Capitolo 6: Computazione quantistica

I computer oggi in circolazione sottostanno alle leggi proprie di quella branca della fisica che studia i fenomeni macroscopici ovvero alle leggi della fisica classica.

Il computer quantistico è, invece, un elaboratore il cui funzionamento segue le leggi della meccanica quantistica.

Tali leggi giocano un ruolo fondamentale nel fornire alla computazione potenzialità sinora non ottenibili classicamente.

Pertanto prima di poter analizzare o, addirittura, implementare un algoritmo quantistico è indispensabile acquisire la nuova logica di calcolo.

Se è concesso un paragone, è come la differenza tra un elicottero e un aereoplano: entrambi volano ma ognuno secondo schemi differenti. Pertanto un pilota di aereo, prima di poter far volare un elicottero, dovrà apprendere nuovi schemi di funzionamento ed imparare a governarli.

§.26 Notazione di Dirac

Il grosso ostacolo all'assimilazione dei postulati della meccanica quantistica non risiede tanto nei contenuti quanto nel formalismo e in particolare nell'insolita notazione dovuta a Dirac (e ormai adottata da tutti i fisici).
Di seguito quindi riportiamo una tabella di decodifica.

Notazione	**Descrizione**
z^*	complesso coniugato del numero $z \in \mathbb{C}$
$\lvert y \rangle$	vettore (ket)
$\langle y \rvert$	vettore duale del vettore $\lvert y \rangle$ (bra)
A^*	complesso coniugato della matrice A
A^T	trasposta della matrice A
$A^\dagger = \left(A^T\right)^*$	aggiunta (trasposta coniugata) della matrice A
$\langle y \lvert j \rangle$	prodotto interno (bra-ket) tra $\lvert y \rangle$ e $\lvert j \rangle$
$\langle j \lvert M \lvert y \rangle$	prodotto interno tra $\lvert j \rangle$ e $M \lvert y \rangle$ oppure tra $M^\dagger \lvert j \rangle$ e $\lvert y \rangle$
$\lvert j \rangle\langle y \rvert$	prodotto esterno
$\lvert j \rangle \otimes \lvert y \rangle$	prodotto tensore

§.27 Definizioni "spicciole"

Per la comprensione di quanto segue è necessario conoscere almeno a livello intuitivo il significato di alcuni concetti quali lo spazio di Hilbert e il prodotto tensore.
Poichè nel nostro caso avremo a che fare unicamente con spazi a dimensione finita, a questo caso particolare nel seguito ci limiteremo.

Ecco quindi le definizioni "spicciole" di questi concetti. Per una trattazione formale e rigorosa si rimanda a testi specifici.

La definizione di **spazio di Hilbert** è assai semplice: è uno spazio vettoriale completo munito di prodotto interno. La condizione di completezza diventa superflua nel caso finito-dimensionale.

La definizione di **prodotto tensore** è invece leggermente più complicata: dati due spazi vettoriali finito-dimensionali $\mathcal{U}$ e $\mathcal{V}$ che insistono sul medesimo campo e aventi, rispettivamente, basi ortonormali $\{u_i\}_{i=1}^{n}$ e $\{v_j\}_{j=1}^{m}$ chiameremo prodotto tensore $\mathcal{U}\otimes\mathcal{V}$ lo spazio vettoriale delle combinazioni lineari (a coefficienti sul campo originale) di tutte le possibili coppie ordinate $\{u_i\otimes v_j\}$ di vettori delle basi ortonormali di $\mathcal{U}$ e di $\mathcal{V}$ con la proprietà di bilinearità ovvero:

$$\left(\sum_{i=1}^{n} l_i u_i\right)\otimes v=\sum_{i=1}^{n} l_i\left(u_i\otimes v\right)\ \text{e}\ u\otimes\left(\sum_{j=1}^{m} l_j v_j\right)=\sum_{j=1}^{m} l_j\left(u\otimes v_j\right) \tag{90}$$

Due osservazioni fondamentali ci saranno utili nel seguito. Le esporremo confrontando il prodotto tensore $\mathcal{U}\otimes\mathcal{V}$ col prodotto cartesiano $\mathcal{U}\times\mathcal{V}$.

Una base ortonormale di $\mathcal{U}\times\mathcal{V}$ è $\{u_1,\ldots,u_n,v_1,\ldots,v_m\}$ da cui ricaviamo che $\dim\mathcal{U}\times\mathcal{V}=\dim\mathcal{U}+\dim\mathcal{V}$ mentre una base ortonormale di $\mathcal{U}\otimes\mathcal{V}$ è $\{u_1\otimes v_1,\ldots,u_1\otimes v_m,u_2\otimes v_1,\ldots,u_n\otimes v_m\}$ da cui risulta evidente che $\dim\mathcal{U}\otimes\mathcal{V}=\dim\mathcal{U}\cdot\dim\mathcal{V}$.
Quindi tranne nel caso particolare $n=m=2$ risulterà sempre $\dim\mathcal{U}\otimes\mathcal{V}\neq\dim\mathcal{U}\times\mathcal{V}$.

Per la seconda osservazione restiamo nel caso $n=m=2$.
Dato un qualsiasi vettore $w\in\mathcal{U}\times\mathcal{V}$ allora $w=au_1+bu_2+cv_1+dv_2$ e quindi esiste sempre una coppia $u=au_1+bu_2$ e $v=cv_1+dv_2$ tale che $w=u\times v$.
Invece un qualsiasi vettore $w\in\mathcal{U}\otimes\mathcal{V}$ può essere ovviamente scritto come combinazione lineare della base ortonormale di $\mathcal{U}\otimes\mathcal{V}$ ovvero $w=\sum_{i,j=1}^{2} l_{ij}u_i\otimes v_j$ tuttavia non è assolutamente detto che esista una coppia u e v tale che $w=u\otimes v$.
Infatti la coppia u,v che fattorizza w esiste solo a certe condizioni che, nel caso in esame, sono: $l_{11}l_{22}=l_{12}l_{21}$.

Ad esempio il vettore $w = u_1 \otimes v_1 + u_2 \otimes v_2$ (in cui $l_{11} = l_{22} = 1$ e $l_{12} = l_{21} = 0$) non fattorizza in alcun modo.

§.28 I postulati della meccanica quantistica

Ora siamo pronti per acquisire gli schemi di comportamento della materia a livello quantistico.

Non sarà necessario studiare la meccanica quantistica. Basterà soltanto assimilare quattro suoi postulati che ci consentiranno di determinare il modello computazionale.

Il primo postulato definisce l'ambito in cui si colloca la meccanica quantistica:

1) *ad ogni sistema quanto-meccanico isolato è associato uno spazio vettoriale complesso munito di prodotto interno, cioè uno spazio di Hilbert, noto come spazio degli stati del sistema. Il sistema è completamente descritto dal suo vettore di stato che è un vettore unitario appartenente allo spazio degli stati.*

Il secondo postulato definisce come lo stato di un sistema quanto-meccanico cambia nel tempo:

2) *L'evoluzione di un sistema quanto-meccanico isolato è descritto da una trasformazione unitaria. In altri termini lo stato* $|y\rangle$ *del sistema all'istante* t_1 *è collegato allo stato* $|y'\rangle$ *all'istante* t_2 *da un operatore unitario* $U(t_1, t_2)$ *ovvero dalla relazione:* $|y'\rangle = U|y\rangle$.

Questo postulato richiede che il sistema descritto sia isolato. Ciò significa che non deve interagire in alcun modo con altri sistemi. Nella realtà ciò non accade mai perché ogni sistema (escludendo, ovviamente, l'intero universo) interagisce anche se in minima parte con altri sistemi. Comunque ci sono un buon numero di sistemi che possono essere descritti con buona approssimazione da un sistema isolato, la cui evoluzione può, pertanto, essere descritta da operatori unitari con approssimazione altrettanto buona. Ricordiamo che una trasformazione U è detta unitaria se $U^\dagger U = I$.

Il terzo postulato ci dice come effettuare delle misurazioni sul sistema e in quale stato il sistema si troverà dopo tali misurazioni:

3) *Le misurazioni di un sistema quanto-meccanico relative ad un fissato esperimento sono descritte da una collezione* $\{M_m\}$ *di operatori di proiezione agenti sullo spazio degli stati del sistema oggetto di misurazione. L'indice* m *fa riferimento ai valori da misurare risultanti dall'esperimento. Se lo stato del sistema quanto-meccanico è* $|y\rangle$ *immediatamente prima della misurazione allora la probabilità che* m *sia il valore risultante è data da* $p(m) = \langle y | M_m^\dagger M_m | y \rangle$ *e lo stato del sistema dopo la misurazione è* $\frac{M_m |y\rangle}{\sqrt{p(m)}}$. *L'operatore di misurazione deve*

soddisfare l'equazione di completezza $\sum_m M_m^\dagger M_m = I$ *che esprime la condizione che la somma delle probabilità sia pari a 1 indipendentemente dallo stato del sistema cioè* $\sum_m p(m) = 1 \ \forall \ |y\rangle$.

Il quarto ed ultimo postulato ci dice come costruire lo spazio degli stati di un sistema composto a partire dallo spazio degli stati che lo compongono:

4) *Lo spazio degli stati di un sistema quanto-meccanico composto è il prodotto tensoriale degli spazi degli stati dei sistemi componenti. Inoltre, se* $|y_i\rangle$ *rappresenta lo stato dell'i-esimo sistema componente, lo stato del sistema composto sarà dato da* $|y_1\rangle \otimes |y_2\rangle \otimes \cdots \otimes |y_n\rangle$.

§.29 Qubit

Introduciamo innanzi tutto un concetto nuovo ovvero il **quanto di informazione**. Con quanto di informazione si intende la più piccola porzione in cui una qualsiasi informazione può essere scomposta ed è quindi l'unità di misura dell'informazione codificata.

Così come il **bit** è il quanto di informazione della computazione classica, la computazione quantistica si basa su un concetto analogo: il *quantum bit* o *qubit* che ne è la contrazione. Al pari del bit, il **qubit** è un oggetto matematico con determinate specifiche proprietà. Il vantaggio nel trattare i qubit come entità astratte risiede nella libertà di costruire una teoria generale della computazione quantistica che non dipende dagli specifici sistemi utilizzati per la sua realizzazione.

Così come il bit classico ammette due stati, cioè lo stato (0) e lo stato (1), altrettanto accade al qubit. Per analogia con il caso classico chiameremo questi due stati $|0\rangle$ e $|1\rangle$. Ma grazie al **principio di sovrapposizione**, che emerge dal primo postulato, è anche possibile combinare linearmente i due stati $|0\rangle$ e $|1\rangle$ per ottenere lo stato di sovrapposizione:

(91) $$|y\rangle = a|0\rangle + b|1\rangle$$

in cui a e b sono due numeri complessi tali per cui

(92) $$|a|^2 + |b|^2 = 1.$$

Detto in altri termini, lo stato di un qubit è un vettore unitario dello spazio degli stati hilbertiano di dimensione 2 in cui gli stati speciali $|0\rangle$ e $|1\rangle$ formano una base ortonormale detta *base computazionale*.

Nel caso classico è sempre possibile esaminare un bit per determinare se esso sia nello stato (0) o nello stato (1). Di converso, nel caso quantistico, non è possibile esaminare un qubit per determinarne il suo stato, cioè per determinare i due coefficienti a e b. Il

terzo postulato ci dice che è possibile acquisire una quantità più limitata di informazioni relative allo stato quantistico. Quando misuriamo lo stato di un qubit possiamo ottenere il risultato $|0\rangle$ con una probabilità $|a|^2$ o il risultato $|1\rangle$ con probabilità $|b|^2$.

Proviamo ad applicare le regole dettate dal terzo postulato in questo semplice ma significativo caso. Abbiamo già visto che la misurazione può avere soltanto due esiti definiti dai due operatori di misurazione $M_0 = |0\rangle\langle 0|$, $M_1 = |1\rangle\langle 1|$.

Notiamo che ogni operatore di misurazione è hermitiano ($M^\dagger = M$) e che $M_0^2 = M_0$, $M_1^2 = M_1$ e ciò ci garantisce che la condizione di completezza è soddisfatta.

Supponiamo che lo stato oggetto di misurazione sia $|y\rangle = a|0\rangle + b|1\rangle$. Allora la probabilità di ottenere $|0\rangle$ come risultato della misurazione è data da $p(0) = \langle y | M_0^\dagger M_0 | y \rangle = \langle y | M_0 | y \rangle = |a|^2$. Analogamente la probabilità di ottenere $|1\rangle$ è data da $p(1) = |b|^2$.

Lo stato del sistema dopo la misurazione sarà, nei due casi rispettivamente: $\frac{M_0|y\rangle}{|a|} = \frac{a}{|a|}|0\rangle$ e $\frac{M_1|y\rangle}{|b|} = \frac{b}{|b|}|1\rangle$ dove i coefficienti $\frac{a}{|a|}$ e $\frac{b}{|b|}$ sono fattori di fase che non incidono sullo stato del sistema e che possono essere, quindi, trascurati consentendoci di arrivare ai risultati attesi.

Per vedere meglio quanto affermato facciamo uso di vettori e matrici per rappresentare in maniera tradizionale gli stati e gli operatori in gioco.
Poniamo $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ e $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, allora $|y\rangle = \begin{bmatrix} a \\ b \end{bmatrix}$ e i due operatori di proiezione avranno forma: $M_0 = |0\rangle\langle 0| = \begin{bmatrix} 1 \\ 0 \end{bmatrix}\begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ e $M_1 = |1\rangle\langle 1| = \begin{bmatrix} 0 \\ 1 \end{bmatrix}\begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$.
La probabilità di ottenere $|0\rangle$ sarà $p(0) = \langle y | M_0 | y \rangle = \begin{bmatrix} a & b \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = |a|^2$ che è quanto ci aspettavamo. Infine, lo stato del qubit dopo la misurazione sarà proprio $\frac{M_0|y\rangle}{|a|} = \frac{1}{|a|}\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \frac{a}{|a|}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{a}{|a|}|0\rangle$.

La capacità di un qubit di essere in uno stato di sovrapposizione che non possiamo nemmeno misurare va contro il nostro senso comune: il bit classico è come una moneta che, una volta lanciata, cadrà a terra mostrando inesorabilmente una delle due facce mentre il qubit può essere immaginato come una moneta che, una volta lanciata, cadrà a terra continuando a ruotare su se stessa senza arrestarsi fino a che qualcuno non la

schiacci con una mano bloccandone la rotazione e obbligandola finalmente a mostrare una delle sue facce.

Prima di concludere è utile porsi la domanda: “ma quante informazioni possono essere rappresentate da un qubit?” alla quale già possiamo dare risposta definitiva.

Paradossalmente ci sono un numero infinito di combinazioni lineari della base ortonormale così da permettere, almeno in linea di principio, la rappresentazione in un unico qubit di tutto lo scibile umano.

Ma questa conclusione risulta erronea in virtù del comportamento del qubit in fase di misurazione. Va tenuto presente, infatti, che l'esito della misurazione dello stato di un qubit può essere soltanto $|0\rangle$ oppure $|1\rangle$.
Di più, la misurazione del qubit ne cambia inesorabilmente lo stato riducendo la sovrapposizione in uno dei due specifici stati rappresentati dai vettori della base computazionale così come prescritto dal terzo postulato.

Quindi, dalla misurazione di un qubit, è possibile ottenere la stessa quantità di informazione rappresentabile con un bit classico.

§.30 Registri quantistici

Chiameremo registro quantistico un sistema di n qubit composti secondo le regole dettate dal quarto postulato.
Esso sarà dunque rappresentato da uno spazio di Hilbert di dimensione $q = 2^n$.

Ricordiamo che come base di tale spazio vettoriale possiamo prendere una qualsiasi q-upla di vettori indipendenti.

La base computazionale sarà pertanto costituita da una qualsiasi di queste q-uple a patto che i vettori siano tutti ortogonali tra loro.

Alla base computazionale verranno associati tutti i possibili stati che il nostro registro costituito da n qubit può assumere dopo la misurazione (tanti quanti ne può assumere un registro di bit) dunque la dimensione della base computazionale, ripetiamo, sarà sempre $q = 2^n$.

Le seguenti notazioni del vettore di stato di un registro costituito da n qubit sono equivalenti: $|x_1\rangle \otimes \cdots \otimes |x_n\rangle$, $|x_1\rangle \cdots |x_n\rangle$, $|x_1, \ldots, x_n\rangle$, $|x_1 \cdots x_n\rangle$.

Spesso, inoltre, i $q = 2^n$ vettori della base computazionale verranno indicati in notazione decimale ovvero con i numeri interi $|j\rangle$ $j = 0,1,\ldots,q-1$ ai quali i vettori

della base risultano naturalmente abbinati. Si osservi che se $j = \sum_{i=0}^{q-1} x_i 2^i$ allora $x_1,\ldots,x_n$ è proprio l'espansione binaria di j ed ecco perchè risulta naturale (e pratico) chiamare più semplicemente $|j\rangle$ il vettore $|x_1,\ldots,x_n\rangle$.

§.31 Porte logiche quantistiche

Una porta logica altro non è che un operatore unitario dello spazio vettoriale degli stati del registro e che trasforma dunque uno stato del registro in un altro.

Abbiamo già visto che un qualunque stato $|j\rangle$ può essere espresso come combinazione lineare dei vettori della base computazionale

(93) $$|j\rangle = \sum_{i=0}^{q-1} j_i |i\rangle$$

Ricordiamo che i coefficienti della combinazione lineare devono essere normalizzati ovvero sottostare alla condizione:

(94) $$\sum_{i=0}^{q-1} |j_i|^2 = 1$$

e che di siffatti coefficienti ne abbiamo veramente tantissimi perchè il campo sottostante al nostro spazio vettoriale che contiene tutti i possibili stati del registro, campo dal quale vengono attinti i coefficienti dei vettori, è addirittura $\mathbb{C}$.

Ad un generico stato $|j\rangle$ del registro possiamo quindi associare un vettore $\left[j_0,\ldots,j_{q-1}\right]^T$ e ad una qualsiasi porta (che per il primo postulato dovrà essere inesorabilmente una trasformazione unitaria) possiamo associare un operatore unitario rappresentabile con una matrice unitaria $q\times q$.

Ora limitiamoci al caso di un registro costituito da un singolo qubit.
La base computazionale sarà costituita da due stati distinti che chiameremo $|0\rangle$ e $|1\rangle$.
Una qualsiasi porta unitaria U sarà pertanto rappresentata dalla matrice $U = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ i cui elementi dovranno soddisfare la condizione necessaria a garantire l'unitarietà dell'operatore ad essa associato.

In virtù di queste equivalenze potremo esprimere l'azione della porta sul registro sia con la notazione di Dirac: $|y\rangle = U|j\rangle$ che con la classica matriciale: $\begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} j_0 \\ j_1 \end{bmatrix}$.

La sua azione sulla base computazionale sarà invece rappresentata così:

(95)
$$U:\begin{cases}|0\rangle \to a|0\rangle + b|1\rangle \\ |1\rangle \to c|0\rangle + d|1\rangle\end{cases}$$

Consideriamo, come esempio, la porta NOT.
Immaginiamo un processo fisico che porti lo stato $|0\rangle$ nello stato $|1\rangle$ e viceversa. Tale processo è sicuramente un buon candidato a rappresentare la porta NOT quantistica, ma senza ulteriori informazioni sulle proprietà di un siffatto processo non siamo in grado di conoscere gli effetti su un qubit in stato di sovrapposizione.

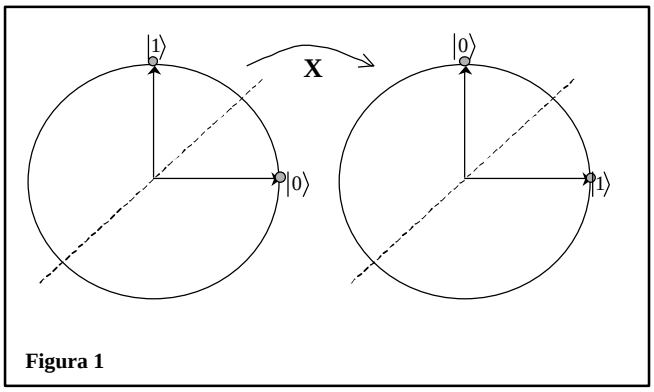

Figura 1

Tuttavia la porta logica NOT agisce linearmente e quindi, noto il suo comportamento sui vettori della base, è nota la sua azione sull'intero spazio. Allora l'operatore lineare unitario NOT può essere rappresentato dalla matrice $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ che è una riflessione rispetto alla bisettrice principale e la sua azione sullo stato del qubit sarà interamente descritta da $X|y\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} b \\ a \end{bmatrix}$ (vedi fig.1).

Nel caso quantistico la porta X non esaurisce l'insieme degli operatori non banali che agiscono su un singolo qubit.
Altre tre porte rilevanti sono $Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$, $Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ e $H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$.

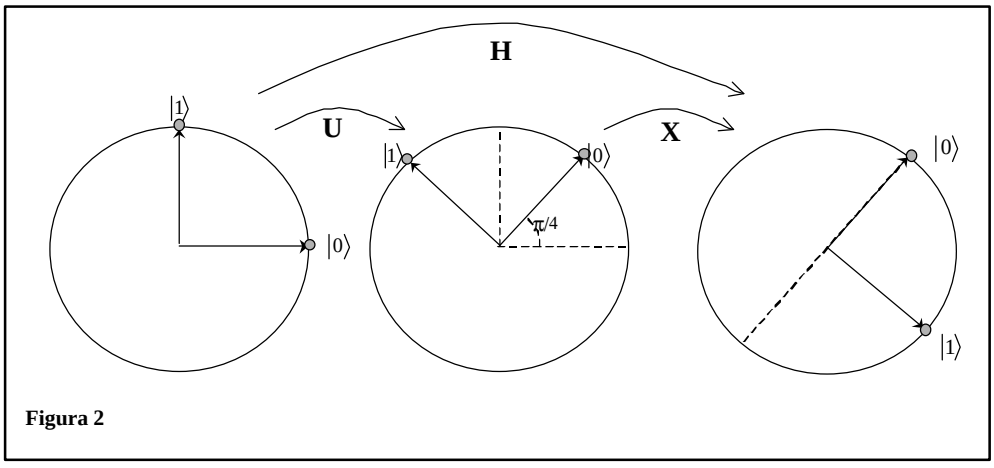

Figura 2

L'operatore H (fig.2), noto come **porta di Hadamard** o come $\sqrt{\text{NOT}}$, può essere visto come il prodotto dell'operatore X per una rotazione U di $\frac{p}{4}$.

Le tre matrici X, Y e Z, unitamente all'identità I, sono conosciute come *matrici di Pauli* e giocano un ruolo importante nello studio della computazione quantistica.

L'operatore H, invece, è una delle porte logiche quantistiche più utili per predisporre il qubit in uno stato di sovrapposizione.

Abbiamo quindi cominciato a conoscere alcune delle porte logiche agenti su un singolo qubit che i postulati della meccanica quantistica ci consentono di costruire.

Passiamo ora al caso di due qubit. Ripetiamo che, così come lo stato di un qubit è rappresentabile come vettore unitario di uno spazio di Hilbert di dimensione due, lo stato di un insieme di n qubit è rappresentabile come vettore unitario di uno spazio di Hilbert che è il prodotto tensore dei singoli spazi ed avrà, pertanto, dimensione 2^n.

Nel caso in esame, in cui $n=2$, la base computazionale sarà costituita dal prodotto tensore delle due basi cioè dai quattro stati ortogonali $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ la cui notazione vettoriale sarà, rispettivamente:

(96)
$$\begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\end{bmatrix}.$$

Gli operatori agiranno su uno spazio vettoriale di dimensione 4 e saranno, quindi, rappresentati da matrici del medesimo ordine.

Un esempio di porta logica multipla (cioè agente su più qubit) è la porta *controlled*-NOT o CNOT. Questa porta ammette due qubit come input, denominati qubit di controllo il primo e qubit target il secondo.

L'azione della porta consiste nel cambiare o lasciare inalterato lo stato del qubit target in funzione dello stato del qubit di controllo. La matrice rappresentativa ha la forma:

(97)
$$U_{CN} = \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{bmatrix}$$

Un altro modo di descrivere l'azione della porta CNOT è la generalizzazione della porta classica XOR, infatti $U_{CN}:|A,B\rangle \to |A,B\oplus A\rangle$ dove il simbolo $\oplus$ rappresenta la somma modulo 2 che è esattamente l'azione della porta classica XOR.

§.32 Porta di Hadamard

Abbiamo visto che l'azione di una porta di Hadamard su un singolo qubit è:

(98)
$$H:\begin{cases}|0\rangle \to \dfrac{|0\rangle+|1\rangle}{\sqrt{2}}\\ |1\rangle \to \dfrac{|0\rangle-|1\rangle}{\sqrt{2}}\end{cases}$$

da cui ricaviamo immediatamente la sua notazione matriciale (quella utile per i conti):

(99)
$$H=\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$$

Questa procedura può essere generalizzata a funzioni su un arbitrario numero di qubit usando la *trasformata di Hadamard* costituita da n porte di Hadamard applicate in parallelo ad n qubit. Ad esempio, due qubit inizialmente nello stato $|0\rangle$ vengono trasformati dalla porta $H\otimes H$ nello stato:

(100)
$$\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)=\frac{|00\rangle+|01\rangle+|10\rangle+|11\rangle}{2}.$$

Denoteremo l'azione parallela di n porte di Hadamard con $H^{\otimes n}$ dove il simbolo $\otimes$ rappresenterà il prodotto tensoriale tra le singole porte.

In generale la porta di Hadamard applicata allo stato $|0\rangle$ della base computazionale (corrispondente al registro con tutti i qubit impostati a 0) produrrà lo stato di sovrapposizione equilibrata di tutti gli stati della base computazionale:

(101)
$$|j\rangle=\frac{1}{\sqrt{q}}\sum_{i=0}^{q-1}|i\rangle$$

Non solo, ma la sua azione è estremamente efficiente in quanto produce la sovrapposizione di 2^n stati usando soltanto n porte.

Questo però non basta per individuare la matrice di trasformazione quindi ci serve anche l'azione su tutti gli altri vettori della base computazionale:

(102) $$H^{\otimes n}:|j\rangle \to H^{\otimes n}|j\rangle = \frac{1}{\sqrt{q}}\sum_{i=0}^{q-1}(-1)^{i\cdot j}|i\rangle \quad \forall j=0,\ldots,q-1$$

dove con $i\cdot j$ si è indicato il prodotto interno bit per bit modulo 2 ovvero, detto $w_r^j = 0,1$ lo stato dell'r-esimo bit dell'espansione binaria $w^j = \left(w_n^j,\ldots,w_1^j\right)$ di j,

(103) $$i\cdot j = \sum_{r=1}^{n} w_r^j w_r^i \bmod 2$$

Tanto per prendere dimestichezza con la notazione verifichiamo le formule precedenti cercando di riottenere l'azione della porta di Hadamard nei casi $n=1$ e $n=2$:

(104) $$H:\begin{cases}|0\rangle \to \frac{1}{\sqrt{2}}\left[(-1)^{0\cdot 0}|0\rangle + (-1)^{0\cdot 1}|1\rangle\right]\\ |1\rangle \to \frac{1}{\sqrt{2}}\left[(-1)^{1\cdot 0}|0\rangle + (-1)^{1\cdot 1}|1\rangle\right]\end{cases}$$

e poichè nel nostro caso ad un solo qubit i vettori i e j hanno identica rappresentazione decimale e binaria, il risultato è immediato.

Proviamo quindi con due qubit:

(105) $$H^{\otimes 2}:\begin{cases}|0\rangle \to \frac{1}{2}\left[(-1)^{0\cdot 0}|0\rangle + (-1)^{0\cdot 1}|1\rangle + (-1)^{0\cdot 2}|2\rangle + (-1)^{0\cdot 3}|3\rangle\right]\\ |1\rangle \to \frac{1}{2}\left[(-1)^{1\cdot 0}|0\rangle + (-1)^{1\cdot 1}|1\rangle + (-1)^{1\cdot 2}|2\rangle + (-1)^{1\cdot 3}|3\rangle\right]\\ |2\rangle \to \frac{1}{2}\left[(-1)^{2\cdot 0}|0\rangle + (-1)^{2\cdot 1}|1\rangle + (-1)^{2\cdot 2}|2\rangle + (-1)^{2\cdot 3}|3\rangle\right]\\ |3\rangle \to \frac{1}{2}\left[(-1)^{3\cdot 0}|0\rangle + (-1)^{3\cdot 1}|1\rangle + (-1)^{3\cdot 2}|2\rangle + (-1)^{3\cdot 3}|3\rangle\right]\end{cases}$$

Ricordando che l'espansione binaria dei numeri ottenibili con due bit è:

(106) $$0\equiv(0,0),\ 1\equiv(0,1),\ 2\equiv(1,0) \text{ e } 3\equiv(1,1)$$

e indicando con $\times$ l'ordinario prodotto tra scalari, ricaviamo:

(107)
$$
\begin{aligned}
&i\cdot 0 = 0\cdot i = (0,0)\cdot\left(w_0^i, w_1^i\right) = \left(0\times w_0^i + 0\times w_1^i\right)\bmod 2 = 0\ \forall i\\
&1\cdot 1 = (0,1)\cdot(0,1) = (0\times 0 + 1\times 1)\bmod 2 = 1\\
&2\cdot 1 = 1\cdot 2 = (0,1)\cdot(1,0) = (0\times 1 + 1\times 0)\bmod 2 = 0\\
&2\cdot 2 = (1,0)\cdot(1,0) = (1\times 1 + 0\times 0)\bmod 2 = 1\\
&3\cdot 1 = 3\cdot 2 = 1\cdot 3 = 2\cdot 3 = (1\times 1 + 0\times 1)\bmod 2 = 1\\
&3\cdot 3 = (1\times 1 + 1\times 1)\bmod 2 = 0
\end{aligned}
$$

arrivando infine alla matrice:

(108)
$$
\frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1 \end{bmatrix}
$$

E' interessante osservare la relazione $H^{\otimes 2} = \frac{1}{\sqrt{2}}\left[\begin{array}{c|c} H & H \\ \hline H & -H \end{array}\right]$ ed è immediato verificare che vale la relazione più generale come conseguenza della definizione di prodotto tensore:

(109)
$$
H^{\otimes(n+1)} = \frac{1}{\sqrt{2}}\left[\begin{array}{c|c} H^{\otimes n} & H^{\otimes n} \\ \hline H^{\otimes n} & -H^{\otimes n} \end{array}\right] = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1\\ 1 & -1 \end{bmatrix}\otimes H^{\otimes n} = H\otimes H^{\otimes n}
$$

§.33 Circuiti quantistici

Analogamente a quanto accade con i circuiti classici che, nella loro essenza, sono costituiti da fili e porte logiche, un circuito quantistico è costituito da fili che trasportano le informazioni lungo i circuiti e da porte logiche quantistiche che le manipolano.

Abbiamo già incontrato alcuni semplici circuiti logici costituiti da una sola porta e, in virtù della loro semplicità, è stato possibile visualizzare direttamente l'azione sui vettori della base computazionale.

Nel seguito verranno presentati circuiti logici di maggiore complessità per i quali la rappresentazione matriciale diventa compito tedioso (già solo con tre qubit gli operatori sono rappresentati da matrici di ordine 8) e, inoltre, concentrando l'attenzione sugli effetti della singola porta, fa perdere di vista il quadro complessivo di funzionamento del circuito.

Introduciamo allora una modalità di visualizzazione che ricorda moltissimo quella utilizzata per rappresentare i circuiti elettrici.

In essa ogni operatore sarà rappresentato da un simbolo la cui azione potrà essere esplicitata riportando nel diagramma gli stati dei qubit di input e di output.

I simboli saranno collegati da linee, che rappresentano i fili, indicando così il fatto che l'output di una porta logica è input per la porta logica successiva (ovvero l'operatore corrispondente alla seconda porta agisce sullo spazio immagine dell'operatore corrispondente alla prima porta).

E' da notare che, nel caso quantistico, i fili non corrispondono necessariamente a fili fisici, come succede nel caso dei circuiti elettrici, ma possono corrispondere al fatto che una particella fisica, come un fotone, si muove da un posto ad un altro attraverso lo spazio.

Il flusso logico nel circuito scorre da sinistra verso destra e convenzionalmente si assume che lo stato iniziale sia uno di quelli appartenenti alla base computazionale, solitamente quello in cui tutti i qubit di lavoro sono posti a $|0\rangle$.

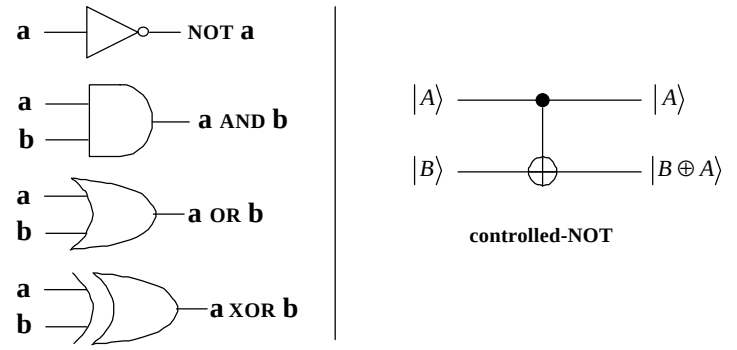

Figura 3

I fili che congiungono verticalmente due simboli rappresentano i condizionamenti e indicano che lo stato di un qubit incide sul risultato dell'operazione effettuata sull'altro qubit.

I due simboli che incontreremo spesso sono il cerchio pieno ($\bullet$), utilizzato per rappresentare l'identità in presenza di condizionamenti, e la somma modulo 2 ($\oplus$). In fig.3 diamo un esempio dei simboli utilizzati per rappresentare le principali porte logiche classiche (a sinistra in figura) e il prototipo di porta multipla, cioè la porta CNOT, (a destra).

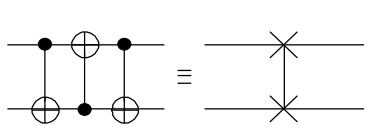

Figura 4

Per cominciare a prendere confidenza con questa simbologia analizziamo un semplice circuito che svolge l'utile compito di scambiare tra loro gli stati di due qubit. In fig.4 è rappresentata la sua notazione simbolica (a destra) e la sua decomposizione nel prodotto di tre porte CNOT (a sinistra).

La sua azione ha i seguenti effetti sullo stato $|a,b\rangle$ della base computazionale:

$$|a,b\rangle \to |a, a\oplus b\rangle \to |a\oplus(a\oplus b), a\oplus b\rangle = |b, a\oplus b\rangle \to |b,(a\oplus b)\oplus b\rangle = |b,a\rangle .$$

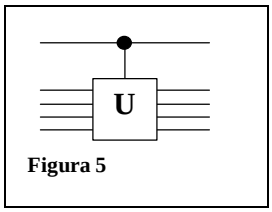

Figura 5

Una immediata e naturale generalizzazione della porta controlled-NOT è la porta controlled-U, illustrata in fig.5, in cui supponiamo che U sia un qualsiasi operatore unitario agente su n qubit. Tale porta ha un singolo qubit di controllo e n qubit target. Se il qubit di controllo è impostato a $|0\rangle$ allora i qubit target non vengono manipolati. Se, invece, il qubit di controllo è impostato a $|1\rangle$ allora l'operatore U viene applicato ai qubit target.

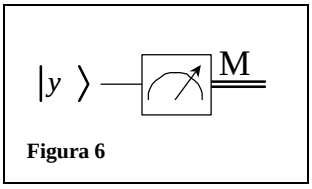

Figura 6

Infine, un'altra importante operazione è la misurazione, che rappresenteremo con un "contatore", come illustrato in fig.6. Abbiamo già visto che l'operazione di misurazione converte lo stato $|y\rangle = a|0\rangle + b|1\rangle$ di un singolo qubit nella versione probabilistica di un bit classico che avrà stato (0) con probabilità $|a|^2$ e stato (1) con probabilità $|b|^2$. Tale fatto verrà indicato usando un doppio filo uscente dal simbolo di misurazione.

§.34 Rappresentazione dell'ambiente quantistico

Innanzi tutto scegliamo di rappresentare un qubit per mezzo delle sue due coordinate complesse, normalizzate, rispetto alla base computazionale $|0\rangle$ e $|1\rangle$.

Quindi, se $|x\rangle = x_0|0\rangle + x_1|1\rangle$, lo stato del qubit sarà rappresentato dal vettore bidimensionale a coordinate complesse:

(110) $$\overline{x} \equiv \begin{bmatrix} a_0 + ib_0 \\ a_1 + ib_1 \end{bmatrix}$$

Conseguentemente, un registro costituito da n qubit sarà rappresentato da un vettore di $q = 2^n$ coordinate complesse normalizzate.

Ovviamente sarebbe meno dispendioso "immagazzinare" lo stato dei singoli qubit (ciò richiederebbe solo $2n$ coordinate complesse!) e ricostruire lo stato del registro, come prodotto tensore degli stati dei singoli qubit componenti, solo all'occorrenza.

Ma ricordando l'esistenza di stati entangled, ovvero di stati del registro che non possono essere scomposti nel prodotto tensore di stati dei singoli qubit, desumiamo che tale semplificazione ci è vietata.

Non ci resta, dunque, che la seguente rappresentazione:

(111) $$|x\rangle = x_0|0\rangle + \cdots + x_{q-1}|q-1\rangle \equiv \overline{x} \equiv \begin{bmatrix} a_0 + ib_0 \\ \cdots \\ a_{q-1} + ib_{q-1} \end{bmatrix}$$

Poichè nella realtà non è possibile “impostare a mano” l’istanza di input del nostro registro, ogni algoritmo verrà preceduto da una fase di assegnazione che sarà ovviamente costituita da un operatore unitario.

Generalmente ogni algoritmo prevede come istanza di input uno stato puro (ovvero coincidente con uno dei q vettori della base del registro) quindi si tratterà di definire una famiglia di operatori A_k di assegnazione tali che $A_k|x\rangle = |k\rangle$ con $k = 0,\ldots,q-1$ qualunque sia lo stato $|x\rangle$ del registro al momento dell’assegnazione.

Ogni altro stato del registro, ovvero gli stati di sovrapposizione (detti stati misti) e in particolare gli stati misti entangled potranno comunque essere ottenuti tramite l’applicazione di un operatore di assegnazione costruito ad hoc.

Gli operatori verranno rappresentati per mezzo di matrici a coefficienti complessi.
In accordo col secondo postulato saranno lecite solo le matrici unitarie.

Un operatore unitario agente su un registro di n qubit sarà pertanto rappresentato da una matrice unitaria di dimensione $q = 2^n$ a sua volta costituita da 2^{2n} coefficienti complessi (ovvero da 2^{2n+1} numeri reali).

Poichè, in ultima analisi, un qualsiasi circuito quantistico è una composizione di porte logiche quantistiche agenti su alcuni o tutti i qubit del registro, la traduzione di un algoritmo (rappresentato appunto da un circuito) in codice consisterà semplicemente nella scrittura di matrici elementari e nella composizioni tramite le ordinarie operazioni tra matrici.

L’esecuzione dell’algoritmo consisterà dunque nell’applicazione dell’operatore unitario al vettore rappresntativo del registro.

Tale applicazione sarà preceduta dalla fase di assegnazione dello stato di input al registro e seguita dalla fase di misurazione dello stato di output.

Alcuni operatori elementari, per la loro semplicità, verranno definiti dichiarando direttamente i coefficienti della matrice rappresentativa.

Sostanzialmente stiamo parlando dei seguenti operatori agenti su un singolo qubit:

(112) Identità: $I \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

(113) Hadamard: $H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

(114) Pauli-X (NOT): $X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

(115) $$\text{Pauli-Y: } Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

(116) $$\text{Pauli-Z: } Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

(117) $$R_k \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix}$$

Di quest'ultimo operatore daremo spiegazioni in seguito.

Per altri operatori elementari risulterà invece più semplice lasciare la costruzione al programma stesso e a tale scopo faremo uso sostanzialmente del prodotto tensore e del prodotto esterno.

Il prodotto tensore di due matrici quadrate $A \equiv \{a_{ij}\}$ e $B \equiv \{b_{hk}\}$ rispettivamente di ordine n e m è la matrice $C = A \otimes B \equiv \{c_{rs}\}$ i cui coefficienti sono individuati da:

(118) $$c_{rs} = a_{ij} b_{hk} \text{ con } r = i \cdot m + h \text{ e } s = j \cdot m + k$$

Il prodotto esterno $|x\rangle\langle y|$ tra due vettori $|x\rangle$ e $|y\rangle$ rispettivamente di coordinate $\{x_h\}$ e $\{y_k\}$ e dimensioni n e m è una matrice $A = |x\rangle\langle y| \equiv \{a_{ih}\}$ di dimensione $(n \times m)$ i cui coefficienti sono determinati da:

(119) $$a_{ih} = x_i y_h .$$

Noi useremo tale prodotto solo nel caso $n = m$.

Ora siamo in grado di costruire alcuni operatori più complessi.

Ad esempio se abbiamo un registro di 3 qubit e dobbiamo mettere in stato di sovrapposizione equilibrata il primo e il terzo qubit dovremo passare solo il primo e il terzo qubit per una porta di Hadamard $H^{\otimes 2}$.

La matrice A che fa questa cosa la costruiamo così:

(120) $$A = H \otimes I \otimes H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Un esempio importante di operatore ricavato tramite composizione, già stato visto nei precedenti paragrafi, è la porta multipla di Hadamard:

(121) $$H^{\otimes n} = H \otimes \cdots \otimes H$$

Un altro esempio fondamentale è la costruzione della porta controlled-NOT. La sua composizione discende immediatamente dalla descrizione della sua azione: se il qubit di controllo è nello stato $|0\rangle$ il qubit target rimane inalterato, viceversa se il qubit di controllo è nello stato $|1\rangle$ allora sul qubit target viene fatta agire una porta NOT:

(122) $$cX = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

La sua generalizzazione controlled-U che esegue l'azione U su un registro target costituito da n qubit in funzione dello stato del qubit di controllo è immediata:

(123) $$|0\rangle\langle 0| \otimes I^{\otimes n} + |1\rangle\langle 1| \otimes U$$

Come esempio costruiamo la porta controlled-SWAP chiamata anche porta di Fredkin.

La porta SWAP che scambia tra loro due qubit può essere realizzata applicando in sequenza una porta cNOT, una porta NOTc e una porta cNOT.

Costruiamo quindi la porta NOTc:

(124) $$Xc = I \otimes |0\rangle\langle 0| + X \otimes |1\rangle\langle 1| = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Ed ora costruiamo la porta SWAP:

(125) $$cX \cdot Xc \cdot cX = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Infine la porta di Fredkin potrà essere ricavata eseguendo esplicitamente i seguenti calcoli:

(126) $$|0\rangle\langle 0| \otimes I^{\otimes 2} + |1\rangle\langle 1| \otimes SWAP$$

Come ultimo esempio costruiamo una porta agente su tre qubit detta porta di Toffoli.

La sua azione è la seguente.

Se i primi due qubit sono nello stato $|11\rangle$ il terzo qubit cambia di stato altrimenti resta inalterato e ciò è sufficiente per scrivere la matrice:

(127) $$\sum_{j=0}^{2} |j\rangle\langle j| \otimes I + |3\rangle\langle 3| \otimes X = \left[\begin{array}{c|c|c|c} I & 0 & 0 & 0 \\ \hline 0 & I & 0 & 0 \\ \hline 0 & 0 & I & 0 \\ \hline 0 & 0 & 0 & X \end{array}\right]$$