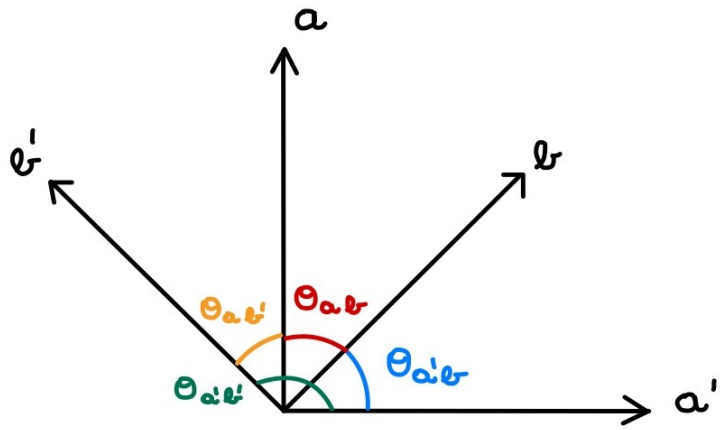


Figura 2.1: Direzioni spaziali delle 4 osservabili misurate da Alice e Bob. Si noti che l'apparato di uno è ruotato di 45° rispetto a quello dell'altro perciò $\theta_{a'b} = \theta_{ab} = \theta_{ab'} = 45°$ e $\theta_{a'b'} = 135°$.

dove θ è l'angolo tra $\hat{c}$ e $\hat{d}$. Supponiamo che Alice e Bob decidano di misurare nelle direzioni indicate dagli angoli di figura 2.1. Utilizzando la (2.3.2) possiamo calcolare il valore di aspettazione di C:

$$\langle C\rangle = \langle\psi|ab + a'b + ab' - a'b'|\psi\rangle = -\left[\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} - \left(-\frac{1}{\sqrt{2}}\right)\right] = -2\sqrt{2}\,.$$

Abbiamo quindi ricavato che, secondo la QM, $|\langle C\rangle| = 2\sqrt{2}$, in disaccordo[iii] con il risultato classico (2.3.1): la probabilità "quantistica" è intrinsecamente differente dalla probabilità "classica"!
Altri esperimenti degni di nota sono quelli condotti da Freedman e Clauser nel 1972, la serie di esperimenti condotti da Aspect negli anni 1981 e 1982, da Tittel e il gruppo Geneva nel 1988 e da Weihs sotto condizioni di località "strettamente einsteniane" nel 1998. La serie di esperimenti sulle disuguaglianze di Bell, di crescente sofisticazione, ha ridotto i critici, che mettono in discussione i risultati, a indicare falle in tale esperimenti, alcune delle quali distorcerebbero i risultati sperimentali in favore della meccanica quantistica. Nel 2015 è stato pubblicato il primo esperimento dichiarato totalmente privo di falle (loopholes), che ha confermato i risultati degli esperimenti precedenti.

[iii] In realtà esiste un teorema che stabilisce che $2\sqrt{2}$ è il più grande valore che può essere ottenuto.

Capitolo 3

Algoritmi quantistici

Lezione 5 - 18/10/2021

Prima di cominciare la vera discussione riguardante gli algoritmi più importanti e conosciuti della computazione quantistica, affrontiamo l'analisi della crittografia quantistica, la quale mostra ancora una volta la potenzialità dei metodi quantistici rispetto a quelli classici.

3.1 Crittografia quantistica

Molti anni prima che fu introdotta la crittografia RSA che utilizziamo oggigiorno, molte persone pensavano che gli stati quantistici e il "bizzarro" comportamento della QM potessero essere utilizzati per scopi crittografici. Il nome del protocollo quantistico che fu pensato per trasmettere dati criptati è **protocollo BB84**. Consideriamo, come al solito, Alice e Bob in differenti città e supponiamo che vogliano comunicare tra loro tramite una linea criptata. Vediamo come viene affrontato questo problemi sia dal punto di vista classico che quantistico.

3.1.1 Esempio di crittorafia classica

Classicamente entrambi possiedono una sequenza S di bit casuali, chiamata **codepad**. Immaginiamo che Alice voglia inviare a Bob un messaggio M: un modo classico di inviare il messaggio criptato è quello di inviare la sequenza $M \oplus S$, dove il simbolo "$\oplus$" indica l'**addizione bit a bit modulo 2**. Ad esempio supponiamo che il codepad sia $S = 0110$ e il messaggio sia $M = 1111$: la sequenza $M \oplus S$ è data da

S	0	1	1	0	$= 6$
M	1	1	1	1	$= 15$
$M \oplus S$	1	0	0	1	$= 9$

dove nell'ultima colonna abbiamo inserito il numero associato a quel messaggio (ad esempio, leggendo da destra verso sinistra, per S si ha $0 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 = 6$). Il vantaggio di questo modo di crittografare messaggi risiede nel fatto che anche se si

parte con una stringa M sensata, l'operazione $M \oplus S$ la trasforma in una sequenza apparentemente casuale di 0 e 1 che può essere decifrata solamente se si possiede il codepad. Infatti, una volta che Bob riceve $M \oplus S$, si può facilmente ricostruire il messaggio originale calcolando $(S \oplus M) \oplus S = M \oplus (S \oplus S) = M$ dato che

$$x \oplus x = \begin{cases} 0 \oplus 0 = 0 \\ 1 \oplus 1 = 0 \end{cases} \quad , \quad \forall\, x\,.$$

Il problema di un tale protocollo crittografico, oltre al fatto che il codepad non debba essere scoperto da nessun altro al di fuori di Alice e Bob, risiede nel fatto che non sia molto efficiente a seguito del cosiddetto "one-time codepad", dato che la stringa S può essere utilizzata una volta sola. Per capirne il motivo supponiamo che Alice voglia inviare entrambi i messaggi M_1 e M_2: il messaggio ricevuto da Bob è $(M_1 \oplus S) \oplus (M_2 \oplus S) = M_1 \oplus M_2$, il quale è costituito da una sequenza di 0 e 1 abbastanza randomica. Se Bob è abbastanza abile e conosce almeno una parte del messaggio che Alice voleva inviare allora ci sono possibilità che riesca a decifrare M_1 e M_2 separatamente riconoscendo degli opportuni schemi in $M_1 \oplus M_2$; tuttavia non è detto che chi riceva il messaggio sia sempre così abile !

3.1.2 Il protocollo BB84

La versione quantistica viene chiamata **protocollo BB84** ed è abbastanza simile al caso classico ma molto più potente: lo scopo è quello di creare un codepad S che non possa essere in alcun modo (o quasi[i]) intercettato da una terza persona, che chiameremo Eve, la quale vuole rovinare i piani di Alice e Bob. Il protocollo funziona come segue: Alice possiede una serie di qubit che vorrebbe inviare a Bob tramite un canale sicuro; invia allora casualmente dei qubit che sono preparati nella base computazionale $C = \{|0\rangle\,, |1\rangle\}$ oppure nella base di Hadamard $H = \{|+\rangle\,, |-\rangle\}$ (si vedano le (1.2.2)). Quindi Alice, prima di inviare i qubit, effettua due scelte: sceglie la base e poi sceglie uno stato di quella base da inviare. Nel frattempo, Bob riceve i qubit inviati e tiene attentamente conto dell'ordine di ricezione di questi qubit, dopodiché effettua una misurazione scegliendo randomicamente la base C oppure[ii] o la base H. Dato che Bob sceglie una delle due basi, per ogni qubit che riceve ci sono due possibilità:

- Sceglie la stessa base di Alice. Ad esempio se Alice avesse scelto $(C, |0\rangle)$ allora necessariamente, dai postulati della QM, sappiamo che Bob misura obbligatoriamente $(C, |0\rangle)$ con probabilità 1 (nel caso in cui nessuno abbia intercettato il messaggio).
- Sceglie una base differente da quella di Alice. Ad esempio se Alice invia $(C, |0\rangle)$ e Bob sceglie la base H allora sappiamo che ha il 50% di possibilità di trovare $|0\rangle$ nella propria misurazione.

Notiamo che i risultati ottenuti da Bob nelle proprie misurazioni non sono in alcun modo correlati con le informazioni che Alice vuole inviare. Riassumendo, gli step necessari sono i seguenti:

[i] Si sfrutta la natura probabilistica della QM quindi se i qubit inviati da Alice sono in gran numero, è solo una questione di tempo prima che una terza persona venga scoperta intercettare i messaggi. Si veda la discussione di seguito per chiarimenti più espliciti.

[ii] Ad esempio, se Alice invia delle particelle dotate di spin, Bob può scegliere, mediante un apparato simile a quello dell'esperimento di Stern e Gerlach, di misurare lo spin lungo la direzione z (base C) oppure lungo la direzione x (base H).

1. Alice sceglie una base e invia casualmente i qubit.

2. Bob riceve i qubit tenendo conto dell'ordine di arrivo e misura con il proprio apparato scegliendo casualmente una delle due basi.

3. Alice e Bob comparano **solamente le basi** di un numero arbitrario di qubit concordato a priori dai due (alcuni e non tutti perché in generale Alice potrebbe inviare un numero altissimo di qubit) tramite una linea non sicura (ossia che può essere intercettata da Eve). Questo significa che per ogni qubit che confontano, i due si scambiano la base in cui è stata effettuata la misura, non lo stato misurato.

4. Infine Bob, usando parte dei qubit inviati da Alice, ossia solamente quelli che ha misurato nella sua stessa base, può costruire un codepad comune e stabilire se qualcuno ha intercettato i qubit inviati (si veda la discussione dopo l'esempio 3.1) confrontando direttamente i qubit delle misure con la stessa base.

Per capire al meglio il funzionamento di questo meccanismo illustriamolo con un esempio.

Esempio 3.1. *Immaginiamo che le basi scelte e i risultati delle misurazioni effettuate da Alice e Bob siano quelli mostrati nella Tabella 3.1.*

Alice	base	→	C	H	H	C	C	H	C	H	C
	qubit	→	0	1	0	0	0	0	1	0	1
Bob	base	→	H	H	H	H	C	H	H	C	C
	qubit	→	1	1	0	0	0	0	1	1	1

Tabella 3.1: Basi scelte e rispettive misurazioni effettuate da Alice e Bob. Si noti che nelle righe dei qubit misurati si è indicato solamente il bit di informazione inviato da Alice o ottenuto da Bob, ossia: $|0\rangle, |+\rangle \to 0$ e $|1\rangle, |-\rangle \to 1$. Nella tabella sono state colorate in grigio le colonne corrispondenti alle misurazioni effettuate nella medesima base.

Una volta che Alice ha terminato[iii] *la sequenza di qubit che voleva inviare, comunica a Bob, mediante un canale classico, la sequenza di basi scelte, ossia la prima riga della tabella: dalle regole della QM sappiamo che ogniqualvolta che Bob sceglie (per coincidenza) la medesima base di Alice, il risultato della misurazione che ottiene è obbligatoriamente il medesimo qubit che Alice ha scelto di inviare (a tal proposito si vedano infatti le colonne colorate). Notiamo che per una coincidenza fortuita le misurazioni nelle colonne 4 e 7 sono le medesime sebbene la base scelta fosse differente: in questa situazione, ossia quando i due sperimentatori scelgono una base diversa, Bob ottiene casualmente 0 o 1 con probabilità* $1/2$*. Una volta effettuata la chiamata, i due decidono di tenere solamente i risultati in cui hanno scelto le stesse basi e formano con tali misure un codepad comune: nella situazione della Tabella 3.1, solo le colonne colorate hanno la stessa base, quindi il codepad non è altro che* $S = 10001$*. Ovviamente questo codepad è comune perchè Alice e Bob si sono scambiati, oltre alle basi, anche i qubit delle misure effettuate nella stessa base.*

[iii] In generale esistono varie versioni di questa procedura perché dal punto di vista pratico è molto difficile accumulare una sequenza di qubit mantenendoli tutti inalterati. Una versione alternativa più conveniente e realistica prevede Alice che invia il suo qubit e subito dopo comunica immediatamente la base che ha scelto, in maniera tale che una volta che Bob abbia ricevuto il qubit possa scartare quelli misurati in basi differenti.

Per quale ragione il codepad comune formato da Alice e Bob è più protetto di quello classico ? Come interviene Eve nella trasmissione delle informazioni per capire ciò che è stato inviato ? Eve può semplicemente intercettare il qubit durante il transito: dalla QM sappiamo che è obbligata ad effettuare una misurazione, la quale disturba inevitabilmente il sistema. Dato che Eve non conosce la base in cui il qubit è stato preparato è costretta a fare una scelta ! Nel caso in cui sia fortunata, scegliendo cioè la stessa base di Alice, Eve vede il qubit inviato senza modificare lo stato, tuttavia quando sceglie la base opposta ottiene un numero casuale 0 o 1 con probabilità 1/2 e causa il collasso dello in uno dei due stati della base utilizzata.
Capiamo meglio questo discorso con un esempio.

Esempio 3.2. *Supponiamo che Alice abbia scelto di inviare* $(C, |0\rangle)$ *e che Eve scelga di misurare nella base H ottenendo* $|+\rangle$*: lo stato è ora collassato in* $|+\rangle$*, quindi se Bob effettua una misurazione in C, egli può ottenere sia* $|0\rangle$ *sia* $|1\rangle$ *con probabilità* $1/2$*, nonostante Alice avesse inviato* $(C, |0\rangle)$*. Se dovesse succedere che Bob misuri* $(C, |1\rangle)$*, allora Alice e Bob concludono che hanno misurato due stati differenti, nonostante abbiano scelto la medesima base, ma questo è impossibile dalla QM se nessuno è intervenuto sullo stato !*

A seguito del collasso dello stato in uno stato di una base differente da quella scelta da Alice, può accadere che nelle colonne colorate della Tabella 3.1 (misurazioni con stesse basi) i due sperimentatori ottengano uno stato differente: se nessuno sta intercettando gli stati in transito questo è impossibile per le leggi della QM ! In questo modo, una volta comunicati i qubit misurati nelle stesse basi, Bob capisce che qualcuno ha interferito con i qubit che Alice sta inviando.
Statisticamente, quante volte Eve sta ascoltando sistematicamente il messaggio e vi è una possibilità che Alice e Bob non concordino su una misura effettuata nella stessa base ? Tipicamente per 1/4 delle volte. Il motivo è dato dal fatto che Eve può essere fortunata e misurare nella stessa base di Alice (probabilità 1/2 per questa scelta) e inoltre anche se Eve sceglie la base sbagliata, Bob deve effettuare una misurazione in cui ottiene lo stesso qubit di Alice il 50% delle volte: quindi $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$, dove il primo 1/2 deriva dalla scelta di Eve e il secondo dalla misura di Bob.

3.1.3 Quantum non-demolition measures

Chiaramente ci si potrebbe domandare se Eve possa fare di meglio. Esiste una possibilità in cui possa misurare senza recare alcun disturbo allo stato ? Delle volte queste misure vengono chiamate in letteratura **quantum non-demolition measures**: si trattano di particolari misure in cui Eve effettua la misurazione senza disturbare lo stato oppure disturba lo stato ma è in grado di resettarlo all'originale inviato da Alice. La risposta alla domanda precedente è no per un motivo simile alla dimostrazione del teorema di No-cloning.
Supponiamo che Alice stia inviando l'insieme di stati $|\phi_\mu\rangle = \{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$, dove $\mu = 0, 1, 2, 3$, e inoltre assumiamo che Eve possieda un proprio computer quantistico sul quale può effettuare operazioni. Nell'intercettare il messaggio, Eve osserva lo stato $|\phi_\mu\rangle \otimes |\phi\rangle$, dove $|\phi\rangle$ si trova nel suo computer. Supponiamo inoltre che nel suo computer ci sia un altro insieme di stati $|\psi_\mu\rangle$, con $\mu = 0, 1, 2, 3$, tale che possa essere distinto da una misura effettuata da Eve stessa. La domanda é: esiste qualche sorta di processo quantistico (gate unitario U) che agisce come

$$U\left(|\phi_\mu\rangle \otimes |\phi\rangle\right) = |\phi_\mu\rangle \otimes |\psi_\mu\rangle \ , \tag{3.1.1}$$

ossia tale che quando Eve misura $|\psi_\mu\rangle$ e legge il valore μ allora con probabilità 1 legge anche lo stesso μ che Alice sta inviando, senza però disturbare $|\phi_\mu\rangle$? La risposta è no, similmente al teorema di No-cloning. Per dimostrare questo fatto calcoliamo il prodotto scalare di ambo i membri della (3.1.1), il quale, come sappiamo a seguito dell'unitarietà di U, deve rimanere preservato:

$$(\langle\phi_\mu| \otimes \langle\phi|)\,(|\phi_\nu\rangle \otimes |\phi\rangle) \stackrel{?}{=} (\langle\phi_\mu| \otimes \langle\psi_\mu|)\,(|\phi_\nu\rangle \otimes |\psi_\mu\rangle)\ , \quad \text{con } \mu \neq \nu \text{ in generale.}$$
$$\Rightarrow \qquad \langle\phi_\mu|\phi_\nu\rangle \underbrace{\langle\phi|\phi\rangle}_{1} \stackrel{?}{=} \langle\phi_\mu|\phi_\nu\rangle\,\langle\psi_\mu|\psi_\nu\rangle\ ,\ \forall \text{ paia di indici } (\mu,\nu)\,.$$

Analizziamo i casi in cui $\mu \neq \nu$. Quando $(\mu = 2, \nu = 3)$ e $(\mu = 0, \nu = 1)$ (o viceversa) si ha l'identità $0 = 0$, che non è interessante (ricordare sopra gli stati $|\phi_\mu\rangle$ di Alice). Nei casi invece $(\mu = 0, \nu = 2)$, $(\mu = 0, \nu = 3)$, $(\mu = 1, \nu = 2)$ oppure $(\mu = 1, \nu = 3)$ (o viceversa) i prodotti scalari $\langle\phi_\mu|\phi_\nu\rangle$ non sono nulli e possono essere semplificati ad entrambi i membri. Per queste scelte otteniamo quindi che $\langle\psi_\mu|\psi_\nu\rangle = 1$, ossia $|\psi_\mu\rangle = |\psi_\nu\rangle$ a meno di una fase. Ma questo significa allora che $|\psi_0\rangle = |\psi_1\rangle = |\psi_2\rangle = |\psi_3\rangle$ e quindi, non essendo stati differenti, Eve non può in alcun modo distinguere ciò che ha inviato Alice.
La conclusione è che non esiste alcun modo di effettuare una misura con operazioni unitarie che distingua il qubit inviato da Alice senza necessariamente disturbare il sistema.

3.2 Proprietà dei gate

Nella sezione 1.5 abbiamo introdotto alcuni concetti preliminari riguardanti i gate, i circuiti e i computer quantistici. In molti casi nei computer si hanno degli algoritmi, ossia una ben precisa sequenza di istruzioni, che permettono di calcolare risultati desiderati. Approfondiamo le analogie e differenze dei gate classici e quantistici.

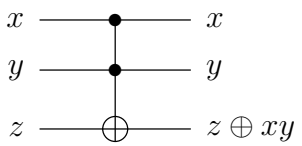
3.2.1 Gate classici: il `TOFFOLI-gate`

In CC si hanno i bit 0 e 1 e le funzioni classiche sono tali che $f:\ \{0,1\}^{\otimes n} \to \{0,1\}^{\otimes m}$, sono cioè mappe da n a m bit classici. Quindi in generale abbiamo

$$f_i(x_1, x_2, \ldots, x_n) = \{0,1\}\,,\ \text{ dove } i = 1, \ldots, n\,,\ \text{ e } x_i = 0, 1\,.$$

Si vorrebbe che il computer sia in grado di calcolare tali funzioni e che inoltre gli strumenti a disposizione siano sufficientemente efficienti per farlo: quello che uno vorrebbe è poter calcolare funzioni generali con l'ausilio di solamente pochi gate. In CC si ha che con le seguenti operazioni è possibile calcolare quasi tutti i conti di algebra e aritmetica: `NOT`, $a \to -a$; `AND`, indicato con $a \wedge b$ e `OR`, indicato con $a \vee b$. Questo insieme di operazioni è detto **universale** perché utilizzando questi pochi gate è possibile calcolare tutte le operazioni di aritmetica di interesse.
Sempre nella sezione 1.5 abbiamo osservato che il CC **non** é **reversibile**[iv] (in generale). Si pensi ad esempio all'`AND-gate`. Nel corso degli anni si è studiato numerosi metodi per implementare operazioni reversibili: questo è possibile mediante il cosiddetto **Toffoli gate** o `cont-cont-NOT`. Il circuito classico è

[iv] Si pensi ad esempio al fatto che le operazioni non reversibili dissipano calore all'interno della macchina. Si tratta di tutte quelle situazioni in cui si parte con molta informazione e si giunge alla fine ad un singolo risultato, creando nel frattempo numerosi risultati di scarto.



Quando uno dei due tra x e y é 0, allora xy è 0 e niente succede all'output di z. L'output si modifica solamente quando sono 1 perché controllano entrambi il risultato di z: quando $xy = 1$ allora $z \oplus 1$ inverte il valore iniziale di z.

Esempio 3.3 (Azione Toffoli gate)**.** *In pratica il* `TOFFOLI-gate` *agisce come mostrato in Tabella 3.2:*

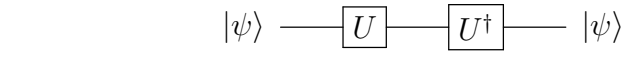
Bit iniziali			Bit finali		
x	y	z	x	y	$z \oplus xy$
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Tabella 3.2: Azione del `TOFFOLI-gate` su tutti i possibili bit.

E' possibile dimostrare che mediante l'uso del `TOFFOLI-gate` si possono realizzare tutte le operazioni base, quindi è universale e reversibile. Notiamo che è reversibile poiché, come evidenziato nelle ultime due righe della Tabella 3.2, esso agisce sulle stringhe facendo una **permutazione**, la quale è invertibile.

3.2.2 Gate quantistici: reversibili e continui

Consideriamo ora il caso dei gate quantistici. Per definizione, essendo implementati da operatori unitari, sono sempre dei gate **reversibili**. Questo significa ad esempio che



dato che $UU^\dagger = \mathbb{I}$. E' possibile implementare il `TOFFOLI-gate` anche in un computer quantistico ? La risposta è sì: consideriamo una base di stati per 3 qubit

$$\{|000\rangle, |001\rangle, |010\rangle, |100\rangle, |101\rangle, |011\rangle, |110\rangle, |111\rangle\}.$$

La matrice unitaria U_T 8×8 che agisce sul vettore contenenti gli stati della base è

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} |000\rangle \\ |001\rangle \\ |010\rangle \\ |100\rangle \\ |101\rangle \\ |011\rangle \\ |110\rangle \\ |111\rangle \end{pmatrix} = \begin{pmatrix} |000\rangle \\ |001\rangle \\ |010\rangle \\ |100\rangle \\ |101\rangle \\ |011\rangle \\ |111\rangle \\ |110\rangle \end{pmatrix}.$$

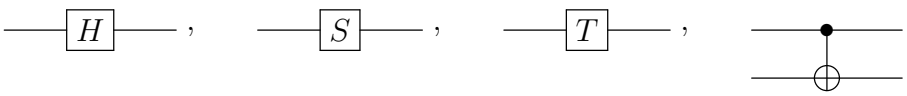
Notiamo infatti che U_T è unitaria ($U_T U_T^\dagger = \mathbb{I}$). Tramite U_T possiamo realizzare su un computer quantistico le stesse operazioni che faremmo su un computer classico. Fino ad ora non abbiamo ancora detto se effettivamente queste operazioni possano essere eseguite in maniera più efficiente su un QC.
Il secondo fatto importante dei gate quantistici è che sono **continui**: matrici unitarie possono dipendere da parametri reali continui. Ad esempio, per un sistema di 1 qubit abbiamo visto nella (1.5.4) come si scrive la più generale matrice 2×2 unitaria (gruppo $U(2)$) tramite l'implementazione delle rotazioni di angolo λ sulla sfera di Bloch e lungo la direzione generica $\vec{n}$ (si veda la (1.5.3)). Abbiamo visto che la (1.5.4) dipende in generale da 4 parametri reali arbitrari, quindi persino per un singolo qubit si ha un insieme continuo di gate ! Il problema è che le cose si complicano notevolmente se si passa ad un sistema generico di n-qubit. In tale situazione lo spazio di Hilbert associato ha dimensione 2^n, quindi le matrici unitarie che agiscono su tale spazio sono $2^n \times 2^n$, le quali formano il gruppo $U(2^n)$. Ognuna di queste matrici contiene in generale 2^{2n} parametri reali ! Il problema è quindi dato dal fatto che il numero di parametri cresce esponenzialmente con il numero di qubit: il numero di gate è estremamente grande e non vogliamo un QC in cui possiamo implementare qualsiasi trasformazione con 2^{2n} parametri reali. Una tale situazione è troppo difficile da realizzare, tuttavia preferiamo considerare un numero limitato di gate e costruire le operazioni desiderate tramite composizione. Si noti inoltre che un insieme continuo di operazioni é impossibile per la memoria limitata di un computer.
Il meglio che possiamo fare è introdurre una nozione di **universalità** e approssimare abbastanza bene una generica trasformazione unitaria utilizzando solamente un insieme finito di gates. Qual è il significato di tale approssimazione ? Dobbiamo definire un'opportuna nozione di **distanza** tra matrici:

Definizione 3.1 (**Distanza tra matrici**). *Date due matrici U e V, definiamo la seguente funzione* ***distanza***

$$E(U,V) = \max_{|\psi\rangle} \|(U-V)\,|\psi\rangle\|\,,$$

dove $|\psi\rangle$ è un vettore arbitrario.

E' possibile trovare un insieme discreto di matrici tale che tutte le possibili matrici unitarie possano essere realizzate a partire da tale insieme a meno di un errore ε arbitrario ? La risposta è sì: un possibile insieme di gate che soddisfa la precedente nozione di "approssimazione universale" è dato dai seguenti 4



Si vedano esplicitamente le matrici (1.5.1) e (1.5.2). Questi gate prendono il nome di `H-gate` H (Hadamard gate), `Phase-gate` S, $\pi/4$-`gate` T e il `CNOT-gate`. Si noti che il nome della matrice T deriva dal fatto che

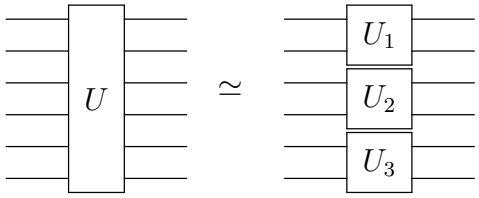
$$T = e^{i\frac{\pi}{8}} \begin{pmatrix} e^{-i\frac{\pi}{8}} & 0 \\ 0 & e^{i\frac{\pi}{8}} \end{pmatrix}.$$

Non lo dimostriamo esplicitamente, ma l'insieme di questi 4 gate è universale. E' importante sottolineare che H, S e T sono gate agenti sui singoli qubit, mentre il `CNOT-gate`

agisce sempre su almeno 2 qubit. Dato che avevamo visto che $T^2 = S$ potrebbe sorgere spontanea la domanda: perché è necessario considerare entrambi T e S ? Di solito si preferisce tenere anche S per la cosiddetta **Fault tolerance computation**, che approfondiremo quando parleremo di propagazione degli errori nei circuiti quantistici.
Alcune importanti proprietà che ci servirà sapere sui gate quantistici sono le seguenti:

1. *Tutti i gate agenti su n qubit possono essere approssimati come un prodotto di un opportuno numero di gate agenti su* 2 *qubit.*

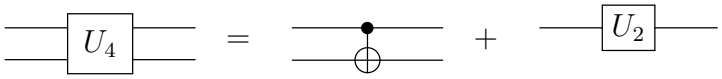
   Chiaramente il numero di gate agenti su 2 qubit deve essere sufficientemente grande per approssimare una matrice $2^n \times 2^n$ (matrice agente su n qubit). Per capire il significato di questa affermazione si pensi al circuito seguente di 6 qubit:



   In questo esempio il gate originale U è stato approssimato fattorizzandolo in 3 gate agenti ciascuno localmente solo su 2 qubit: il numero di operazioni per ricostruire la matrice $2^n \times 2^n$ del circuito a LHS è di ordine $\mathcal{O}(2^{2n})$. Chiaramente si tratta solamente di algebra: questo non è un modo molto efficiente di approssimare un gate agente su n qubit perché tipicamente si hanno comunque 2^{2n} fattori da tenere in considerazione.

2. *I gate agenti su 2 qubit possono essere scritti in termini di un* `CNOT-gate` *e di un gate agente su un singolo qubit.*

   Notiamo che, a differenza della proprietà precedente, questa proprietà è esatta e può essere svolta senza alcuna approssimazione. In termini di circuiti stiamo dicendo che



   dove $U_4 \in U(4)$ e $U_2 \in U(2)$. In generale è possibile dimostrare che per costruire una generica matrice di $U(4)$ si possono utilizzare due opportune matrici $A, B \in U(2)$ tali che $[A, B] \neq 0$.

3. *I gate agenti sui singoli qubit possono essere approssimati come prodotto di matrici H e T con un errore, il quale può essere arbitrariamente scelto più piccolo di* ε.

   Questo significa che data V la generica matrice unitaria 2×2 da approssimare (ricordare che contiene $2^2 = 4$ parametri reali) possiamo scrivere un'opportuna sequenza di prodotti tra H e T tali che

$$E(V, \ldots HHTH \ldots T \ldots H \ldots) < \varepsilon \,.$$

   Chiaramente più lunga è la sequenza più piccolo sarà l'errore entro il quale si può approssimare V. In realtà H e T non sono matrici speciali: questo argomento funziona con qualsiasi $A, B \in U(2)$ tali che $[A, B] \neq 0$. Matematicamente questa

proprietà è dovuta al fatto che il sottogruppo dato dai prodotti di H e T è **denso** in $U(2)$.

Ci si può chiedere se un'approssimazione mediante prodotti di H e T possa essere efficiente. Ancora una volta, fortunatamente la risposta è sì: esiste un teorema, chiamato **teorema di Solovay-Kitaev**, che stabilisce che il numero di prodotti tra H e T per approssimare una generica matrice di $U(2)$ è dell'ordine di $\mathcal{O}(\log_{10}^{c}(1/\varepsilon))$ dove $c \sim 2$.

Lezione 6 - 22/10/2021

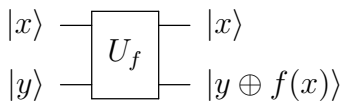
3.3 Quantum Parallelism

Definizione 3.2 (**Quantum Parallelism**). *Il **quantum parallelism** è una delle caratteristiche fondamentali di molti algoritmi quantistici. Consente ai computer quantistici di valutare una funzione $f(x)$ per molti valori diversi di x contemporaneamente.*

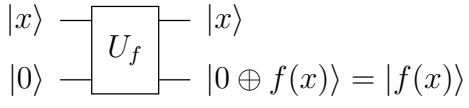
Supponiamo di considerare la più semplice funzione possibile $f(x) : \{0,1\}^{\otimes n} \to \{0,1\}$ definita su un dominio (insieme di numeri costruiti con n cifre di 0 e 1) e a elementi in un intervallo di bit. Assumiamo inoltre si saper calcolare efficientemente nel nostro computer tale funzione. Ciò che calcoliamo, dal punto di vista della computazione classica, lo possiamo valutare nella computazione quantistica, pertanto tutte le operazioni aritmetiche possono essere svolte dal calcolo quantistico. Un modo quindi di calcolare questa funzione su un computer quantistico è quello di considerare due differenti stati: immaginiamo un qubit $|y\rangle$ e uno stato che può essere un prodotto tensoriale di qubit, come ad esempio $|0\rangle^{\otimes n}$. Spesso considereremo lo stato $|0\rangle^{\otimes n}$ come stato iniziale in cui il computer quantistico viene preparato mediante una misurazione nella base computazionale perché è facilmente costruibile: ad esempio nel caso $n = 3$ se, a seguito di una misurazione, lo stato nel QC collassa in $|\psi\rangle \to |1\rangle \otimes |0\rangle \otimes |1\rangle$, basterà applicare un `X-gate` al primo e al terzo qubit per costruire lo stato voluto $|0\rangle^{\otimes 3}$.
Chiamiamo lo stato iniziale totale $|x, y\rangle$, dove x contiene l'informazione iniziale data in input e y conterrà, dopo delle opportune operazioni, il risultato cercato. Con un'appropriata sequenza di gate è possibile effettuare la trasformazione

$$|x, y\rangle \xrightarrow{U_f} |x, y \oplus f(x)\rangle \ , \tag{3.3.1}$$

dove U_f è un opportuno gate unitario che implementa l'operazione desiderata. Il circuito che implementa la (3.3.1) è



dove $|x\rangle$ prende il nome di **data register** e $|y\rangle$ prende il nome di **target register**. Questa rappresentazione è utile perché quando $|y\rangle = |0\rangle$ l'output del target register è esattamente l'oggetto che si vuole calcolare



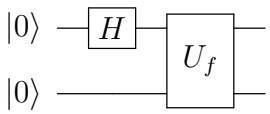
Notiamo che la (3.3.1) è invertibile: se applichiamo U_f due volte, otteniamo:

$$|x, y\rangle \to |x, y \oplus f(x)\rangle \to |x, y \oplus f(x) \oplus f(x)\rangle = |x, y\rangle \ ,$$

siccome $f(x) \oplus f(x) = 0$ indipendentemente dai valori di f. Fino ad ora avremmo potuto effettuare tutte queste operazioni in CC. L'importanza del QC risiede nel fatto che si possano considerare sovrapposizioni di stati appartenenti ad una base. Consideriamo il caso $n = 1$ (il data register è un qubit) e assumiamo il seguente stato iniziale

$$|x, y\rangle \equiv \underbrace{\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)}_{|x\rangle} \otimes \underbrace{|0\rangle}_{|y\rangle} = \frac{1}{\sqrt{2}}\left(|00\rangle + |10\rangle\right) \ ;$$

Se assumiamo che il computer sia preparato in $|0\rangle \otimes |0\rangle$ come possiamo rappresentare $|x, y\rangle$ in un circuito ? Possiamo sfruttare l'`H-gate` in questo modo:



infatti, utilizzando la (3.3.1), avremo

$$|0, 0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}\left(|00\rangle + |10\rangle\right) \xrightarrow{U_f} \frac{1}{\sqrt{2}}\left(|0, f(0)\rangle + |1, f(1)\rangle\right) \ .$$

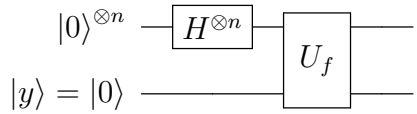
Questo circuito è particolarmente interessante perché l'output è una sovrapposizione di differenti stati contenenti informazioni riguardo la funzione: $f(0)$ e $f(1)$ appaiono simultaneamente nel medesimo stato. È come se avessimo valutato $f(x)$ per due valori di x contemporaneamente, parallelamente! A differenza del classic parallelism, in cui più circuiti vengono costruiti per calcolare $f(x)$ ed eseguiti simultaneamente, qui viene impiegato un singolo circuito per valutare la funzione $f(x)$ per più valori di x nello stesso momento: si sta sfruttando la capacità di un computer quantistico di essere in sovrapposizioni di stati diversi. Qui risiede il **quantum parallelism**.
Questo discorso può essere facilmente generalizzato al caso di n-qubit. Supponiamo che il data register si trovi in $|0\rangle^{\otimes n}$. Usiamo il fatto che l'azione dell'`H-gate` su n-qubit possa essere scritta nel seguente modo:

$$\begin{aligned} H^{\otimes n}|0\rangle^{\otimes n} &= \underbrace{H \otimes \cdots \otimes H}_{n\text{-volte}} \underbrace{|0\rangle \otimes \cdots \otimes |0\rangle}_{n\text{-volte}} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \cdots \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ &= \frac{1}{\sqrt{2^n}}(|000\ldots0\rangle + |010\ldots0\rangle + \ldots + |111\ldots1\rangle) = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \ , \end{aligned} \quad (3.3.2)$$

dove x rappresenta tutte le possibili stringhe di n-volte 0 e 1. Se il target si trova in $|y\rangle = |0\rangle$ e applichiamo ora U_f, il risultato è:

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

dove si è fatto uso della (3.3.1) con $|y\rangle = |0\rangle$. In termini di circuiti avremo



In un certo senso, il quantum parallelism consente di valutare simultaneamente tutti i possibili valori della funzione $f(x)$, anche se apparentemente abbiamo valutato $f(x)$ in una singola volta. Precisiamo che la misura dello stato nel caso del qubit singolo ci darà solamente $|0, f(0)\rangle$ oppure $|1, f(1)\rangle$. In maniera analoga per il caso generale, la misura dello stato $\sum_x |x, f(x)\rangle$ ci darà un solo $f(x_0)$ per un singolo valore casuale x_0. Ovviamente un computer classico può farlo più facilmente! La computazione quantistica richiede qualcosa di più del semplice quantum parallelism per essere utile; richiede cioè la capacità di estrarre informazioni su più di un valore di $f(x)$ da stati di sovrapposizione, come $\sum_x |x, f(x)\rangle$. Come vedremo nella prossima sezione, il "trucco" di considerare una sovrapposizione lineare ci permetterà di estrarre alcune informazioni su f in un modo più efficiente del CC.

3.4 Algoritmo di Deutsch

Una semplice modifica del circuito precedente dimostra come i circuiti quantistici possano essere più performanti rispetto a quelli classici. Nelle ultime righe del paragrafo precedente abbiamo detto che la computazione quantistica richiede qualcosa di più oltre al quantum parallelism per essere utilizzabile. L'**algoritmo di Deutsch** combina il meccanismo del **quantum parallelism** con la proprietà della meccanica quantistica dell'**interferenza**.
Si tratta di un algoritmo un po' accademico (le funzioni sono banali), tuttavia utile per illustrare l'idea di algoritmo quantistico. Lasciamo che entrambi input e output register contengano ciascuno un solo qubit, quindi stiamo esplorando le funzioni $f(x)$ che convertono un singolo bit in un singolo bit: $f(x) : \{0,1\} \to \{0,1\}$. Ci sono due modi piuttosto diversi di pensare a tali funzioni. Il primo modo è notare che ci sono solo quattro di queste funzioni, come mostrato nella Tabella 3.3.

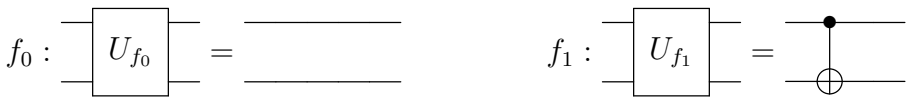
	$x=0$	$x=1$
f_0	0	0
f_1	0	1
f_2	1	0
f_3	1	1

Tabella 3.3: Esistono solo quattro funzioni distinte $f_j(x)$ che convertono un bit in un bit, tutte facilmente implementabili sia in un computer classico che quantistico.

Supponiamo che ci venga data una black-box (ossia un gate ignoto che indicheremo con `U-gate`) che calcola una di queste quattro funzioni eseguendo la seguente trasformazione unitaria:

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

In questo caso, se implementiamo in circuiti la Tabella 3.3 avremo:



Dato che la regola che vogliamo implementare è $|x,0\rangle \to |x,f(x)\rangle$ ($|y\rangle$ inizializzato a $|0\rangle$), in termini matematici questo significa scrivere:

$$
\begin{aligned}
f_0 &: \quad |x,0\rangle \longrightarrow |x,0\rangle \ , \\
f_1 &: \quad |x,0\rangle \xrightarrow{\texttt{CNOT}} \begin{cases} |0,0\rangle \ , & \text{per } x=0 \\ |1,1\rangle \ , & \text{per } x=1 \end{cases} , \\
f_2 &: \quad |x,0\rangle \xrightarrow{X} |x,1\rangle \xrightarrow{\texttt{CNOT}} \begin{cases} |0,1\rangle \ , & \text{per } x=0 \\ |1,0\rangle \ , & \text{per } x=1 \end{cases} , \\
f_3 &: \quad |x,0\rangle \xrightarrow{X} |x,1\rangle \ ,
\end{aligned}
$$

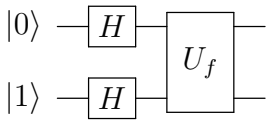
Supponiamo che ci venga data una black-box che esegua U_f per una delle quattro funzioni, ma non ci venga detto quale delle quattro operazioni. Ovviamente possiamo scoprirlo lasciando agire due volte la black-box, prima su $|0\rangle\otimes|0\rangle$ e poi su $|1\rangle\otimes|0\rangle$. Ma supponiamo di poter far agire la black-box solo una volta. Cosa possiamo conoscere di $f(x)$?
In un computer classico, dove siamo effettivamente limitati a lasciare che la black-box agisca sui qubit in uno dei quattro stati di base computazionale, possiamo conoscere il valore di:

- $f(0)$, lasciando che U_f agisca su uno dei due $|0\rangle \otimes |0\rangle$ o $|0\rangle \otimes |1\rangle$;
  - In tal caso possiamo limitare la scelta a f_0 o f_1 (se $f(0)=0$) oppure f_2 o f_3 (se $f(0)=1$).
- $f(1)$, lasciando che U_f agisca su $|1\rangle \otimes |0\rangle$ o $|1\rangle \otimes |1\rangle$;
  - In questa situazione abbiamo ristretto la funzione ad essere f_0 o f_2 (se $f(1)=0$) oppure f_1 o f_3 (se $f(1)=1$).

In definitiva, un computer classico necessita di due esecuzioni per determinare se f sia costante o meno. Sorprendentemente, risulta che con un computer quantistico questo non è necessario perché il problema può essere risolto con una singola esecuzione. Il punto interessante è che l'algoritmo non riguarda il calcolo preciso della funzione, ma piuttosto la comprensione di una o più sue proprietà: quando l'algoritmo viene lanciato non impariamo nulla sui valori individuali di $f(0)$ e $f(1)$, ma siamo comunque in grado di rispondere alla domanda sui loro valori relativi. Chiaramente otteniamo meno informazioni di quelle che otterremmo rispondendo alla domanda con un computer classico, ma, rinunciando alla possibilità di acquisire quella parte dell'informazione che è irrilevante per la domanda a cui vogliamo rispondere, possiamo ottenere la risposta con una sola applicazione di U_f. Come sottolineato in precedenza l'algoritmo combina il quantum parallelism e l'interferenza: possiamo preparare il computer nello stato $|0\rangle\otimes|1\rangle$ della base canonica e applicare l'`H-gate` a entrambi i qubit:

$$
(H\otimes H)\,|0\rangle\otimes|1\rangle = \underbrace{\frac{|0\rangle+|1\rangle}{\sqrt{2}}}_{\substack{\text{quantum}\\ \text{parallelism}}} \otimes \underbrace{\frac{|0\rangle-|1\rangle}{\sqrt{2}}}_{\text{interferenza}} ; \tag{3.4.1}
$$

in un circuito significa scrivere



Chiamando per semplicità $|x\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ e applicando U_f alla (3.4.1) tramite (3.3.1), possiamo esplicitamente vedere che cosa implica il termine di interferenza:

$$|x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \xrightarrow{U_f} \frac{1}{\sqrt{2}}\left(|x, 0 \oplus f(x)\rangle - |x, 1 \oplus f(x)\rangle\right)$$
$$= \begin{cases} \frac{1}{\sqrt{2}}\left(|x, 0 \oplus 0\rangle - |x, 1 \oplus 0\rangle\right) = |x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle), & \text{per } f(x) = 0 \\ \frac{1}{\sqrt{2}}\left(|x, 0 \oplus 1\rangle - |x, 1 \oplus 1\rangle\right) = -|x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle), & \text{per } f(x) = 1 \end{cases}.$$

Combinando i due casi in un'unica espressione compatta abbiamo ottenuto

$$|x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \xrightarrow{U_f} (-1)^{f(x)}|x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle), \tag{3.4.2}$$

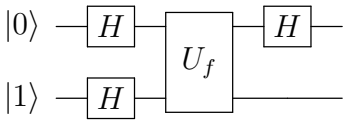
Sostituendo $|x\rangle$ con lo stato iniziale che implementava il quantum parallelism avremo

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{U_f} \frac{1}{\sqrt{2}}\left[(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle\right] \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}};$$

dato che il segno relativo nella parentesi quadra dipende dal fatto che $f(0)$ e $f(1)$ siano uguali o meno, possiamo riscrivere quest'ultima espressione come

$$\begin{cases} (-1)^{f(0)}\frac{|0\rangle+|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle-|1\rangle}{\sqrt{2}}, & \text{per } f(0) = f(1) \\ (-1)^{f(0)}\frac{|0\rangle-|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle-|1\rangle}{\sqrt{2}}, & \text{per } f(0) \neq f(1) \end{cases}.$$

Come ultimo passaggio si applica l'`H-gate` al primo qubit in maniera tale che il circuito totale diventi:



Questa modifica trasforma il risultato precedente in

$$\begin{cases} (-1)^{f(0)}\frac{|0\rangle+|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle-|1\rangle}{\sqrt{2}} \xrightarrow{H} (-1)^{f(0)}|0\rangle \otimes \frac{|0\rangle-|1\rangle}{\sqrt{2}}, & \text{per } f(0) = f(1) \\ (-1)^{f(0)}\frac{|0\rangle-|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle-|1\rangle}{\sqrt{2}} \xrightarrow{H} (-1)^{f(0)}|1\rangle \otimes \frac{|0\rangle-|1\rangle}{\sqrt{2}}, & \text{per } f(0) \neq f(1) \end{cases}.$$

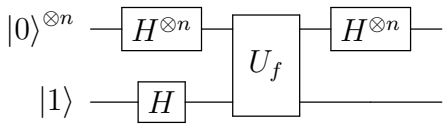
Il risultato finale ci suggerisce che possiamo effettuare solamente una misurazione sul primo qubit: ottenendo $|0\rangle$ o $|1\rangle$ siamo in grado, con una singola misura, di stabilire se $f(0) = f(1)$ oppure $f(0) \neq f(1)$. Questo significa che siamo in grado di escludere 2 delle 4 funzioni con una singola esecuzione dell'algoritmo.
Questo esempio permette di evidenziare quale sia la differenza tra il quantum parallelism e gli algoritmi randomizzati classici. Ingenuamente, si potrebbe pensare che lo stato finale corrisponda piuttosto a un calcolatore classico probabilistico che valuta $f(0)$ con probabilità $\frac{1}{2}$, o $f(1)$ con probabilità $\frac{1}{2}$. La differenza è che in un computer classico queste due alternative si escludono sempre mentre in un computer quantistico è possibile che le due alternative interferiscano l'una con l'altra per ottenere alcune proprietà globali della funzione $f(x)$. Utilizzando un opportuno gate (nel nostro caso l'`H-gate`) siamo in grado di ricombinare le diverse alternative.

3.5 Algoritmo di Deutsch-Jozsa

L'algoritmo di Deutsch è un semplice caso di un algoritmo quantistico più generale, noto come **algoritmo di Deutsch-Jozsa**, che evidenzia esplicitamente come il QC offra un grosso miglioramento rispetto ai metodi del CC. Supponiamo di avere una black-box che calcola una funzione booleana $f(x) : \{0,1\}^{\otimes n} \to \{0,1\}$ e supponiamo di sapere per certo che $f(x)$ sia solamente una delle seguenti alternative:

- **Funzione costante** (*constant*): l'output è sempre 0 oppure 1 indipendentemente dall'input.
- **Funnzione bilanciata** (*balanced*): l'output è costituito per metà dal valore 0 e metà dal valore 1.

Lo scopo dell'algoritmo è quello di capire quale delle due sia l'alternativa corretta con il minor numero di esecuzioni. Classicamente potremmo risolvere questo problema calcolando $2^{n-1}+1$ valori della funzione perché è necessario calcolare almeno una metà dei valori più un valore aggiuntivo. Chiaramente si tratta di un numero esponenzialmente grande. Quello che fa l'algoritmo di Deutsch-Jozsa è risolvere il problema perfettamente con una sola query quantistica. Cominciamo scrivendo il circuito che descrive tale algoritmo, il quale è molto simile a quello di Deutsch con la sola differenza che il data register non è un singolo qubit, ma piuttosto un prodotto tensoriale di n-qubit:



Vediamo nello specifico cosa succede all'interno del circuito:

1. Viene inizializzato (preparato) lo stato in $|0\rangle^{\otimes n} \otimes |1\rangle$;
2. Creiamo una sovrapposizione di stati usando l'`H-gate` su tutti gli $n+1$ qubit:

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

   dove si è fatto uso della (3.3.2). Notiamo che ora nell'output register è presente lo stato che nella sezione precedente avevamo visto essere associato all'interferenza.

3. Valutiamo la funzione $f(x)$ usando la block-box di U_f

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

   dove, essendo $|x\rangle$ arbitrario, abbiamo fatto uso della (3.4.2).

4. Applichiamo nuovamente l'`H-gate` ai primi n qubit. Per capire il risultato di $H^{\otimes n}|x\rangle$ consideriamo per semplicità il caso $n=1$: formalmente avremo

$$H|x\rangle = \sum_{z=0}^{1} \frac{(-1)^{xz}}{\sqrt{2}} |z\rangle \ , \quad \text{dove } x = 0 \text{ oppure } 1 .$$

Per n generico possiamo generalizzare scrivendo

$$\begin{aligned} H^{\otimes n}\left|x\right\rangle &= (H \otimes \ldots \otimes H)\left|x_0\right\rangle \otimes \left|x_1\right\rangle \otimes \ldots \otimes \left|x_{n-1}\right\rangle \\ &= \sum_{z_0=0}^{1} \cdots \sum_{z_{n-1}=0}^{1} \frac{(-1)^{x_0 z_0}(-1)^{x_1 z_1} \cdots (-1)^{x_{n-1} z_{n-1}}}{\sqrt{2^n}} \left|z\right\rangle \,, \end{aligned}$$

dove $\left|z\right\rangle \equiv \left|z_0, z_1, \ldots, z_{n-1}\right\rangle$. In maniera più compatta possiamo scrivere quindi l'azione dell'`H-gate` sugli n qubit (nonché risultato finale del circuito) come

$$\sum_{z=0}^{2^n-1} \sum_{x=0}^{2^n-1} \frac{(-1)^{f(x)+x\cdot z}}{2^n} \left|z\right\rangle \otimes \frac{\left|0\right\rangle - \left|1\right\rangle}{\sqrt{2}} \,, \tag{3.5.1}$$

dove abbiamo indicato con $x \cdot z$ il **prodotto bit a bit modulo 2**:

$$x \cdot z = (x_0 z_0 + \ldots + x_{n-1} z_{n-1}) \mod 2 \,.$$

5. Infine misuriamo per ottenere lo stato finale $\left|z\right\rangle$.

Ricordiamo che il problema è quello di determinare se f sia constant o balanced. Notiamo dal risultato (3.5.1) che il data register ora contiene una sovrapposizione lineare di tutti i possibili stati che si scrivono come stringhe contenenti n volte 0 e 1. In $\left|z\right\rangle$ è presente un caso particolare: consideriamo la situazione in cui $\left|z\right\rangle = \left|00\ldots0\right\rangle = \left|0\right\rangle^{\otimes n}$ e cerchiamo la probabilità di ottenere tale stato guardando il modulo quadro del coefficiente:

$$P\left(\left|z\right\rangle = \left|0\right\rangle^{\otimes n}\right) = \left|\sum_{x=0}^{2^n-1} \frac{(-1)^{f(x)}}{2^n}\right|^2 = \begin{cases} 1\,, & \text{se } f(x) \text{ è constant} \\ 0\,, & \text{se } f(x) \text{ è balanced} \end{cases} .$$

Notiamo che quando la probabilità è 1 a numeratore si hanno 2^n termini tutti uguali ($(-1)^1$ oppure $(-1)^0$) che si semplificano con il fattore $1/2^n$; quando invece la probabilità è nulla a numeratore si ha uno stesso numero di $(-1)^1$ e $(-1)^0$ che si cancellano esattamente. Come abbiamo detto $\left|z\right\rangle = \left|0\right\rangle^{\otimes n}$ è un caso particolare molto importante perché permette di risolvere il problema mediante la misura dello stato. Se misurando z otteniamo $\left|0\right\rangle^{\otimes n}$ allora, con probabilità 1 (quindi sempre), lo stato è $\left|0\right\rangle^{\otimes n}$ e la funzione è constant; al contrario quando la misura di z produce un qualsiasi stato differente da $\left|0\right\rangle^{\otimes n}$ allora, essendo $P\left(\left|z\right\rangle = \left|0\right\rangle^{\otimes n}\right) = 0$, lo stato $\left|0\right\rangle^{\otimes n}$ non è nemmeno presente in z e possiamo stabilire con assoluta certezza che la funzione è balanced. Il fatto importante è che essendo queste misure mutualmente esclusive, possiamo determinare se f sia constant o balanced con una singola misurazione. Quindi si tratta di effettuare una sola misurazione in QC contro $\mathcal{O}(2^n)$ misure in CC.
Osserviamo che il confronto tra algoritmi classici e quantistici è in qualche modo un confronto delicato, poiché il metodo per valutare la funzione è abbastanza diverso nei due casi. Se fosse consentito utilizzare un computer probabilistico classico, per valutare $f(x)$ per pochi x scelti a caso, si può determinare molto rapidamente con alta probabilità se $f(x)$ è *constant* o *balanced*. Questo scenario probabilistico è forse più realistico dello scenario deterministico che abbiamo considerato.
Ribadiamo nuovamente che questo algoritmo è un esempio molto accademico in quanto non esistono problemi fisici o matematici reali che necessitano di sapere se una funzione sia constant o balanced. Nonostante ciò il fatto importante è che grazie a questo algoritmo

quantistico non è più necessario aspettare un tempo esponenzialmente[v] crescente nel numero di bit per sapere il risultato.

3.6 Algoritmo di Bernstein-Vazirani

Consideriamo un altro algoritmo di black-box per il quale gli algoritmi quantistici forniscono un vantaggio: l'**algoritmo di Bernstein-Vazirani**. Qui, a differenza dei due casi precedenti, abbiamo accesso alla funzione della black-box $f : \{0,1\}^n \to \{0,1\}$. Supponiamo che la funzioni sia data da[vi]:

$$f(x) = a \cdot x = (a_0 x_0 + \ldots + a_{n-1} x_{n-1}) \mod 2\,, \quad \text{dove } a \geqslant 0 \text{ e } x < 2^n\,.$$

Sappiamo che la funzione è lineare, tuttavia l'obiettivo di questo algoritmo è trovare il valore di a. Classicamente, questo problema potrebbe richiedere n query poiché ogni query può fornire solo un nuovo bit di informazioni su a, ma a possiede n bit: dobbiamo valutare $f(1000\ldots) = a_0$, $f(0100\ldots) = a_1$ e così via con n valutazioni fino a $f(111\ldots1) = a_{n-1}$. L'algoritmo di Bernstein-Vazirani, invece, risolve il problema quantisticamente utilizzando una sola query!
Consideriamo il medesimo circuito dell'algoritmo di Deutsch-Josza e il suo output (3.5.1): nel caso in cui $f(x) = a \cdot x$ esso diventa

$$\sum_{z=0}^{2^n-1} \sum_{x=0}^{2^n-1} \frac{(-1)^{x\cdot(a+z)}}{2^n} |z\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}\,.$$

Come nell'algoritmo precedente guardiamo il coefficiente di $|z\rangle$:

$$\frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{x\cdot(a+z)} = (-1)^{x_0(a_0+z_0)+\ldots+x_{n-1}(a_{n-1}+z_{n-1})} = \frac{1}{2^n} \prod_{j=0}^{n-1} \left(\sum_{x_j=0}^{1} (-1)^{x_j(a_j+z_j)} \right)\,,$$

ma ogni termine nella parentesi tonda è la somma di termini che possono essere ± 1 a seconda dell'esponente. Distinguiamo i due casi:

- Se $(a_j + z_j = 0) \mod 2$ allora il coefficiente è

$$\frac{1}{2^n} \prod_{j=0}^{n-1} (2) = 1\,, \quad \Rightarrow \quad \text{Probabilità } 1\,.$$

- Al contrario quando $(a_j + z_j = 1) \mod 2$ allora il coefficiente diventa

$$\frac{1}{2^n} \prod_{j=0}^{n-1} \left[(-1)^{0\cdot1} + (-1)^{1\cdot1}\right] = 0\,, \quad \Rightarrow \quad \text{Probabilità } 0\,.$$

[v] Talvolta non si vuole sapere con precisione assoluta se f sia constant o balanced, ma è sufficiente stabilirlo entro un errore dato ε. Un ipotetico algoritmo classico e probabilistico di questo tipo diventa di ordine polinomiale in n: passare da $\mathcal{O}$(polinomio in n) a $\mathcal{O}(1)$ mediante la controparte quantistica non è più un miglioramento così estremo come passare da $\mathcal{O}(2^n)$ ad $\mathcal{O}(1)$!

[vi] Come prima il simbolo "$\cdot$" indica il prodotto bit a bit modulo 2

Ancora una volta, i due casi della probabilità sono mutualmente esclusivi e quindi avremo

$$\begin{aligned}(a_j + z_j = 0) \mod 2\,, \quad &\Rightarrow \quad a = z\,, \quad \Rightarrow \quad \text{Probabilità } 1\,,\\ (a_j + z_j = 1) \mod 2\,, \quad &\Rightarrow \quad a \neq z\,, \quad \Rightarrow \quad \text{Probabilità } 0\,.\end{aligned}$$

Questo significa che il nostro stato, in realtà, non è una sovrapposizione lineare, ma contiene bensì solamente lo stato

$$|a\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}\,;$$

e quindi attraverso un'unica operazione di misura sui primi n-qubit, otteniamo a, la nostra incognita.

Lezione 7 - 25/10/2021

I prossimi due algoritmi sono tra quelli più conosciuti, in termini di algoritmi quantistici, sia dal punto di vista storico del QC sia dal punto di vista applicativo:

- L'algoritmo di ricerca del periodo di una funzione: l'**algoritmo di Shor**;
- L'algoritmo di ricerca di particolari elementi in un database: l'**algoritmo di Grover**.

Prima di addentrarci nello studio del più difficile (non vedremo tutto il discorso legato alla teoria dei numeri) dei due, l'algoritmo di Shor, introduciamo il seguente concetto:

3.7 Quantum Fourier Transform

Il cuore dell'algoritmo di Shor è la **QFT** o **Quantum Fourier Transform**, che può essere eseguita da un circuito quantistico. La QFT di n qubit è definita come quella trasformazione unitaria $\hat{U}_{\text{FT}}$ la cui azione su un elemento $|x\rangle \in \mathcal{H}$ è data da:

$$\hat{U}_{\text{FT}}\,|x\rangle = \frac{1}{2^{\frac{n}{2}}} \sum_{y=0}^{2^n-1} e^{2\pi i \frac{x\cdot y}{2^n}}\,|y\rangle \tag{3.7.1}$$

dove con la notazione precedente intendiamo $|x\rangle \equiv |x_{n-1}\rangle \otimes |x_{n-2}\rangle \otimes \ldots \otimes |x_0\rangle$, in cui ciascun qubit $|x_i\rangle$ può essere un elemento della base computazionale, quindi $|0\rangle$ o $|1\rangle$. Notiamo inoltre che il prodotto $x \cdot y$ ad esponente è un prodotto scalare tra interi e non un prodotto bit a bit modulo 2. Lo stato $|x\rangle$ può essere scritto utilizzando anche la codifica digitale degli interi, ossia

$$x = 2^{n-1}x_{n-1} + 2^{n-2}x_{n-2} + \ldots + 2^0 x_0\,, \quad \text{dove } 0 \leqslant x \leqslant 2^n - 1\,.$$

Notiamo che il fattore davanti alla sommatoria in (3.7.1) è un fattore di normalizzazione perchè abbiamo diviso per la radice del numero totale degli stati: lo spazio di Hilbert di $|x\rangle$ ha infatti $\dim \mathcal{H} = 2^n$ poiché è frutto del prodotto tensoriale degli n spazi associati ai

singoli qubit. Dato che $\hat{U}_{\text{FT}}$ è un operatore che agisce su $\mathcal{H}$, possiamo applicare la (3.7.1) ad una sovrapposizione di stati $|x\rangle$ con ampiezze complesse $\gamma(x)$:

$$\hat{U}_{\text{FT}} \left(\sum_{x=0}^{2^n-1} \gamma(x) |x\rangle \right) = \sum_{x,y=0}^{2^n-1} \frac{\gamma(x)}{2^{\frac{n}{2}}} e^{2\pi i \frac{x\cdot y}{2^n}} |y\rangle = \sum_{y=0}^{2^n-1} \hat{\gamma}(y) |y\rangle \ , \tag{3.7.2}$$

dove abbiamo ottenuto un'altra sovrapposizione con ampiezze che sono legate a $\gamma(x)$ dalla **DFT** o **Discrete Fourier Transform**:

$$\hat{\gamma}(y) = \sum_{x=0}^{2^n-1} \frac{e^{2\pi i \frac{x\cdot y}{2^n}}}{2^{\frac{n}{2}}} \gamma(x) \, . \tag{3.7.3}$$

Si noti che la (3.7.1) agisce sui coefficienti $\gamma(x)$ come in (3.7.3), ossia tramite una versione discretizzata della trasformata di Fourier standard. In generale la DFT è largamente utilizzata nella teoria dei segnali.
Per calcolare ciascun coefficiente $\hat{\gamma}(x)$ in (3.7.3) si richiedono $2^n \times 2^n = 2^{2n}$ operazioni (dimensione della matrice), le quali sono un enormità ! In CC esiste un celebre algoritmo chiamato **FFT** o **Fast Fourier Transform** che migliora il numero precedente fino a $\mathcal{O}(n2^n)$, ottenendo quindi un modo molto più efficiente per calcolare $\hat{\gamma}(x)$. In realtà esiste un algoritmo quantistico per eseguire la trasformazione unitaria $\hat{U}_{\text{FT}}$ in un tempo esponenzialmente più veloce, perché cresce solo come $\mathcal{O}(n^2)$. Il problema, come al solito, è che non si può conoscere l'insieme completo dei coefficienti di Fourier, come si fa dopo aver applicato la FFT: il risultato è infatti una sovrapposizione $\sum_y \hat{\gamma}(y) |y\rangle$ sulla quale è necessario effettuare una misurazione che permetterà di ottenere solamente 1 coefficiente. Nonostante quindi l'algoritmo per il calcolo della QFT non migliori l'algoritmo classico della FFT, la (3.7.1) si è rivelata molto utile per la risoluzione di problemi del mondo quantistico. Ad esempio, se γ è una funzione periodica con un periodo r non maggiore di $2^{\frac{n}{2}}$, allora un registro nello stato (3.7.2) può fornire potenti indizi sul valore preciso del periodo, anche se r può essere lungo centinaia di cifre. Per il momento il nostro scopo è mostrare che è possibile costruire un circuito che calcoli in un numero di step di ordine $\mathcal{O}(n^2)$ la QFT.
Consideriamo, come al solito, come punto di partenza lo stato $|0\rangle^{\otimes n}$. Sappiamo dalla (3.3.2) che se applichiamo l'`H-gate` su tale stato avremo

$$H^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{2^{\frac{n}{2}}} \sum_{y=0}^{2^n-1} |y\rangle \ ,$$

ossia una somma su tutti gli stati nella base computazionale. Definiamo ora un operatore $\mathcal{Z}$ che agisce nel modo seguente:

$$\mathcal{Z} |y\rangle = e^{2\pi i \frac{y}{2^n}} |y\rangle \ ;$$

in questo modo l'operatore $\hat{U}_{\text{FT}}$ della (3.7.1) può essere riscritto come

$$\hat{U}_{\text{FT}} |x\rangle = \mathcal{Z}^x H^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{2^{\frac{n}{2}}} \sum_{y=0}^{2^n-1} \mathcal{Z}^x |y\rangle \ , \quad \text{con } \mathcal{Z} = e^{2\pi i \frac{xy}{2^n}} \ ; \tag{3.7.4}$$

si ricordi sempre che x è un intero. Cerchiamo di capire cosa sia $\mathcal{Z}$. Consideriamo il caso del qubit singolo ($n = 1$):

$$\mathcal{Z} |y\rangle = e^{\pi i y} |y\rangle = \begin{cases} |y\rangle \ , & \text{per } y = 0 \\ -|y\rangle \ , & \text{per } y = 1 \end{cases} \ , \quad \Rightarrow \quad \mathcal{Z} = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} .$$

Si noti che un altro modo conveniente di scriverlo è come esponenziale

$$Z = e^{i\pi n}, \quad \text{dove } n = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

Per passare alla generalizzazione per n qubit ricordiamo che, come lo stato $|x\rangle$ di (3.7.1), possiamo scrivere nella base computazionale che $|y\rangle = |y_{n-1}\rangle \otimes \ldots \otimes |y_0\rangle$ e analogamente come intero avremo $y = 2^{n-1}y_{n-1} + \ldots + 2^0 y_0$. Introduciamo n differenti matrici, che chiameremo n_i con $i = 0, \ldots, n-1$, che agiscono sul corrispondente qubit di $|y\rangle$ dando 0 o 1 a seconda del valore del qubit: questo significa scrivere che

$$\left(2^{n-1}n_{n-1} + 2^{n-2}n_{n-2} + \ldots + n_0\right)|y\rangle = 2^{n-1}y_{n-1}\,|y_{n-1}\rangle \otimes \ldots \otimes 2^0 y_0\,|y_0\rangle = y\,|y\rangle\,,$$

quindi si tratta di un particolare modo di calcolare la codifica digitale, ossia l'intero y, dello stato $|y\rangle$. Notiamo che la matrice n_{n-1} agisce su $|y_{n-1}\rangle$, n_{n-2} agisce su $|y_{n-2}\rangle$ e così via fino a n_0 che agisce su $|y_0\rangle$, questo perché in generale $n_p\,|y_p\rangle = y_p\,|y_p\rangle$. Utilizzando quindi questa notazione possiamo riscrivere l'operatore $\mathcal{Z}$ in questo modo:

$$\mathcal{Z}\,|y\rangle = e^{\frac{2\pi i}{2^n}y}\,|y\rangle = e^{\frac{2\pi i}{2^n}(2^{n-1}n_{n-1}+\cdots+n_0)}\,|y\rangle\,,$$

dove si è utilizzata la formula $n_p\,|y_p\rangle = y_p\,|y_p\rangle$ ad esponente e si è riconosciuta la codifica digitale di y. Per calcolare la QFT come in (3.7.4) ci serve saper calcolare $\mathcal{Z}^x$. Al posto che farlo in generale, focalizziamoci su un esempio perché vedremo alla fine che otterremo un circuito il cui schema è facilmente generalizzabile per il calcolo della QFT per un numero generico di qubit.

Esempio 3.4 (**QFT per 3 qubit**). *Vogliamo valutare $\mathcal{Z}^x$. Usando l'espressione di $\mathcal{Z}$ in* (3.7.4) *e ricordando la codifica digitale di x e y per $n = 3$ possiamo facilmente scrivere*

$$\mathcal{Z}^x = e^{\frac{2\pi i}{8}(4x_2+2x_1+x_0)(4n_2+2n_1+n_0)}.$$

Semplifichiamo questa espressione ricordando che $e^{2\pi i n} = \mathbb{I}$, dato che $n = 0, 1$, e molti termini nel prodotto delle tonde ad esponente sono in realtà multipli interi di $2\pi i n$. Più in dettaglio possiamo scrivere la precedente come

$$\mathcal{Z}^x = e^{\pi i\left[n_2x_0 + n_1\left(x_1+\frac{x_0}{2}\right)+n_0\left(x_2+\frac{x_1}{2}+\frac{x_0}{4}\right)\right]}.$$

Scriviamo quindi la (3.7.4)*:*

$$\mathcal{Z}^x H^{\otimes 3}\,|0\rangle^{\otimes 3} = e^{i\pi n_2 x_0} H_2\,|0\rangle_2 \otimes e^{i\pi n_1\left(x_1+\frac{x_0}{2}\right)} H_1\,|0\rangle_1 \otimes e^{i\pi n_2\left(x_2+\frac{x_1}{2}+\frac{x_0}{4}\right)} H_0\,|0\rangle_0\,, \quad (3.7.5)$$

dove il label su ogni `H-gate` *indica su quale qubit quell'operatore sta agendo. Per calcolare l'azione di ciascun operatore sul rispettivo qubit utilizziamo il seguente stratagemma: le matrici H_i non commutano con gli esponenziali alla loro sinistra, tuttavia possiamo scrivere che*

$$e^{i\pi x n} H\,|0\rangle = H\,|x\rangle\,, \quad \textit{dove } x = 0, 1\,;$$

infatti, ricordando le (1.2.2)*, avremo*

$$\begin{cases} H\,|0\rangle = H\,|0\rangle\,, & x = 0 \\ e^{i\pi n} H\,|0\rangle = ZH\,|0\rangle = Z\,|+\rangle = |-\rangle = H\,|1\rangle\,, & x = 1 \end{cases}.$$

Usando questo risultato, la (3.7.5) *può essere riscritta nel seguente modo*

$$\begin{aligned}\mathcal{Z}^x H^{\otimes 3} |0\rangle^{\otimes 3} &= H_2 |x_0\rangle_2 \otimes e^{i\pi n_1 \frac{x_0}{2}} H_1 |x_1\rangle_1 \otimes e^{i\pi n_0 \left(\frac{x_1}{2}+\frac{x_0}{4}\right)} H_0 |x_2\rangle_0 \\ &= H_2 e^{i\pi n_1 \frac{x_0}{2}} H_1 e^{i\pi n_0 \frac{x_1}{2}} e^{i\pi n_0 \frac{x_0}{4}} H_0 |x_0\rangle_2 \otimes |x_1\rangle_1 \otimes |x_2\rangle_0 \ ,\end{aligned}$$

dove nell'ultimo passaggio abbiamo raggruppato tutti gli operatori a sinistra e diviso gli esponenziali contenenti n_0 dato che commutano tra loro. Notiamo che durante questo conto abbiamo ottenuto una permutazione dei qubit iniziali ($x_0 \leftrightarrow x_2$). Lo stato $|x_0\rangle_2 \otimes |x_1\rangle_1 \otimes |x_2\rangle_0$ è un autostato degli operatori numerici n_2, n_1, n_0 con i rispettivi autovalori x_0, x_1, x_2. Consideriamo il primo qubit $H_2 e^{i\pi n_1 \frac{x_0}{2}} |x_0\rangle_2$: sappiamo che $n_2 |x_0\rangle_2 = x_0 |x_0\rangle_2$ quindi possiamo tranquillamente rimpiazzare $n_2 \leftrightarrow x_0$ ad esponente. Chiaramente lo possiamo fare perché solamente la matrice di Hadamard H_2 agisce su $|x_0\rangle_2$, ed essa si trova a sinistra dell'esponenziale (in generale le matrici di Hadamard non commutano con questi esponenziali, tuttavia in questa situazione si trovano tutte a sinistra). Un discorso analogo vale anche per gli altri due qubit. Riassumendo: possiamo sostituire ad esponente ogni x_i con l'operatore numerico n_{2-i}:

$$\mathcal{Z}^x H^{\otimes 3} |0\rangle^{\otimes 3} = H_2 e^{i\pi \frac{n_1 n_2}{2}} H_1 e^{i\pi \frac{n_0 n_1}{2}} e^{i\pi \frac{n_0 n_2}{4}} H_0 |x_0\rangle_2 \otimes |x_1\rangle_1 \otimes |x_2\rangle_0 \ ;$$

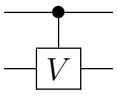
infine, se definiamo l'operatore unitario P che realizza la permutazione degli stati della base computazionale, ossia $P|x\rangle = P(|x_2\rangle \otimes |x_1\rangle \otimes |x_0\rangle) = |x_0\rangle \otimes |x_1\rangle \otimes |x_2\rangle$, possiamo scrivere

$$U_{FT}|x\rangle = \mathcal{Z}^x H^{\otimes 3} |0\rangle^{\otimes 3} = H_2 e^{i\pi \frac{n_1 n_2}{2}} H_1 e^{i\pi \frac{n_0 n_1}{2}} e^{i\pi \frac{n_0 n_2}{4}} H_0 P |x\rangle \ .$$

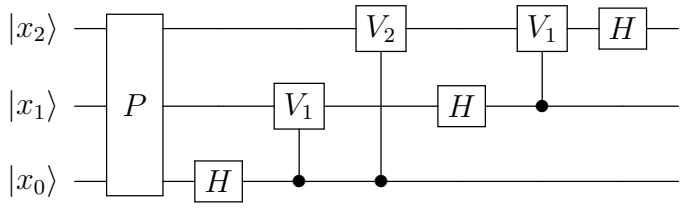
Per capire che tipologia di operatore sia U_{FT} ricordiamo che sappiamo bene come agiscono gli `H-gate`*, inoltre non è difficile costruire un opportuno* `P-gate` *che inverta l'ordine dei qubit. Gli esponenziali, invece, sono operatori che contengono delle paia di matrici n_i agenti sui singoli qubit: tutti questi sono della forma*

$$V_{ij} = e^{i\pi \frac{n_i n_j}{2^{|i-j|}}} \ ,$$

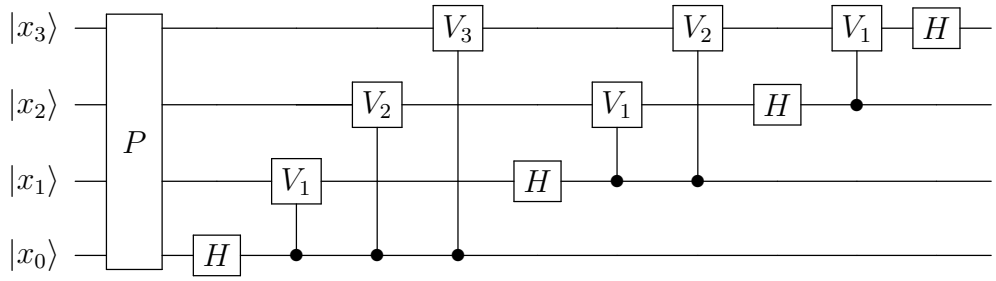
dove $|i-j|$ è la distanza tra i qubit i e j nell'array contenente tutti i qubit. Qual è l'effetto esplicito di ciascun V_{ij} sui qubit i e j ? Quando i è nello stato $|0\rangle$ allora $n_i = 0$ e l'esponenziale non fa nulla; ma quando i è nello stato $|1\rangle$ allora $n_i = 1$ e l'esponenziale agisce come $e^{i\pi \frac{n_j}{2^{|1-j|}}}$. Quindi si tratta di una sorta di `Controlled-V-gate` *che agisce solamente quando il primo qubit è $|1\rangle$:*



Si noti che il `CNOT-gate` *è un caso particolare del* `Controlled-V-gate` *quando $V = X$. In termini di circuiti, ponendo $V_k = e^{i\pi \frac{n}{2^k}}$, l'azione dell'operatore che calcola la QFT per 3 qubit può essere rappresentata come:*



Cosa succede nel caso in cui $n = 4$? La struttura del circuito dell'esempio precedente può essere facilmente generalizzata, infatti:



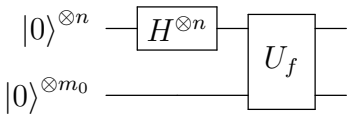
Qual è il numero totale di gate necessari ? Dal circuito precedente ($n = 4$) si hanno $4+3+2+1 = 10 \sim \mathcal{O}(4^2)$ gate, quindi in generale avremo $n+(n-1)+(n-2)+\ldots \sim \mathcal{O}(n^2)$, dove il massimo è proprio n^2. Dunque l'algoritmo quantistico per il calcolo della QFT è di ordine $\mathcal{O}(n^2)$ nel numero di qubit n.

3.8 Algoritmo di Shor: period finding

La ricerca del periodo di una funzione è importante per diverse ragioni: vedremo in che modo possiamo usare questo risultato per rompere la crittografia RSA standard, tuttavia è importante anche per simulazioni quantistiche, come ad esempio quando si vogliono trovare gli autovalori di matrici unitarie molto grandi.
Supponiamo di avere una funzione di interi e periodica di periodo r:

$$f : \mathbb{Z} \to \{0,1\}^{\otimes m_0} ,$$

dove sappiamo per certo che $\exists\, r \in \mathbb{Z} : f(x+r) = f(x)$ dove $r \leqslant N \equiv 2^{n_0}$. Quindi si tratta di trovare il periodo di una funzione periodica data in input: chiaramente se la funzione fosse definitiva sui reali il problema sarebbe banale perché richiederebbe un semplice disegno di un plot. Il miglior algoritmo classico ("general number field sieve") richiede un numero di operazioni di ordine $\mathcal{O}\Big(e^{n_0^{1/3}\log^{2/3} n_0}\Big)$, quindi presenta un comportamento esponenziale in n_0. L'algoritmo di Shor, invece, richiede solamente un numero di operazioni di ordine $\mathcal{O}\big(n_0^2 \log^2 n_0\big)$, un bel vantaggio rispetto al caso classico perché presenta un comportamento polinomiale in n_0.
L'algoritmo funziona come segue. Innanzitutto consideriamo un data register costituito da n qubit preparati in $|0\rangle^{\otimes n}$ e un output register (che conterrà il risultato della funzione f) fatto di m_0 qubit preparati in $|0\rangle^{\otimes m_0}$. Il circuito che vogliamo applicare è il seguente:



I qubit nel data register sono tipicamente di più di quanti ne necessitiamo per valutare il periodo (n_0), infatti di solito $n \sim 2n_0$ in maniera tale che $2^n \sim N^2$. Come al solito, l'`H-gate` e U_f agiranno nel seguente modo:

$$U_f\left[\left(H^{\otimes n}|0\rangle^{\otimes n}\right)\otimes|0\rangle^{\otimes m_0}\right] = U_f\left(\sum_{x=0}^{2^n-1}\frac{1}{2^{\frac{n}{2}}}|x\rangle\otimes|0\rangle^{\otimes m_0}\right) = \frac{1}{2^{\frac{n}{2}}}\sum_{x=0}^{2^n-1}|x\rangle\otimes|f(x)\rangle\ .$$

Come al solito, dopo U_f abbiamo una sovrapposizione di tutti i possibili valori di $f(x)$ in un colpo solo. Ora facciamo una misura sull'output register, cioè su $|f(x)\rangle$: dalla meccanica quantistica, che valuta tutti i valori di $f(x)$ all'interno della black-box, otteniamo un valore random della funzione

$$f_0 = f(x_0) = f(x_0 + r)\,;$$

ma questa funzione, in realtà, è valutata in differenti valori di x in quanto periodica: abbiamo trovato diversi valori $x_0 + jr$ dell'input register che sono associati al medesimo output; più precisamente il vincolo che deve essere soddisfatto è che $0 \leqslant x_0 + jr \leqslant 2^n$. Chiaramente il numero preciso di valori $x_0 + jr$ dipende da quanto 2^n è più grande rispetto a r: supponiamo di aver trovato m valori di output, allora, siccome $r \leqslant N$ e $2^n \sim N^2$, asintoticamente avremo $0 \leqslant x_0 + jN \lesssim N^2$ e quindi m sarà dell'ordine di N (un numero molto grande). Per cui il nostro stato complessivo è collassato in

$$|\psi\rangle = \underbrace{\frac{1}{\sqrt{m}} \sum_{k=0}^{m-1} |x_0 + kr\rangle}_{\text{Data register}} \otimes \underbrace{|f(x_0)\rangle}_{\substack{\text{Output} \\ \text{register}}}\,. \tag{3.8.1}$$

A questo punto lo stato $|f(x_0)\rangle$ è lo stesso per qualsiasi valore di $|x_0 + kr\rangle$, perciò nella discussione che segue è irrilevante e possiamo dimenticarcene. Ricordiamo che il nostro scopo è quello di ottenere r: se ora si effettuasse una misura si otterrebbe $x_0 + kr$, il quale sarebbe un ottimo risultato se non ci fosse il numero casuale x_0, il quale non conosciamo. Analogamente sarebbe bello poter effettuare due misurazioni (non lo possiamo fare per la regola di Born e il teorema di No-cloning): gli ipotetici risultati $x_0 + kr$ e $x_0 + k'r$ potrebbero essere sottratti per ottenere la differenza $(k - k')r$, la quale è un multiplo del periodo cercato. Naturalmente, se eseguissimo di nuovo l'intero algoritmo, ci ritroveremmo con uno stato della forma (3.8.1) per un altro valore casuale di x_0, che non consentirebbe alcun confronto utile con quanto appreso dalla prima esecuzione. In realtà possiamo fare qualcosa di più allo stato (3.8.1) prima di effettuare la misurazione finale. Come evidenziato, il problema risiede nella presenza del numero casuale x_0, che trasla kr e impedisce di estrarre qualsiasi informazione su r in una singola misura. Abbiamo bisogno di una trasformazione unitaria che trasformi la dipendenza da x_0 in un fattore di fase complessivo (e innocuo). Ciò si ottiene applicando la Quantum Fourier Transform in (3.7.1) a (3.8.1):

$$\begin{aligned} U_{\text{FT}} \left(\frac{1}{\sqrt{m}} \sum_{k=0}^{m-1} |x_0 + kr\rangle \right) &= \frac{1}{\sqrt{m} 2^{\frac{n}{2}}} \sum_{y=0}^{2^n-1} \sum_{k=0}^{m-1} e^{\frac{2\pi i}{2^n}(x_0 + kr)y} |y\rangle \\ &= \sum_{y=0}^{2^n-1} \underbrace{e^{2\pi i \frac{x_0 y}{2^n}} \sum_{k=0}^{m-1} \frac{e^{2\pi i \frac{kry}{2^n}}}{\sqrt{m} 2^{\frac{n}{2}}}}_{\text{coefficiente di ogni } |y\rangle} |y\rangle\ ; \end{aligned}$$

in questo modo abbiamo ottenuto una sovrapposizione di tutti i possibili interi nella base computazionale, i cui coefficienti sono dati dai fattori sottolineati. Se ora effettuiamo una misura, la probabilità $P(|y\rangle)$ di ottenere il risultato y è data dal modulo quadro dell'ampiezza del coefficiente di $|y\rangle$:

$$P(|y\rangle) = \left| e^{\frac{2\pi i}{2^n}(x_0 y)} \sum_{k=0}^{m-1} \frac{e^{\frac{2\pi i}{2^n}(kry)}}{\sqrt{m} 2^{\frac{n}{2}}} \right|^2 = \frac{1}{m 2^n} \left| \sum_{k=0}^{m-1} e^{\frac{2\pi i}{2^n}(kry)} \right|^2\ ; \tag{3.8.2}$$

è evidente come lo scomodo x_0 sia scomparso a seguito del fatto che apparisse unicamente come una pura fase all'interno del modulo quadro. Studiamo la probabilità (3.8.2) in dettaglio.
Un esempio di plot è mostrato nel grafico 3.1. Notiamo che vi sono differenti picchi di

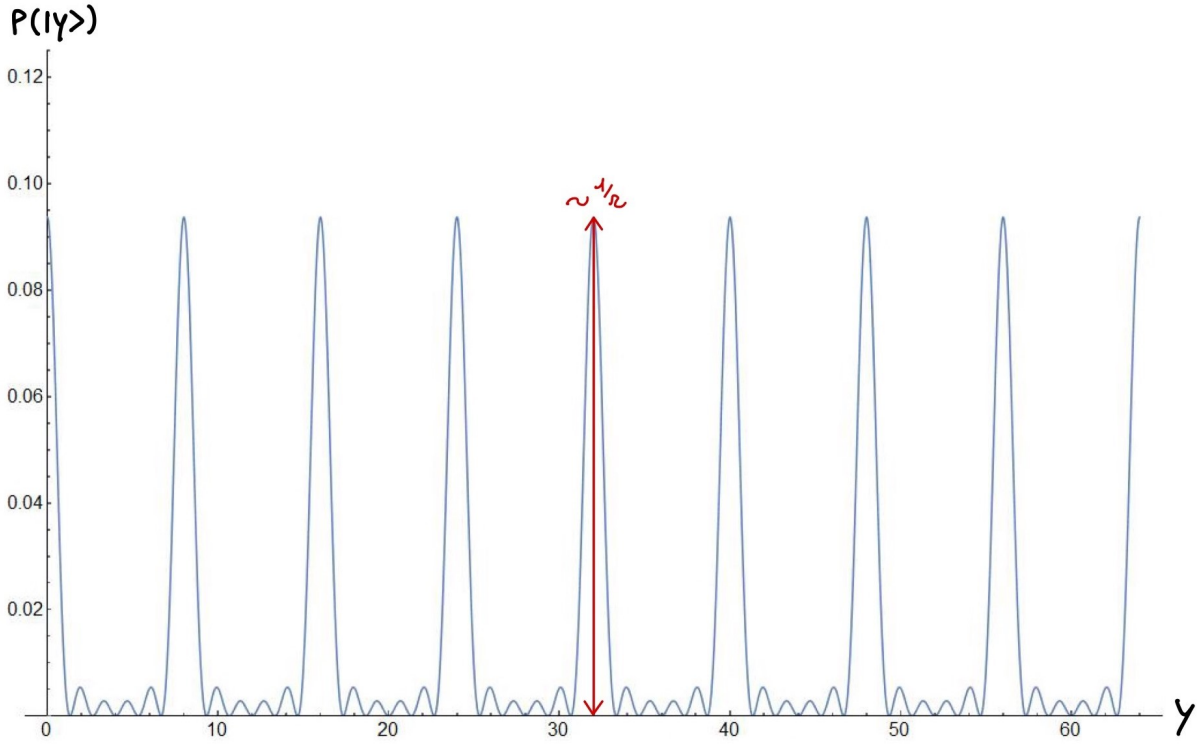


Figura 3.1: Esempio di grafico della probabilità di misurare y della formula (3.8.2), dove $0 \leqslant y \leqslant 2^n$. In questo esempio si è usato $n = m = 6$ e $r = 8$. Si noti come il numero di picchi sia $\mathcal{O}(r)$ e la loro altezza, come ordine di grandezza, comparabile a $1/r$.

diverse ampiezze: ciò è dovuto al fatto che ci sono valori di y dove abbiamo interferenza costruttiva, mentre in altri si ha interferenza distruttiva. Nel realizzare tale grafico, però, abbiamo assunto che $y \in \mathbb{R}$, ma bisogna ricordare che $y \in \mathbb{Z}$, quindi si tratta di un'approssimazione perché non tutti i punti della curva devono essere disegnati ! In realtà, per valori di n molto grandi 2^n è enorme quindi la discretizzazione è minima e quasi impercettibile. I punti in cui la probabilità è maggiore, in corrispondenza dei picchi più alti, si ha $y = j\frac{2^n}{r}$ dove $j \in \mathbb{N}$. Per tali valori, gli esponenziali dentro la probabilità in (3.8.2) non sono altro che $e^{2\pi ijk} = 1$ perché $j, k \in \mathbb{Z}$, quindi la (3.8.2) diventa:

$$P(|y\rangle) = \frac{1}{m2^n}\left|\sum_{k=0}^{m-1} 1\right|^2 = \frac{m^2}{m2^n} = \frac{m}{2^n}\,,$$

il quale è il valore massimo della probabilità. Per capire per quale ragione abbiamo indicato nel grafico 3.1 che l'altezza dei picchi è di circa $1/r$, ricordiamo che $2^n \sim N^2$, $r \sim N$ e anche $m \sim N$, per cui:

$$P(|y\rangle) = \frac{m}{2^n} \sim \frac{N}{N^2} \sim \frac{1}{N} \sim \frac{1}{r}\,.$$

Notiamo che sebbene abbiamo disegnato una funzione continua, l'approssimazione è comunque molto buona perché la probabilità totale è correttamente normalizzata a 1: infatti

l'altezza dei picchi per il loro numero non è altro che $\frac{1}{r} \times m \sim \frac{1}{r} \times r \simeq 1$, quindi gran parte della probabilità è saturata i corrispondenza dei picchi (si può dimostrare che nel limite in cui $n \to \infty$ i picchi tendono a delle delta function).
Un risultato fondamentale è che passando attraverso delle manipolazioni algebriche di seno e coseno, si può dimostrare che si ha circa il 40% di possibilità ($P(|y\rangle) = \frac{4}{\pi^2}$) di misurare y e ottenere un valore che si trovi in prossimità di uno di questi picchi con un errore di circa $\frac{1}{2}$: ricordando che i picchi sono situati in $y = j\frac{2^n}{r}$, possiamo formalmente scrivere che

$$\left| y - j\frac{2^n}{r} \right| < \frac{1}{2}\,, \quad \Rightarrow \quad \left| \frac{y}{2^n} - \frac{j}{r} \right| < \frac{1}{2^{n+1}}\,, \tag{3.8.3}$$

dove abbiamo diviso per 2^n. Stiamo quindi dicendo che una misura di y soddisfa la disuguaglianza (3.8.3) il 40% delle volte. Possiamo estrarre r dalla (3.8.3) ? Innanzitutto notiamo che, essendo $0 \leqslant y \leqslant 2^n$, abbiamo $0 \leqslant \frac{y}{2^n} \leqslant 1$. Se $n = 2n_0$ allora avremo $2^n = 2^{2n_0} = N^2$, quindi quando la (3.8.3) è verificata esiste un singolo numero razionale della forma $\frac{j}{r}$ che la soddisfa e dal quale è possibile estrarre r.
La logica è mostrata nel disegno della figura 3.2: prendeno il segmento $[0, 1]$ e suddividendolo in step uguali di lunghezza $\frac{1}{2^n}$, possiamo rappresentare con delle barrette verticali tutti i possibili valori di y tali che $0 \leqslant \frac{y}{2^n} \leqslant 1$. La disuguaglianza (3.8.3) ci dice che il numero $\frac{j}{r}$ (indicato con una "×" azzurra) è vicino a $\frac{y}{2^n}$ con una distanza minore di $\frac{1}{2^{n+1}}$.

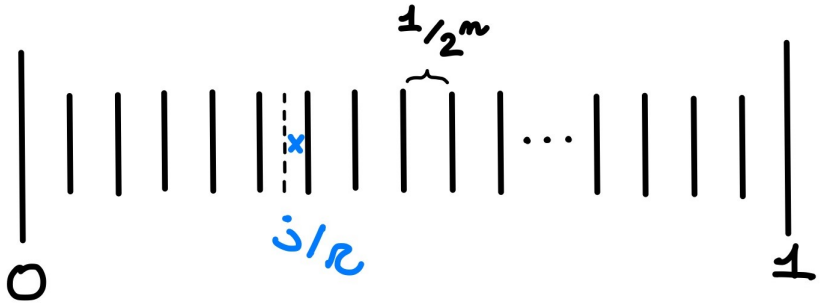


Figura 3.2: Rappresentazione geometrica della disuguaglianza (3.8.3).

La domanda che possiamo porci è se esista più di un numero razionale che soddisfi questa particolare proprietà. Supponiamo per assurdo che esistano due numeri razionali $\frac{j_1}{r_1}$ e $\frac{j_2}{r_2}$ che soddisfano la disuguaglianza (3.8.3) (e la condizione per cui $r_1, r_2 < N$). Allora la differenza tra questi due numeri è

$$\frac{j_1}{r_1} - \frac{j_2}{r_2} = \frac{r_2 j_1 - r_1 j_2}{r_1 r_1}\,, \tag{3.8.4}$$

tuttavia il numeratore è un intero e il denominatore è di ordine $\mathcal{O}(N^2)$: essendo $r_1, r_2 < N$ allora il denominatore è strettamente minore di N^2 quindi la (3.8.4) è $\geqslant \frac{1}{N^2} = \frac{1}{2^n}$, che è proprio la larghezza degli step in cui abbiamo suddiviso $[0, 1]$. Questo significa che se ci sono due soluzioni della (3.8.3) allora la distanza tra le due deve necessariamente essere più grande della misura dello step: in un dato step è possibile trovare una sola soluzione, ossia un solo numero razionale che verifica la (3.8.3).
Riassumendo: misurando un valore y che soddisfa la (3.8.3) otteniamo un unico numero razionale $\frac{j}{r}$. Il punto chiave è quindi trovare $\frac{j}{r}$, tuttavia questo non è così semplice, perché quello che si ottiene dalla misura è un numero scritto in forma decimale, il quale vorremmo

poterlo scrivere come rapporto tra razionali. Nonostante ciò facciamo uso del seguente risultato di teoria dei numeri che non dimostreremo. Riscriviamo la disuguaglianza (3.8.3) come

$$\left| x - \frac{j}{r} \right| \leqslant \frac{1}{2r^2} , \quad \text{dove } x \in [0,1] .$$

Il valore x è dato dalla misura mentre lo scopo è quello di trovare un numero razionale $\frac{j}{r}$ che soddisfi questa disuguaglianza. Il risultato dalla teoria dei numeri asserisce che questo valore appare nell'espansione in frazione continua del numero x:

$$x = \cfrac{1}{x_0 + \cfrac{1}{x_1 + \cfrac{1}{x_2 + \dots}}} .$$

Esempio 3.5 (Espansione in frazione continua)**.** *Supponiamo di considerare il numero $x = 0.256789$. Prendendone l'inverso avremo $\frac{1}{x} = \frac{1}{0.256789} = 3.8942478\dots$. Da questo risultato è evidente che $x_0 = 3$. Nello step successivo si calcola x_1: calcoliamo $\frac{1}{x} - 3$ e poi $\left(\frac{1}{x} - 3\right)^{-1}$, la cui parte intera è x_1. Iterando all'infinito questo procedimento si ottiene l'espansione in frazione continua di x:*

$$x = \cfrac{1}{3 + \cfrac{1}{1 + \cfrac{1}{8 + \dots}}} .$$

Ad ogni step si ottiene quindi un'approssimazione razionale di x, infatti il set di numeri razionali che approssimano il suo valore è dato da $\left\{\frac{1}{3}, \frac{1}{4}, \frac{9}{35}, \frac{19}{74}, \frac{104}{405}, \dots\right\}$. Più si procede, meglio approssimato sarà il valore di x. Il teorema ci dice che a un certo punto possiamo trovare il valore di $\frac{j}{r}$ all'interno di questo insieme e questo con una probabilità del 100%.

Tuttavia di questo procedimento vanno fatte delle opportune precisazioni:

- Supponiamo che j e r abbiano dei fattori comuni (e questo ovviamente non lo sapremo mai), allora il valore di r può essere diverso, infatti:

  $$\frac{j}{r} = \frac{j_0 k}{r_0 k} = \frac{j_0}{r_0} ,$$

  quindi mediante la procedura sopraelencata, al posto di trovare il periodo cercato, si ottiene r_0. Nonostante ciò si è trovato un divisore di r, quindi prendendo la forma analitica di f si può provare a calcolare $f(x + r_0)$, $f(x + 2r_0)$, $f(x + 3r_0)$, ... fino a quando effettivamente si trova un valore uguale a $f(x)$.

- Talvolta si possono misurare dei valori y che non soddisfano la disuguaglianza (3.8.3) (la probabilità di soddisfarla è infatti del 40%). In tal caso basta semplicemente ricominciare l'algoritmo da capo con una nuova esecuzione del circuito e della QFT fino a quando non si ottiene un valore di y che soddisfi la (3.8.3). Inoltre è possibile dimostrare che la probabilità che la (3.8.3) sia soddisfatta cresce fino al 90% se si sa a priori che $r < \frac{N}{2}$.

Concludiamo la discussione dicendo che lo stesso tipo di algoritmo può essere utilizzato per calcolare i *logaritmi discreti*, oppure per la simulazione di sistemi quantistici (si possono calcolare gli autovalori di matrici unitarie molto grandi che servono per il calcolo degli autovalori delle relative hamiltoniane e degli evoluti temporali).

Lezione 8 - 29/10/2021

3.8.1 Violazione della crittografia RSA

Dal momento che l'algoritmo quantistico del period finding di Shor è spesso descritto come un algoritmo di fattorizzazione, concludiamo questa sezione osservando come il period finding porti alla fattorizzazione. Consideriamo solo il caso relativo alla **crittografia RSA** (R. Rivest, A. Shamir e L. Adleman), dove si vuole fattorizzare il prodotto di due grandi numeri primi, sebbene la connessione tra period finding e fattorizzazione sia più generale.
Supponiamo di avere un numero $N = pq$, tale per cui N, p e q siano molto grandi e p, q siano fattori primi. Nel CC fattorizzare N richiederebbe un tempo di esecuzione esponenziale mentre nel QC ci si riduce a un tempo polinomiale. Vediamo ora come interviene l'algoritmo di Shor.
Prendiamo una funzione $f(x) = a^x \pmod N$, di periodo r ($f(x + r) = f(x)$) e dove r è tale per cui:

- $a^r \equiv 1 \pmod N$;
- Se r è la più piccola soluzione del punto precedente allora r è l'ordine di $a \pmod N$;
- r è definito come $r < N$.

Quest'ultimo risultato è ottenuto a partire dalla teoria dei numeri nel caso particolare in cui N è prodotto di due numeri primi, proprio come nel nostro caso.

Introduciamo il seguente teorema

Teorema 3.1 (**Teorema di Eulero**). *Se prendiamo a coprimo a p e q, allora a può essere espresso come*

$$a^{(p-1)(q-1)} = 1 \pmod{pq}$$

Osserviamo che con le assunzioni fatte all'inizio e su r siamo nelle ipotesi del teorema di Eulero, l'algoritmo di Shor può essere dunque utilizzato per trovare l'ordine r di un intero $a \pmod N$.
Iniziamo con il considerare $N = pq$, vogliamo trovare i valori di p e q. Scegliamo a coprimo a N, verificare che siano coprimi tra loro non è difficile, esiste un teorema di Euclide molto famoso che può essere eseguito su un computer classico, ciò che è invece difficile è trovare tutti i possibili divisori di un numero N. A questo punto usiamo il QC per trovare l'ordine r di $a^r \equiv 1 \pmod N$, in particolar modo facciamo due assunzioni:

1. Supponiamo che r sia **pari**, sotto questa ipotesi possiamo definire l'intero $x = a^{\frac{r}{2}} \pmod N$ che ha la seguente proprietà:

   $$(\text{mod } N)0 = a^r - 1 = x^2 - 1 = (x - 1)(x + 1)$$

   Richiediamo che $x - 1 \neq 0 \pmod N$ perché se non fosse così, $a^{\frac{r}{2}} = 1 \pmod N$, l'ordine non sarebbe più r, ma $r/2$;

2. Supponiamo di essere fortunati e che anche $x + 1 \neq 0 \pmod N$.

Ora il prodotto $(x-1)(x+1)$ è un multiplo di N, ma i singoli fattori non sono un multiplo di N, in quanto $x+1 \neq 0 \pmod N$ e $x+1 \neq 0 \pmod N$. Tuttavia $(x-1)$ è multiplo di p e $(x+1)$ è multiplo di q. In particolare noi possiamo andare a calcolare classicamente il massimo comune divisore di

$$p = \gcd(x-1, N) \qquad q = \gcd(x+1, N)$$

Se non siamo fortunati, scegliamo un altro numero che soddisfi le due assunzioni.
Uno potrebbe porsi la seguente domanda: perché fattorizzare gli interi è importante? Perché è possibile violare la crittografia RSA.

Esempio 3.6 (**Protocollo RSA**). *Iniziamo facendo delle premesse: supponiamo che Bob possieda due grandi primi p e q, il cui prodotto sia un grande intero N. Sia c un numero che non ha alcun fattore in comune con $(p-1)(q-1)$. Consideriamo ora Alice, supponiamo che conosca N e c, ma non possieda alcuna informazione su p e q.*
Il protocollo RSA lavora nel seguente modo:

- *Alice possiede un messaggio codificato in a;*
- *Alice calcola $b = a^c \pmod N$;*
- *Bob decodifica il messaggio, può calcolare cd, dove $cd = 1 \pmod{(p-1)(q-1)}$;*
- *Si scopre che, in teoria dei numeri, c'è una dimostrazione che mostra il seguente risultato:*
$$a = b^d \pmod N$$
- *Se conosciamo b e se siamo in possesso di c e d, possiamo ricavare a, cioè il messaggio.*

Questo è come funziona il **protocollo RSA**. Per decodificare il messaggio necessitiamo d, ma per conoscere d abbiamo bisogno di p e q, questo perché conoscere solo N e c non è sufficiente per ricavare d. È qui che si evidenzia l'importanza di fattorizzare gli interi, se vogliamo decodificare il messaggio. La potenza del protocollo RSA sta nel fatto che nel CC sarebbero richieste risorse esponenziali per poter trovare i fattori primi mentre nel QC parliamo di tempi polinomiali.
L'algoritmo di Shor può violare altri protocolli crittografici, come ad esempio il **protocollo Diffie-Hellman** perché è in grado di risolvere i logaritmi discreti, utilizzati in quest'ultimo protocollo.

3.9 Algoritmo di Grover

L'ultimo algoritmo che affrontiamo per mostrare ancora una volta che le prestazioni del QC sono nettamente migliori rispetto a quelle del CC, è l'**algoritmo di ricerca di Grover**. L'idea è quella di avere a disposizione N oggetti e di cercarne uno specifico che identifichiamo con a. In maniera astratta potremmo pensare di avere una funzione $f(x)$ che può valutare i valori di x nell'insieme $\{0, \ldots, N-1\}$ e vale $f(x) = 1$ se $x = a$ mentre vale $f(x) = 0$ per $x \neq a$, in termini matematici:

$$f(x) : \{0, \ldots, N-1\} \to \{0, 1\}$$

$$f(x) = \begin{cases} 1 & x = a \\ 0 & x \neq a \end{cases}$$

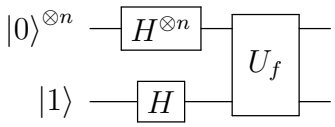
Questo tipo di algoritmo può essere utilizzato per cercare determinati prodotti su un marketplace oppure in un problema matematico.

Esempio 3.7 (**Problema matematico**). *Consideriamo un numero p definito come*

$$p = x^2 + y^2$$

dove x, y sono due numeri interi che vogliamo trovare. Per tutti gli x possiamo valutare $\sqrt{p^2 - x^2}$ fino a quando non troviamo un intero y.

Nel CC, questo tipo di algoritmo può essere risolto con una probabilità del 50% che corrisponde a $\frac{N}{2}$ operazioni, siccome $\frac{N}{2}$ è dell'ordine di N, in termini di esecuzione sono richieste $\mathcal{O}(N)$ operazioni. Dal punto di vista invece del QC, sono richieste $\mathcal{O}\left(\sqrt{N}\right)$ operazioni.
Iniziamo con il nostro solito circuito per capire come funzione l'algoritmo di Grover:



che produce lo stato

$$\frac{1}{2^{\frac{n}{2}}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle \otimes \underbrace{\frac{|0\rangle - |1\rangle}{\sqrt{2}}}_{\substack{\text{Termine} \\ \text{irrilevante}^{\text{vii}}}}.$$

Ricordiamo che l'azione di $f(x)$ produce i valori 0 o 1, in questo caso ciò che si realizza è un termine di fase. Nel gergo comune, si è soliti identificare la nostra black-box U_f con il nome **oracle** (O). La sua azione sarà quindi:

$$O|x\rangle = (-1)^{f(x)} |x\rangle = \begin{cases} -|x\rangle & x = a \\ |x\rangle & x \neq a \end{cases}$$

La sua azione e il resto dell'algoritmo può essere descritto in due modi:

Interpretazione geometrica

L'applicazione di O su un generico stato $|x\rangle$ può essere vista come una riflessione rispetto a un asse. Consideriamo lo spazio di $\mathcal{H}$ come formato dallo spazio contenente soltanto $|a\rangle$ e dallo spazio ortogonale ad $|a\rangle$ che definiamo come:

$$\{|a\rangle\}^{\perp} = \{|x\rangle \in \mathcal{H} : \langle a|x\rangle = 0\}$$

Un generico vettore $|x\rangle \in \mathcal{H}$ può essere rappresentato all'interno di questo piano. [GRAFICO]
L'azione di O può essere interpretata in questo modo:

- Se $|x\rangle = |a\rangle$, allora O esegue una riflessione rispetto all'asse $\{|a\rangle\}^{\perp}$;

- Se $|x\rangle \neq |a\rangle$, allora O non fa nulla, rimane uguale a se stesso.

Possiamo definire O come

$$O = \mathbb{I} - 2\,|a\rangle\langle a|$$

Vediamo la sua azione

$$O\,|a\rangle = |a\rangle - 2\,|a\rangle \underbrace{\langle a|a\rangle}_{1} = |a\rangle - 2\,|a\rangle = -\,|a\rangle \quad \text{riflessione}$$

$$O\,|x\rangle = |x\rangle - 2\,|a\rangle \underbrace{\langle a|x\rangle}_{0} = |x\rangle \qquad \text{nessuna riflessione}$$

Ora, il nostro stato iniziale era

$$|0\rangle^{\otimes n} \to H^{\otimes n}\,|0\rangle^{\otimes n} = \frac{1}{2^{\frac{n}{2}}} \sum_{x=0}^{2^n-1} \equiv |\phi\rangle$$

Definiamo un'altra operazione di riflessione G rispetto a $|\phi\rangle$ tale per cui

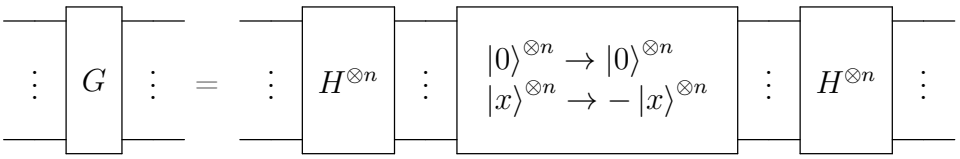
$$G\,|\phi\rangle = \phi$$
$$\langle x|\phi\rangle = 0 \to G\,|x\rangle = -\,|x\rangle$$

In questo modo vediamo che G è della forma:

$$G = 2\,|\phi\rangle\langle\phi| - \mathbb{I}$$

Questo operatore può essere costruito in termini di gate in maniera semplice ricordando che

$$|0\rangle^{\otimes n} \to |0\rangle^{\otimes n}$$
$$|x\rangle^{\otimes n} \to -\,|x\rangle^{\otimes n} \qquad x \neq 0$$



Troviamo che

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

Dal momento che `H-gate` è unitario e $|\phi\rangle = H^{\otimes n}\,|0\rangle^{\otimes n}$. Abbiamo quindi due operazioni O e G che agiscono così [GRAFICO] L'algoritmo di Grover è quindi l'applicazione ripetuta k volte di GO, ricordiamo che l'applicazione viene eseguita da destra verso sinistra, per cui avremo $\underbrace{\ldots GOGOGO}_{k} = (GO)^k$. Dopo $\mathcal{O}\left(\sqrt{N}\right)$ applicazioni, vedremo il nostro elemento $|a\rangle$ con probabilità prossima all'unità. Supponiamo che $|\phi\rangle = \frac{1}{2^{\frac{n}{2}}} \sum_{x=0}^{2^n-1} |x\rangle$, allora il prodotto scalare tra ϕ e a sarà

$$\langle a|\phi\rangle = \frac{1}{2^{\frac{n}{2}}} \ll 1 \quad \text{numero piccolo}$$

Possiamo quindi dire che ϕ è quasi ortogonale ad a. Dal momento che

$$\theta \approx \sin\theta = \cos\left(\frac{\pi}{2} - \theta\right) \Rightarrow \langle a|\phi\rangle = \cos\left(\frac{\pi}{2} - \theta\right) = \frac{1}{2^{\frac{n}{2}}} \ll 1 \Rightarrow \theta \approx \frac{1}{2^{\frac{n}{2}}}$$

Siccome i nostri oggetti nel database sono $N = 2^n$ (con N grande e n numero di qubits), allora $\theta \sim \frac{1}{\sqrt{N}}$. L'azione di O e G dopo, è una doppia riflessione, cioè una rotazione. Per cui dopo l'azione di O e poi G, lo stato ϕ passerà da un angolo θ a un angolo 3θ. Applicando nuovamente la sequenza O e poi G passerà da un angolo 3θ a 5θ. Iterando questo procedimento per un certo numero di volte che indichiamo con r, avremo che $\theta + 2r\theta \approx \frac{\pi}{2}$. Dopo quante iterazione otteniamo $\frac{\pi}{2}$? $r = \frac{\pi}{4\theta} - \frac{1}{2}$. Ma θ è piccolo, per cui:

$$r \approx \frac{\pi}{4\theta} = \frac{\pi}{4}\theta^{-1} \approx \frac{\pi}{4}\sqrt{N}$$

Abbiamo che r è quasi allineato con $|a\rangle$. Dopo $\mathcal{O}\left(\sqrt{N}\right)$ iterazioni di GO, arriviamo allo stato $|a\rangle$ con probabilità quasi 1. Possiamo calcolare $|a\rangle$ attraverso una misura finale dello stato $(GO)^r |\phi\rangle$, infatti

$$(GO)^r |\phi\rangle = \sqrt{1-\varepsilon^2}\,|a\rangle + \varepsilon\,|a\rangle^{\perp} \quad \varepsilon \ll 1$$

Se siamo sfortunati e non otteniamo $|a\rangle$, possiamo eseguire nuovamente l'algoritmo magari incrementando il numero di iterazioni e provare a misurare nuovamente. Dopo un numero limitato di tentativi troveremo a!

Interpretazione grafica: inversione rispetto alla media

Iniziamo con il considerare un generico stato $|\psi\rangle = \sum \alpha_x |x\rangle$. L'operatore O inverte l'ampiezza per $x = a$

$$\begin{cases} \alpha_x \to -\alpha_x & x = a \\ \alpha_x \to \alpha_x & x \neq a \end{cases}$$

L'operatore G è una riflessione lungo lo stato con uguale ampiezza $\alpha_x = \frac{1}{\sqrt{N}}$

$$\begin{aligned} &G = 2\,|\phi\rangle\langle\phi| - \mathbb{I} \\ &|\phi\rangle = \tfrac{1}{\sqrt{N}} \textstyle\sum_{x=0}^{2^n-1} |x\rangle = \tfrac{1}{2^{\frac{n}{2}}} \sum_{x=0}^{2^n-1} |x\rangle \end{aligned}$$

E la sua azione su ψ non è molto complicata

$$\begin{aligned} G\,|\psi\rangle &= 2\frac{1}{N}\sum_{x=0}^{2^n-1} |x\rangle \sum_{y=0}^{2^n-1} \langle y| \left(\sum_{z=0}^{2^n-1} \alpha_z\,|z\rangle\right) - \sum_{x=0}^{2^n-1} \alpha_x\,|x\rangle \\ &= \sum_{x=0}^{2^n-1} \left(2\sum_{y=0}^{2^n-1} \frac{\alpha_y}{N} - \alpha_x\right) |x\rangle \\ &= \sum_{x=0}^{2^n-1} \left(2\,\langle\alpha\rangle - \alpha_x\right) |x\rangle \end{aligned}$$

dove $\langle\alpha\rangle = \sum_{x=0}^{2^n-1} \frac{\alpha_x}{N}$. Perciò su ogni componente

$$\alpha_x \to 2\,\langle\alpha\rangle - \alpha_x$$

c'è una riflessione rispetto all'ampiezza media.