Alma Mater Studiorum · Università di Bologna

Scuola di Scienze
Corso di Laurea in Fisica

Algoritmi quantistici e classi di complessitá

Relatore:
Prof. Elisa Ercolessi

Presentata da:
Lorenzo Gatti

Sessione II
Anno Accademico 2014/2015

Indice

Abstract:

Solitamente il concetto di difficoltá é piuttosto soggettivo, ma per un matematico questa parola ha un significato diverso: anche con l'aiuto dei piú potenti computer puó essere impossibile trovare la soluzione di un sudoku, risolvere l'enigma del commesso viaggiatore o scomporre un numero nei suoi fattori primi; in questo senso le classi di complessitá computazionale quantificano il concetto di difficoltá secondo le leggi dell'informatica classica. Una macchina quantistica, peró, non segue le leggi classiche e costituisce un nuovo punto di vista in una frontiera della ricerca legata alla risoluzione dei celebri problemi del millennio: gli algoritmi quantistici implementano le proprietá straordinarie e misteriose della teoria dei quanti che, quando applicate lucidamente, danno luogo a risultati sorprendenti.

Introduzione

Il mio elaborato vuole delineare il percorso storico che ha portato alla teorizzazione dei calcolatori quantistici a partire da concetti di base dell'informatica classica, come ad esempio quello di algoritmo o di computazione. Si puó dire che lo sviluppo della meccanica quantistica abbia determinato l'inizio dell'era dell'informazione, considerando che il componente hardware principale dei computer odierni, il transistor, é un diretto risultato dell'applicazione della teoria quantistica agli elettroni nei solidi; la sinergia fra meccanica dei quanti e informatica rimane solida ancora oggi e offre diversi spunti per sviluppi tecnologici rivoluzionari. R. Feynman avanzó per primo, nel 1985, l'idea di utilizzare sistemi quantistici per trasmettere informazione[1]; da allora, attraverso i contributi di vari fisici ed informatici, sono stati fatti notevoli progressi e nel 2013 é stato presentato il primo computer quantistico, D-Wave.

Ma cosa rende il computer quantistico cosí interessante? La teoria dell'informazione classica suddivide i problemi risolvibili da un calcolatore a seconda della loro complessitá, ovvero del tempo impiegato dal calcolatore a risolverli in funzione della lunghezza dell'input; apparentemente esistono problemi che risultano irrisolvibili persino da un computer quando le dimensioni dei parametri iniziali diventano rilevanti. Queste leggi, peró, non valgono per un calcolatore quantistico che ha la capacitá di effettuare piú operazioni insieme (parallelismo quantistico); da tutto ció é nata un'improbabile alleanza fra fisici ed informatici con l'obiettivo comune di sviluppare il piú possibile l'idea di Feynman; gli informatici per correggere l'unica vera spina della teoria di Turing, i fisici per cercare di comprendere un po' di piú i misteri della meccanica quantistica. Troviamo allora, come risultato di questa cooperazione, una serie di algoritmi quantistici, strutturati in modo da permettere il manifestarsi di fenomeni quantistici quali il principio di sovrapposizione o l'entanglement; solo sfruttando adeguatamente queste proprietá risulta possibile attingere a tutte le potenzialitá del quantum computing.

Nel primo capitolo introdurremo nozioni di base dell'informatica classica, come quella di Macchina di Turing e di computazione, per poi fornire una definizione formale delle

classi di complessitá computazionale; nel secondo capitolo tratteremo i postulati della meccanica quantistica e qualche loro diretta conseguenza, come il principio di sovrapposizione e il teorema no-cloning, in modo da poter descrivere cos'é un qubit e come rappresentarlo. Infine nel terzo capitolo presenteremo qualche algoritmo quantistico, per poter valutare effettivamente, secondo il formalismo definito nel capitolo 1, la potenza di una macchina quantistica rispetto all'analoga classica.

Capitolo 1

Le classi di complessitá

"In natural science, Nature has given us a world and we're just to discover its laws. In computers, we can stuff laws into it and create a world."
– Alan Kay

Le teorie della computazione e dell'informazione quantistica si occupano di studiare e sviluppare le procedure di scambio ed elaborazione di informazione utilizzando sistemi quantomeccanici. Questo rivoluzionario campo di ricerca nasce dall'unione di idee appartenenti a discipline differenti, quali l'informatica, la matematica e la fisica; perció, prima di introdurre i concetti fondamentali per una trattazione quantistica, mi sembra importante soffermarsi sulle strutture di base dell'informatica classica. La teoria classica dell'informazione, concepita negli anni '30 e rapidamente sviluppata nei decenni seguenti, é stata in grado di fornire un formalismo solido e una serie di oggetti e di tecniche copiosamente sfruttati anche a livello quantistico; inoltre gli informatici hanno speso gran parte del loro tempo nello sviluppare una teoria che permettesse di capire e di quantificare le risorse richieste da un computer classico per risolvere un determinato problema o, piú generalmente, un'intera classe di problemi. Questi studi potranno essere poi facilmente usati per istituire un confronto con il computer quantistico e visualizzarne in modo piú concreto quelli che sono i suoi limiti e le sue potenzialitá.

1.1 La macchina di Turing e il concetto di algoritmo

Intuitivamente un algoritmo puó essere interpretato come una sequenza di istruzioni per risolvere un determinato problema; ovviamente, come in tutti i casi in cui risulta semplice fornire una quantificazione sommaria di un concetto, la sua definizione formale in linguaggio matematico é invece piuttosto articolata e di difficile comprensione. La prima definizione rigorosa di algoritmo fa riferimento al concetto di calcolabilitá e risale al 1936, anno in cui Alan Turing pubblica un articolo intitolato "On computable numbers with an application to the entscheidungsproblem", in cui introduce uno strumento teorico rivoluzionario come la macchina di Turing (MdT), fondando di fatto l'informatica classica[2].

Una MdT é composta da quattro elementi distinti: un programma, un nastro unidimensionale di lunghezza infinita in entrambe le direzioni, una testina di lettura/scrittura e un processore. Per descriverne il funzionamento occorre innanzitutto specificare che il nastro unidimensionale da cui la macchina puó leggere e scrivere é formato da celle, ognuna contenente un simbolo appartenente ad un determinato alfabeto Σ (Se non diversamente indicato si pone $\Sigma = \{0, 1, b, \triangleright\}$, questi ultimi due simboli rappresentano il carattere 'blank', vuoto e quello di inizio nastro.). La testina di lettura/scrittura é in grado di puntare un simbolo sul nastro, in modo da poterne leggere il valore o sovrascriverlo con un altro simbolo del medesimo alfabeto. Il processore contiene un insieme finito di stati in cui la macchina si puó trovare durante l'esecuzione di un programma; piú precisamente in ogni istante di tempo la macchina si trova in uno stato ben definito ed esegue una linea del programma che ha in memoria; quest'azione porta la macchina a modificare il suo stato a seconda del simbolo letto dalla testina sul nastro, generando diverse possibili configurazioni per diversi input iniziali.

Il funzionamento di una MdT é definito da un programma, ovvero da una sequenza finita di istruzioni che permettono al sistema di eseguire tre diverse operazioni:

1. cambia lo stato del processore da s a s$\star$;
2. sostituisci il simbolo a sul nastro puntato dalla testina di lettura/scrittura con un simbolo a$\star$ dell'alfabeto utilizzato;
3. sposta la testina di lettura/scrittura a destra o a sinistra di una cella sul nastro.

Piú sinteticamente, ogni istruzione di un programma é una mappa:

$$(\text{s, a}) \rightarrow (\text{s}\star, \text{a}\star, \text{d})$$

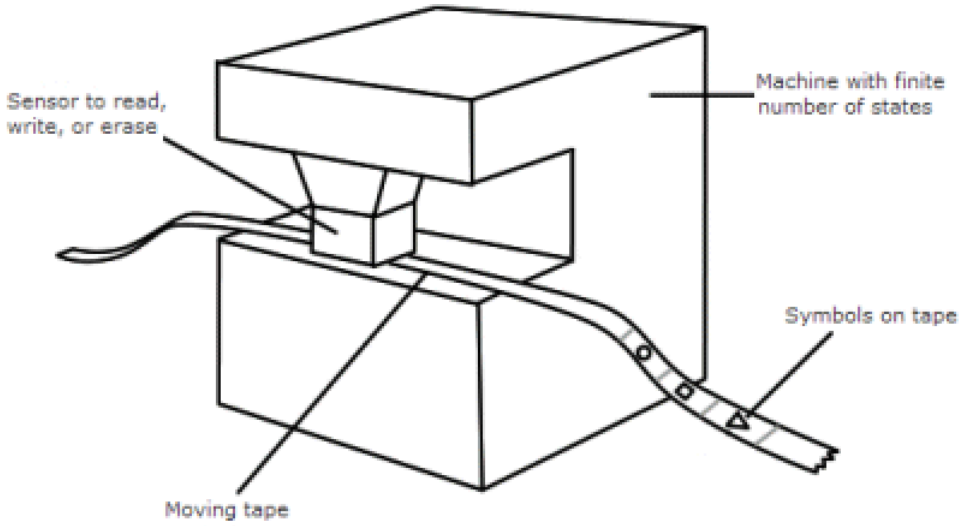

Figura 1.1: Rappresentazione di una macchina di Turing

Ad esempio non é difficile convincersi che le seguenti 7 linee di programma realizzano la funzione costante f(x) = 1, indipendentemente dai valori in input sul nastro (che vengono tutti cancellati e rimpiazzati con caratteri blank 'b'.

$$1:\ \langle q_s, \triangleright, q_1, \triangleright, +1\rangle$$

$$2:\ \langle q_1, 0, q_1, b, +1\rangle$$

$$3:\ \langle q_1, 1, q_1, b, +1\rangle$$

$$4:\ \langle q_1, b, q_2, b, -1\rangle$$

$$5:\ \langle q_2, b, q_2, b, -1\rangle$$

$$6:\ \langle q_2, \triangleright, q_3, \triangleright, +1\rangle$$

$$7:\ \langle q_3, b, q_h, 1, 0\rangle$$

Dove gli stati q_s e q_h sono lo stato di avvio e lo stato di arresto della MdT, mentre il processo che porta da q_s a q_h é detto computazione.

1.2 La Tesi di Church-Turing e il problema della fermata

Dopo aver definito cos'é una MdT possiamo utilizzare questo concetto per derivare una nozione di calcolabilitá; tale risultato é noto sotto il nome di Tesi di Church-Turing, in onore ai due matematici che la ricavarono indipendentemente e con metodi totalmente diversi nel 1936.

La classe delle funzioni calcolabili coincide con quelle calcolabili da una macchina di Turing (ovvero una funzione f é calcolabile quando una MdT con input x e stato interno q_s restituisce dopo un certo numero di passi f(x) in output quando il suo stato é q_h).

In questo modo abbiamo ottenuto che ogni funzione rappresentabile tramite il formalismo delle MdT risulta calcolabile e viceversa; sono funzioni calcolabili ad esempio la somma, la moltiplicazione e il MCD di due numeri. Tuttavia la cosa piú interessante che una MdT é in grado di fare é simulare un'altra MdT! Lo stesso Turing si accorse di questa proprietá e nel suo articolo definí una macchina universale (UTM) che, data una determinata macchina M, prende in input un numero discriminatore della macchina T_M, una stringa x e restituisce come risultato M(x); in effetti é proprio questo il principio su cui si basano i moderni computer programmabili e le architetture hardware-software. Sembrerebbe quindi che non ci sia un limite alle operazioni che una MdT é in grado di compiere e che la potenza di calcolo di questi 'arnesi' sia in qualche modo universale.

Lo stesso Turing si occupó di definire anche quelli che sono i limiti delle macchine da lui create e se ne serví per risolvere il problema della decidibilitá formulato dal matematico David Hilbert qualche anno prima: Hilbert riteneva che ogni enunciato formale potesse essere dimostrato vero o falso all'interno di un sistema coerente ed era cosí interessato alla soluzione di questo problema da inserirlo in una lista di problemi per il XX secolo pubblicata insieme ad Ackermann nel 1928. Solo dopo qualche anno Turing si occupó di usare le macchine da lui inventate per dare una soluzione negativa a questo problema.

Supponiamo l'ipotesi di Hilbert corretta, allora sicuramente saremo in grado di avere una UTM P in grado di determinare se una data MdT M(x) termina o meno una computazione con input x; in particolare poniamo P(M(x)) = 1 se M(x) termina e P(M(x)) = 0 se M(x) non si ferma. Allora é possibile creare una nuova MdT Q(M(x)) a partire da P col seguente pseudocodice:

Q(M(x)): While (P(M(x)) Loop Forever;

Scrivendo Q in questo modo si ottiene che Q(M(x)) entra in un ciclo infinito quando M termina la sua computazione con input x, mentre Q termina la computazione quando M non si ferma con l'input x. Fino a questo punto non ci sono problemi, tuttavia se si analizza quello che succede quando si lancia Q(Q(x)) si ottiene che Q termina la computazione se e solo se Q non termina la computazione. Ne consegue che l'unica possibile soluzione logica é che una macchina fatta come P non puó esistere. Il problema della fermata risulta quindi essere indecidibile, in quanto una MdT non é in grado di determinare se un'altra macchina termina o meno la sua computazione; di fatto Turing non si accontentó solamente di aver inventato il computer, ma lo fece anche bloccare per la prima volta..

1.3 Macchine di Turing e classi di complessitá

Una volta inquadrate le proprietá fondamentali di una MdT diventa spontaneo chiedersi che tipo di risorse spaziali e temporali sono richieste da una macchina per effettuare una computazione, in particolare questo costituisce l'ambito di studio della teoria della complessitá computazionale, il cui compito é quello di fissare dei limiti superiori ed inferiori ai passi necessari ad un algoritmo per risolvere un determinato problema. Il modo migliore per definire il comportamento di un algoritmo si basa sulla notazione asintotica (simboli di Landau):

Date due funzioni f,g: $\mathbb{N} \to \mathbb{R}$, allora

$$f(n) = O(g(n)) \iff \exists\ c \neq 0, n_0 \in \mathbb{N} \setminus f(n) \leq cg(n),\ \forall\ n \geq n_0 \tag{1.1}$$

$$f(n) = \Omega(g(n)) \iff \exists c \neq 0, n_0 \in \mathbb{N} \setminus cg(n) \leq f(n),\ \forall\ n \geq n_0 \tag{1.2}$$

$$f(n) = \Theta(g(n)) \iff f(n) = O(g(n)) = \Omega(g(n)) \tag{1.3}$$

Intuitivamente O e Ω forniscono una stima, piú o meno accurata, dei limiti superiore ed inferiore della funzione in analisi;

La maggior parte dei problemi computazionali possono essere formulati come problemi di tipo decisionale, ovvero dato un alfabeto Σ si definisce L un linguaggio su Σ un sottoinsieme di $\Sigma\otimes\ldots\otimes\Sigma$ costituito da tutte le stringhe finite di elementi di Σ; il problema decisionale viene rappresentato da una MdT con lo stato di arresto q_h rimpiazzato dagli stati q_Y e q_N e definendo il linguaggio L in modo che la macchina, dato in input x, termini la computazione nello stato q_Y se x ∈ L o eventualmente in q_N se x ∉ L. Si puó anche costruire una MdT di natura non deterministica che, utilizzando algoritmi con procedure

di natura stocastica, restituisce lo stato q_Y con una certa probabilitá p > 0,5 se x ∈ L (stesso discorso nel caso q_N). A questo punto fra i modelli costruiti quale risulta essere il piú potente? Stranamente, sembra che l'utilizzo di MdT probabilistiche piuttosto che deterministiche non influenzi in modo apprezzabile la risolubilitá di un problema, infatti é stato dimostrato [3] che, se un problema é risolvibile da una MdT deterministica in n passi, una MdT non deterministica é in grado di risolverlo in O(n^k) passi con k > 0.

Per dare poi un'idea della difficoltá di un problema é possibile definire la classe $\boldsymbol{TIME}(f(n))$, costituita da tutti i problemi che ammettono una MdT in grado di determinare se un candidato (input) x appartiene al linguaggio in un tempo O(f(n)) (si ricordi che questo rappresenta il caso peggiore), dove n é la lunghezza di x. L'insieme di tutti i problemi che sono risolvibili da un algoritmo in tempo polinomiale costituisce un'importante classe di complessitá chiamata $\boldsymbol{P}$; piú formalmente $\boldsymbol{P} \equiv \bigcup_k \boldsymbol{TIME}(n^k)$. Risulta molto comodo, nonché elegante, vedere gli elementi di $\boldsymbol{P}$ come problemi che sono semplici da risolvere efficientemente; inoltre, questa nozione di efficienza non dipende dall'utilizzo di algoritmi deterministici o probabilistici. Un problema appartenente a $\boldsymbol{P}$ é, ad esempio, l'ordinamento di un insieme di dati secondo un criterio specifico che, con algoritmi "intelligenti ", richiede Θ(nlogn) operazioni. Non sempre peró risulta semplice trovare un algoritmo in grado di risolvere un problema in un tempo polinomiale o, peggio, tale algoritmo potrebbe semplicemente non esistere! Un facile esempio é fornito dal gioco della torre di Hanoi:

In tre paletti sono infilati dischi di diverso diametro. All'inizio i dischi sono tutti sul primo paletto, ordinati verso l'alto dal piú grande al piccolo e lo scopo del gioco é trasferire tutti i dischi sull'ultimo paletto, muovendo solo un disco alla volta e facendo in modo che un disco piú grande non si trovi mai sopra uno piú piccolo. A prima vista questo sembra un problema carino, ma non difficile e provando a risolvere qualche caso con un basso numero di dischi é possibile poi ricavare un algoritmo in grado di risolvere il problema per il caso generale di un numero N di dischi. Questo algoritmo puó essere scritto nel seguente modo

1. sposta i primi N-1 dischi dal primo al secondo paletto;
2. sposta il disco piú grande dal primo al terzo paletto;
3. sposta tutti i dischi tranne il piú grande dal secondo al terzo paletto.

e, effettivamente, arriva alla soluzione con tre mosse; tuttavia negli step 1 e 3 si richiede che l'algoritmo sappia come risolvere il problema con un disco in meno. Questa particolare proprietá é detta ricorsivitá e potrebbe giá indurre a sospettare che il numero di

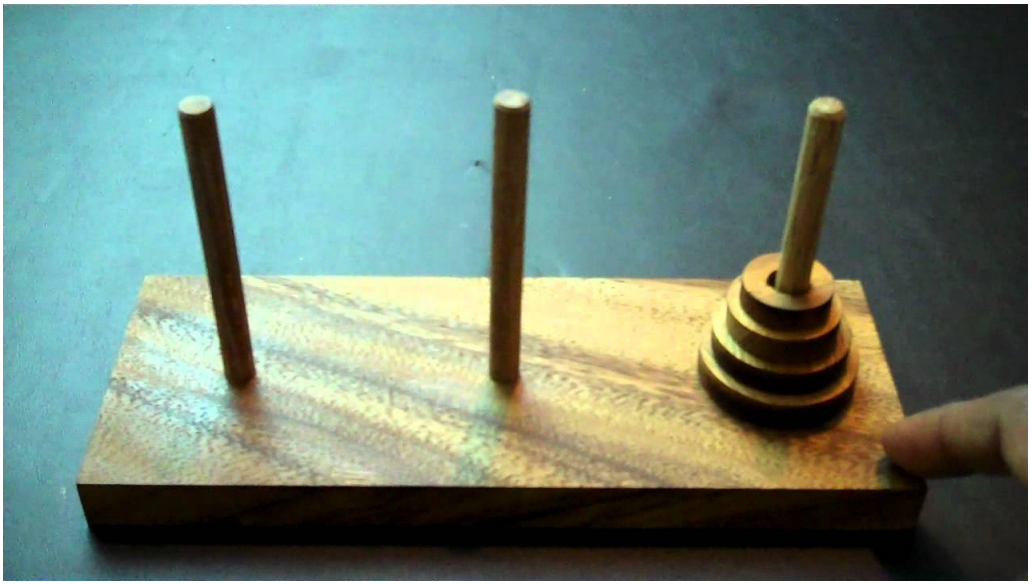

Figura 1.2: Gioco della torre di Hanoi con N=4 dischi.

passi necessari a spostare N dischi non sia polinomiale in N. Difatti adesso dimostreremo che questo numero é $2^N - 1$.

Sia M_N il numero di mosse necessario a spostare N dischi, é banale vedere che $M_1 = 1$. Per applicare l'induzione ora vediamo che $M_{N-1} \rightarrow M_N$; infatti l'algoritmo dato in precedenza permette di scrivere $M_N = 2M_{N-1} + 1 = 2(2^{N-1}$ - 1) + 1 = 2^N - 1.

Il gioco della torre di Hanoi dimostra facilmente l'esistenza di problemi che non sono risolvibili in tempo polinomiale. Sfortunatamente, provare che un singolo problema non puó essere risolto in tempo polinomiale non é altrettanto facile quanto il congetturarlo; ne é un esempio interessante il problema della fattorizzazione: dati due numeri interi positivi m e s, con s < m, m puó essere scritto come prodotto di due fattori p,q con $p \neq 1 < s$? L'interessante proprietá di questo problema é che, data una soluzione, risulta possibile verificare la sua correttezza in tempo polinomiale, tuttavia non sono noti algoritmi che risolvono un'istanza generale del problema in tempo polinomiale. Per questo il problema della fattorizzazione é supposto appartenere ad una classe di complessitá piú grande di $\boldsymbol{P}$, la classe $\boldsymbol{NP}$. In termini di MdT un linguaggio L é in $\boldsymbol{NP}$ quando esiste una MdT M con le seguenti due proprietá:

1. se x ∈ L, allora esiste una stringa 'testimone'w tale che M termina la computazione nello stato q_Y dopo un tempo polinomiale rispetto alla lunghezza di x quando M

nello stato q_S ha come input x-b-w.

2. se x $\notin$ L allora, per tutte le stringhe w che provano a fare da 'testimone', M termina la computazione nello stato q_N dopo un tempo polinomiale rispetto alla lunghezza di x quando M nello stato q_S ha come input x-b-w.

Non é noto, quindi, come risolvere completamente e in modo efficiente un problema in ***NP***, tuttavia nella maggior parte dei casi é semplice verificare la correttezza o meno di una soluzione fornita a parte; non é difficile convincersi che questa classe di complessitá definisce un insieme di problemi "complicati", all'apparenza irrisolvibili.

1.4 I grafi e qualche giochetto con P e NP

Un grafo é una collezione finita di vertici $\{v_1, \ldots, v_n\}$, collegati a coppie da dei lati (v_i, v_j); questi oggetti diventano particolarmente utili nella rappresentazione di certi problemi decisionali, come ad esempio il problema del commesso viaggiatore. Un grafo contiene un ciclo se é possibile collegare i suoi vertici $\{v_1, \ldots, v_n\}$ in modo che i lati siano le coppie (v_1, v_n) e (v_i, v_{i+1}) con $1 \leq i \leq n-1$; in particolare un grafo contiene un ciclo hamiltoniano se contiene un ciclo che comprende ogni suo vertice esattamente una volta sola, ad eccezione del primo.

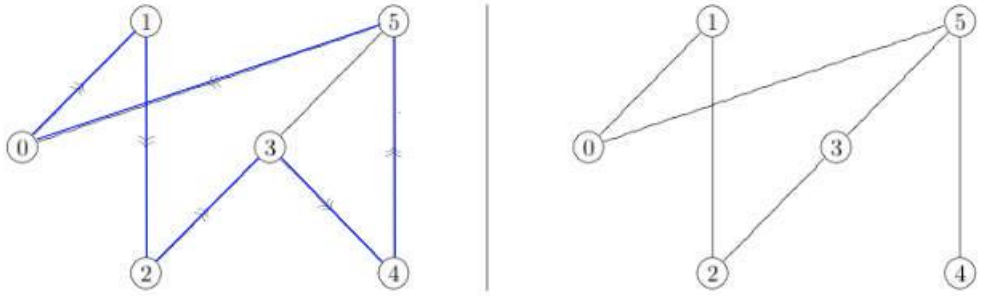

Figura 1.3: Esempi di grafi con 6 vertici, quello a sinistra contiene un ciclo hamiltoniano.

Il problema di determinare se un dato grafo contiene un ciclo hamiltoniano (HC) é un problema decisionale che appartiene alla classe ***NP***; non sono noti algoritmi che lo risolvono efficientemente, anche se la figura 1.3 mostra come, dato un grafo, sia possibile costruire facilmente soluzioni testimone per una particolare istanza.

Un problema simile ad HC, ma dalle proprietá completamente differenti, é il problema del ciclo euleriano (EC). Un grafo contiene un ciclo euleriano se contiene un ciclo che percorre tutti i lati esattamente una volta. A differenza di HC, EC appartiene a $\boldsymbol{P}$ e giá il matematico svizzero Eulero riuscí a risolvere questo problema, noto come problema dei sette ponti di Koenigsberg, nel 1736.

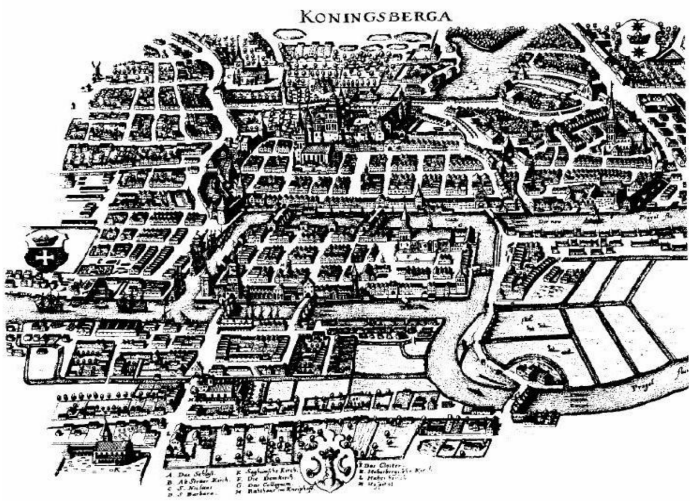

Figura 1.4: La cittá di Koenigsberg (ora Kaliningrad) ai tempi di Eulero.

La cittá di Koenigsberg é attraversata dal fiume Pregel e dai suoi affluenti; questi la dividono in varie zone collegate fra loro per mezzo di sette ponti. Si narra che i suoi abitanti la domenica passeggiassero per la cittá cercando di trovare un modo di partire da un punto e ritornarci dopo aver percorso tutti e sette i ponti non piú di una volta, ovvero un ciclo euleriano. Infatti fu proprio Eulero ad introdurre la teoria dei grafi per risolvere questo curioso problema tramite un teorema che porta il suo nome.

Il teorema di Eulero afferma che un grafo connesso contiene un ciclo euleriano se e solo se ogni vertice é collegato agli altri da un numero pari di lati; nel caso di Koenigsberg, che puó essere schematizzata come nel grafo di figura 1.5, si ha addirittura che nessuno dei vertici ha questa proprietá, per cui un tale cammino non esiste.

Lo studio dei grafi ha permesso una migliore comprensione delle classi di complessitá computazionale e lo sviluppo di strutture di vitale importanza per la teoria degli algoritmi; ciononostante rimane ancora aperta una questione, cosí importante per le sue applicazioni pratiche e tecnologiche da meritarsi un posto fra i problemi del millennio, secondo la lista stilata dal Clay Mathematics Institute. Precedentemente abbiamo in-

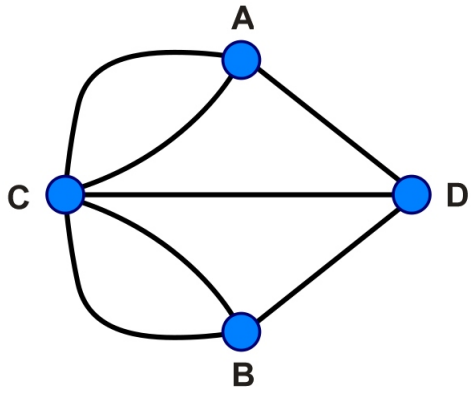

Figura 1.5: Rappresentazione del problema mediante un grafo: i vertici sono le quattro aree della cittá, mentre i collegamenti fra di essi sono i ponti.

tuitivamente supposto che la classe $\boldsymbol{NP}$ fosse piú grande di $\boldsymbol{P}$, in quanto comprende problemi che non si riesce a risolvere efficientemente. Dalla definizione di queste classi risulta scontato che $\boldsymbol{P} \subseteq \boldsymbol{NP}$, anche se non si é ancora trovato un metodo per raffinare ulteriormente l'inclusione; in particolare le conclusioni possibili portano a $\boldsymbol{P} = \boldsymbol{NP}$ oppure $\boldsymbol{P} \subset \boldsymbol{NP}$, sebbene si sospetti che la soluzione corretta sia la seconda.

Capitolo 2

Fondamenti di computazione quantistica

"For those who are not shocked when they first came across quantum theory cannot possibly have understood it"
– Niels Bohr

"Quantum mechanics: Real Black Magic Calculus "
– Albert Einstein

2.1 Operatori lineari e matrici

Prima di introdurre i concetti fondamentali della meccanica quantistica mi sembra necessario soffermarsi su alcuni costrutti matematici derivati dall'algebra lineare che saranno poi utilizzati in seguito. Un operatore lineare A su uno spazio vettoriale V é un'applicazione V $\rightarrow$ V che verifica la proprietá di linearitá:

$$A(\sum_i a_i|v_i\rangle) = \sum_i a_i A(|v_i\rangle) \tag{2.1}$$

certe volte risulta piú comodo utilizzare la rappresentazione matriciale di un operatore lineare, definita da:

$$A|v_j\rangle = \sum_i A_{ij}|v_i\rangle. \tag{2.2}$$

Un vettore non nullo $|v\rangle$ si dice autovettore di un operatore lineare A se $\mathrm{A}|v\rangle = \mathrm{v}|v\rangle$, dove v é un numero complesso detto anche autovalore di A corrispondente a $|v\rangle$. Gli autovalori di A sono le soluzioni dell'equazione caratteristica $\mathrm{c}(\lambda) = 0$, dove $\mathrm{c}(\lambda) \equiv \det |A - \lambda \mathbb{I}|$ é la funzione caratteristica di A. Per il teorema fondamentale dell'algebra ogni polinomio ha almeno una radice complessa, di conseguenza ogni operatore ha almeno un autovalore, con un corrispondente autovettore. Una rappresentazione diagonale per un operatore A su uno spazio vettoriale V é una rappresentazione del tipo $\mathrm{A} = \sum_i \lambda_i |i\rangle\langle i|$, dove i vettori $|i\rangle$ formano un insieme ortonormale di autovettori di A, con corrispondenti autovalori λ_i; un operatore é diagonalizzabile se ammette una rappresentazione diagonale.

Uno spazio di Hilbert é uno spazio vettoriale in cui é definito un prodotto scalare, inoltre lo spazio risulta completo rispetto alla norma indotta da tale prodotto scalare; per comoditá da qui in avanti considereremo sempre operatori in spazi di Hilbert finito-dimensionali, la generalizzazione a dimensioni infinite non cambia le strutture di base, ma rende i calcoli molto piú complessi.

Per ogni operatore lineare A su uno spazio di Hilbert $\mathcal{H}$ esiste un unico operatore lineare $\mathrm{A}^\dagger$ in $\mathcal{H}$ tale che, per tutti i vettori $|v\rangle$, $|w\rangle \in \mathcal{H}$,

$$(|v\rangle, A|w\rangle) = (A^\dagger|v\rangle, |w\rangle). \tag{2.3}$$

l'operatore $\mathrm{A}^\dagger$ viene detto operatore aggiunto dell'operatore A; nel caso in cui un operatore coincida con il suo aggiunto l'operatore viene detto hermitiano, gli operatori hermitiani rappresentano una classe privilegiata di operatori nell'ambito della meccanica quantistica, difatti hanno la proprietá di avere gli elementi di matrice diagonali reali. Considerando la rappresentazione matriciale di un operatore A la relazione (2.3) si scrive come:

$$(A_{ji})^\star = (A^\dagger)_{ij} \rightarrow (A^T_{ij})^\star = (A^\dagger)_{ij} \tag{2.4}$$

Ma $A_{ij} = (A^\dagger)_{ij}$ se A é hermitiano, per cui per l'elemento di matrice diagonale ($A^T_{ii} = A_{ii}$) si ha:

$$A_{ii} = (A_{ii})^\star \tag{2.5}$$

e quindi A_{ii} é reale. Infine un operatore U é detto unitario se $\mathrm{U}^\dagger\mathrm{U} = \mathbb{I}$. Ad esempio non é difficile convincersi che le quattro matrici di Pauli 2 x 2 sono hermitiane e unitarie:

$$\sigma_0 \equiv \mathbb{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \sigma_y = \begin{bmatrix} 0 & -\imath \\ \imath & 0 \end{bmatrix} \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{2.6}$$

Il prodotto tensoriale é un modo di combinare fra di loro spazi vettoriali per ottenere altri spazi di dimensione maggiore. Dati due spazi di Hilbert V e W, di dimensione

m e n rispettivamente il loro prodotto tensoriale si indica con V $\otimes$ W ed é uno spazio vettoriale mn-dimensionale; i suoi elementi sono le combinazione lineari dei prodotti tensoriali $|v\rangle \otimes |w\rangle$ degli elementi $|v\rangle$ di V e $|w\rangle$ di W. Ne consegue che, se $|i\rangle$ e $|j\rangle$ sono basi ortonormali degli spazi V e W, allora $|i\rangle \otimes |j\rangle$ é una base di V $\otimes$ W, il prodotto tensoriale soddisfa per definizione le seguenti proprietá:

1. Dato un qualunque scalare z, un vettore $|v\rangle$ di V e un vettore $|w\rangle$ di W,

$$z(|v\rangle \otimes |w\rangle) = (z|v\rangle) \otimes |w\rangle = |v\rangle \otimes (z|w\rangle). \tag{2.7}$$

2. Per due qualunque vettori $|v_1\rangle$ e $|v_2\rangle$ in V e $|w\rangle$ in W,

$$(|v_1\rangle + |v_2\rangle) \otimes |w\rangle = |v_1\rangle \otimes |w\rangle + |v_2\rangle \otimes |w\rangle. \tag{2.8}$$

3. Per un qualunque vettore $|v\rangle$ in V e due vettori $|w_1\rangle$, $|w_2\rangle$ in W,

$$|v\rangle \otimes (|w_1\rangle + |w_2\rangle) = |v\rangle \otimes |w_1\rangle + |v\rangle \otimes |w_2\rangle. \tag{2.9}$$

2.2 I postulati della meccanica quantistica

All'inizio del ventesimo secolo le teorie della fisica classica erano in grado di descrivere coerentemente la meccanica e la termodinamica dei corpi macroscopici, ciononostante le previsioni riguardanti la struttura atomica in dettaglio e l'interazione radiazione-materia, in particolare il problema di determinare il potere emissivo del corpo nero, fornivano risultati piuttosto dubbi. Queste ambiguitá sono state definitivamente eliminate negli anni '20 con lo sviluppo della meccanica quantistica, grazie ai contributi di fisici come Schroedinger, Heisenberg, Bohr e Pauli. La struttura di questa nuova teoria fisica appena delineata ha messo in luce una serie di proprietá strabilianti e per nulla intuitive relativamente al mondo microscopico, come ad esempio il dualismo onda-particella, il principio di indeterminazione, l'effetto tunnel e l'entanglement. Per di piú la natura intrinsecamente statistica di questa teoria ha introdotto un tipo di indeterminazione che é stato fonte di preoccupazioni e di dibattiti sia per i fisici che per i filosofi riguardo alla struttura della materia. Il punto di vista che in questo momento viene ampiamente accettato dalla comunitá scientifica viene detto interpretazione di Copenhagen[4] e pone al centro della teoria l'atto della misurazione, per citare il fisico tedesco P. Jordan[5] "Le osservazioni non solo disturbano ció che si misura, esse lo producono". Il ruolo della

misura in meccanica quantistica é cosí critico e strano che ognuno puó chiedersi a ragione che cosa costituisca esattamente una misura; per il momento lasciamo al lettore lo stimolo per ulteriori ragionamenti.

Secondo la meccanica classica lo stato di un sistema di n particelle in un determinato istante di tempo t é determinato unicamente dai valori di posizione e velocitá di tutte le particelle a quell'istante di tempo; date le condizioni iniziali e la legge di Newton, le equazioni del moto permettono di calcolare l'evoluzione temporale del sistema ad ogni istante di tempo t. La teoria quantistica si basa invece su architetture matematiche completamente differenti e che possono essere sintetizzate in quattro postulati[6]:

1. L'insieme di tutte le informazioni contenute in un sistema fisico é codificato all'interno di un vettore di stato a norma unitaria, chiamato funzione d'onda ψ, appartenente ad uno spazio di Hilbert associato al sistema. L'evoluzione temporale della funzione d'onda é definita dall'equazione di Schroedinger:

$$\imath\hbar\frac{d}{dt}|\psi(t)\rangle = H|\psi(t)\rangle \tag{2.10}$$

   dove H é un operatore autoaggiunto che definisce l'Hamiltoniana del sistema, mentre $\hbar$ é la costante di Planck ridotta ($\hbar \equiv \frac{h}{2\pi}$, h $\approx$ 6,626 10^{-34} Js)

2. Ogni proprietá misurabile di un sistema quantistico é detta osservabile e viene rappresentata da un operatore autoaggiunto nello spazio di Hilbert del sistema. L'esito di un qualsiasi processo di misura sul sistema di un certo osservabile A restituisce con certezza uno dei suoi autovalori a_i determinati dall'equazione agli autovalori:

$$A|i\rangle = a_i|i\rangle \tag{2.11}$$

   dove $|i\rangle$ é una base ortonormale di autovettori per l'operatore A; a questo punto conviene espandere la funzione d'onda nella stessa base

$$|\psi(t)\rangle = \sum_i c_i(t)|i\rangle \tag{2.12}$$

   di conseguenza la probabilitá che la misura di A fornisca come risultato a_i al tempo t risulta essere:

$$p(a = a_i, t) = |\langle i|\psi(t)\rangle|^2 = |\langle i|\sum_j c_j(t)|j\rangle|^2 = |c_i(t)|^2 \tag{2.13}$$

3. I processi di misura in meccanica quantistica sono tipicamente distruttivi, ovvero dato un sistema nello stato $|\psi\rangle$ la misura dell'osservabile A fornisce come risultato

uno dei suoi autovalori a_i e causa il collasso del sistema nello stato definito da tale autovalore:

$$\frac{P_i|\psi\rangle}{(\langle\psi|P_i|\psi\rangle)^{1/2}} \tag{2.14}$$

dove P_i é l'operatore di proiezione sul sottospazio corrispondente ad a_i, piú specificamente in assenza di degenerazione $P_i = |i\rangle\langle i|$. A questo punto risulta comodo scrivere la probabilitá di ottenere l'autovalore a_i per l'osservabile A come

$$p_i = \langle\psi|P_i|\psi\rangle \tag{2.15}$$

Dalla teoria della probabilitá risulta quindi che il valore medio per l'osservabile A vale:

$$\langle A\rangle = \sum_i a_i p_i = \sum_i a_i\langle\psi|P_i|\psi\rangle = \langle\psi|(\sum_i a_i P_i)|\psi\rangle = \langle\psi|A|\psi\rangle \tag{2.16}$$

4. Lo spazio degli stati di un sistema fisico composto é il prodotto tensoriale degli spazi degli stati dei singoli sistemi fisici componenti. In particolare, dati n sistemi ognuno preparato in uno stato $|\psi_i\rangle$, lo stato complessivo del sistema vale $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \ldots \otimes |\psi_n\rangle$.

2.3 La sovrapposizione e la correlazione Quantistica

Dopo aver immagazzinato qualche concetto base di meccanica dei quanti sorgerebbe spontaneo chiedersi in che modo sia possibile sfruttare la natura quantistica della materia per costruire un calcolatore totalmente innovativo, l'idea si inizia a diffondere negli anni '80 ed é dovuta a R. Feynman. Tipicamente la potenza computazionale di un computer quantistico é dovuta a due fenomeni puramente quantistici come la sovrapposizione (o interferenza) quantistica e l'entanglement.

Il principio di sovrapposizione é una diretta conseguenza della linearitá dell'equazione di Schroedinger: se $|\psi_1(t)\rangle$ e $|\psi_2\rangle$ sono soluzioni linearmente indipendenti dell' eq. (2.8), allora anche una loro combinazione lineare $|\psi(t)\rangle = \alpha|\psi_1(t)\rangle + \beta|\psi_2(t)\rangle$ risolve (2.8), dove α e β sono due numeri complessi qualsiasi. Il fenomeno piú affascinante e controverso della meccanica quantistica risulta peró essere l'entaglement, una proprietá osservata nei sistemi quantistici composti. Per il quarto postulato lo spazio di Hilbert associato ad

un sistema composto é il prodotto tensoriale degli spazi di Hilbert $\mathcal{H}_i$ associati ai singoli componenti del sistema. Consideriamo ora il caso piú semplice di un sistema quantistico bipartito $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2$. Una base per $\mathcal{H}$ si costruisce a partire dal prodotto tensoriale dei vettori di base di $\mathcal{H}_1$ e $\mathcal{H}_2$; nel caso in cui $\mathcal{H}_1$ e $\mathcal{H}_2$ sono bidimensionali i vettori di base sono

$$\{|0\rangle_1, |1\rangle_1\}, \{|0\rangle_2, |1\rangle_2\} \tag{2.17}$$

$$\{|0\rangle_1 \otimes |0\rangle_2, |0\rangle_1 \otimes |1\rangle_2, |1\rangle_1 \otimes |0\rangle_2, |1\rangle_1 \otimes |1\rangle_2\} \tag{2.18}$$

per $\mathcal{H}_1$, $\mathcal{H}_2$ e $\mathcal{H}$ rispettivamente. Lo stato piú generale in $\mathcal{H}$, peró, non risulta essere il prodotto tensoriale di stati rispettivamente in $\mathcal{H}_1$ e $\mathcal{H}_2$, bensí una loro combinazione lineare in virtú del principio di sovrapposizione:

$$|\psi\rangle = \sum_{i,j=0}^{1} c_{ij}|i\rangle_1 \otimes |j\rangle_2 \equiv \sum_{i,j} c_{ij}|ij\rangle. \tag{2.19}$$

Per definizione uno stato in $\mathcal{H}$ viene detto entangled, o non separabile, se non é possibile scriverlo come il semplice prodotto tensoriale di uno stato $|\alpha\rangle_1$ appartenente ad $\mathcal{H}_1$ e uno stato $|\beta\rangle_2$ di $\mathcal{H}_2$; in contrasto, se é possibile scrivere uno stato di $\mathcal{H}$ come

$$|\psi\rangle = |\alpha\rangle_1 \otimes |\beta\rangle_2 \tag{2.20}$$

lo stato $|\psi\rangle$ viene detto separabile. Per esempio lo stato di singoletto di spin

$$|\psi\rangle_1 = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle). \tag{2.21}$$

é uno stato entangled, mentre lo stato

$$|\psi\rangle_2 = \frac{1}{\sqrt{2}}(|01\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |1\rangle \tag{2.22}$$

é uno stato separabile. La sostanziale differenza fra (2.19) e (2.20) rende conto delle intriganti proprietá dell'entanglement, in particolare attraverso questo fenomeno é stato mostrato in un celebre articolo del 1935 firmato da Einstein, Podolski e Rosen (detto anche EPR dalle iniziali degli autori)[7] come la meccanica quantistica abbia una natura contraddittoria se si assume per vera l'ipotesi del realismo locale, basata su due assunzioni:

1. Principio di realtá: Se risulta possibile predire con certezza il valore di una quantitá fisica allora questa quantitá possiede una realtá fisica indipendentemente dall'osservazione; ovvero se la funzione d'onda di un sistema $|\psi\rangle$ si trova in un autostato di un operatore A, $A|\psi\rangle = a|\psi\rangle$, allora il valore a dell'osservabile A é un elemento di realtá fisica.

2. Principio di localitá: Se due sistemi sono causalmente disconnessi, ovvero quando $(\Delta x)^2 > c^2(\Delta t)^2$ dove Δx e Δt sono le distanze spaziali e temporali fra i due eventi in un sistema inerziale, allora il risultato di una misura effettuato sul primo sistema non puó influenzare l'esito di una misura effettuata sul secondo sistema.

Due sistemi entangled come ad esempio lo stato (2.21) non verificano le ipotesi EPR, in quanto l'esito di una misura sul primo componente influenza istantaneamente l'esito della stessa misura sul secondo; vedremo in seguito nel paragrafo 2.6 un importante teorema che garantisce la coerenza interna della meccanica quantistica pur spiegando fenomeni come la sovrapposizione e l'entanglement.

2.4 Il Qubit e la sfera di Bloch

Un bit classico é rappresentato da un sistema che puó esistere in due stati diversi, come ad esempio un circuito in cui passa o meno corrente, invece un qubit (quantum bit) é rappresentato da un sistema quantistico a due livelli in uno spazio di Hilbert bidimensionale. In questo spazio risulta possibile scegliere una coppia di vettori di base ortonormali per rappresentare i corrispondenti valori 0 e 1 del bit classico. Questi due stati rappresentano la base computazionale e un qualsiasi stato di ogni qubit pu essere scritto come

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{2.23}$$

con la condizione di normalizzazione:

$$|\alpha|^2 + |\beta|^2 = 1 \tag{2.24}$$

Tuttavia, a differenza di un bit classico, un qubit vive in uno spazio vettoriale parametrizzato da variabili continue (ad esempio α e β); di conseguenza, per i postulati della meccanica quantistica, puó venire a trovarsi non solo nello stato $|0\rangle$ o nello stato $|1\rangle$, ma in una qualsiasi sovrapposizione di questi stati. A questo punto si potrebbe essere tentati di affermare che un singolo qubit sia in grado di immagazzinare una quantitá infinita di informazione classica, ma ció risulta scorretto, in quanto sarebbero necessarie infinite misure su singoli qubit preparati tutti nello stesso stato per ottenere il giusto valore di α e β.

Un sistema a due livelli puó essere utilizzato come qubit se é possibile effettuare le seguenti operazioni:

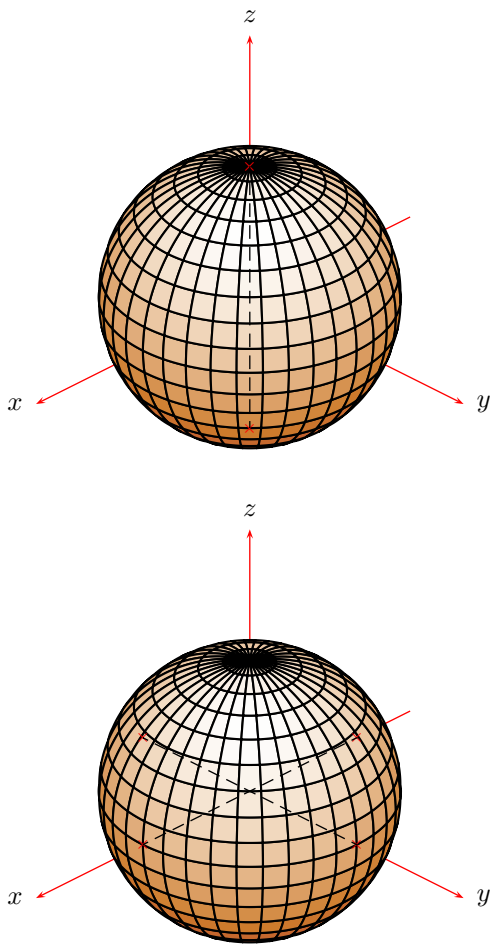

Figura 2.1: Rappresentazione artistica della sfera di Bloch: i due poli della sfera corrispondono agli stati puri $|0\rangle$ e $|1\rangle$, mentre i punti sull'equatore corrispondono ad una sovrapposizione omogenea degli stati $|0\rangle$ e $|1\rangle$

1. Il sistema puó essere preparato in uno stato noto, detto anche stato puro o di fiducia del qubit;

2. Ogni stato del qubit puó essere trasformato in un altro attraverso una trasformazione unitaria;

3. Lo stato del qubit dopo ogni trasformazione puó essere misurato in riferimento alla base computazionale ($|0\rangle$, $|1\rangle$).

Risulta piú immediato, nonché comodo, utilizzare una rappresentazione geometrica del qubit e delle trasformazioni che operano sul suo stato; difatti, la condizione di normalizzazione 2.9 costringe il vettore di stato a muoversi su una sfera di raggio unitario, detta anche sfera di Bloch.

Utilizzando le coordinate polari sferiche la sfera di Bloch puó essere immersa in uno spazio tridimensionale di coordinate Cartesiane x = cos ϕ sin θ , y =sin ϕ sin θ , z = cos θ, di conseguenza un vettore di Bloch é caratterizzato dai due parametri angolari (θ,ϕ) oppure dalle tre coordinate cartesiane (x,y,z) con la condizione di normalizzazione $x^2 + y^2 + z^2 = 1$.

Sulla sfera di Bloch, prendendo come riferimento la base computazionale, lo stato (2.23) puó essere scritto come:

$$\begin{gathered} |\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{\imath\phi}\sin\frac{\theta}{2}|1\rangle \\ (0 \leq \theta \leq \pi, 0 \leq \phi \leq 2\pi). \end{gathered} \tag{2.25}$$

i valori di aspettazione per le tre coordinate di un qubit sulla sfera di Bloch si ottengono calcolando il valore di aspettazione dei rispettivi operatori di Pauli nella base computazionale. Per lo stato ψ dato dalla (2.25) e applicando le matrici di Pauli in (2.6) si ha

$$\begin{aligned} \sigma_x|\psi\rangle &= e^{\imath\phi}\sin\frac{\theta}{2}|0\rangle + \cos\frac{\theta}{2}|1\rangle, \\ \sigma_y|\psi\rangle &= -\imath e^{\imath\phi}\sin\frac{\theta}{2}|0\rangle + \imath\cos\frac{\theta}{2}|1\rangle, \\ \sigma_z|\psi\rangle &= \cos\frac{\theta}{2}|0\rangle - e^{\imath\phi}\sin\frac{\theta}{2}|1\rangle. \end{aligned} \tag{2.26}$$

da cui i valori di aspettazione

$$\begin{aligned} \langle\psi|\sigma_x|\psi\rangle &= \sin\theta\cos\phi = x, \\ \langle\psi|\sigma_y|\psi\rangle &= \sin\theta\sin\phi = y, \\ \langle\psi|\sigma_z|\psi\rangle &= \cos\theta = z. \end{aligned} \tag{2.27}$$

2.5 Circuiti e Logica Quantistica

Un computer classico, secondo il modello delle MdT, puó essere convenientemente rappresentato come un registro finito di n bits; attraverso operazioni logiche elementari, come il NOT e l'AND risulta poi possibile agire su un singolo bit o su coppie o gruppi di bits per poi produrre una qualsiasi funzione logica complessa; le porte AND, OR e NOT hanno la peculiaritá di essere universali, per cui qualsiasi altra operazione logica puó essere costruita a partire da una combinazione di questi 3 gates. In realtá non sono nemmeno necessarie 3 porte diverse, in quanto AND, OR e NOT possono essere tutti costruiti a partire dalla funzione logica NAND[8], implementata tramite transistor a giunzione bipolare (BJT) o ad effetto di campo (FET)[9]. In notazione binaria un qualsiasi numero intero i $\in [0, 2^n - 1]$,

$$i = i_{n-1}2^{n-1} + \cdots + i_1 2 + i_0 \tag{2.28}$$

e $i_0, i_1, \cdots, i_{n-1}$ costituiscono la rappresentazione binaria di i. Risulta possibile estendere questo modello circuitale anche alla computazione quantistica; in particolare lo stato di un computer quantistico puó essere visto come lo stato composto degli n qubits che costituiscono il suo registro. In questo caso

$$|\psi\rangle = \sum_{i=0}^{2^n-1} c_i|\psi\rangle = \sum_{i_{n-1}=0}^{1} \cdots \sum_{i_1=0}^{1}\sum_{i_0=0}^{1} c_{i_{n-1},\cdots,i_1,i_0}|i_{n-1}\rangle \otimes \cdots \otimes |i_1\rangle \otimes |i_o\rangle, \tag{2.29}$$

con la condizione di normalizzazione

$$\sum_{i=0}^{2^n-1} |c_i|^2 = 1 \tag{2.30}$$

in questo modo discende direttamente dai postulati della meccanica quantistica che lo stato di un computer quantistico a n-qubit é una funzione d'onda che vive in uno spazio

di Hilbert 2^n-dimensionale, costruito a partire dai prodotti tensoriali di n spazi di Hilbert bidimensionali. Considerando che lo stato di ogni qubit é definito da due numeri complessi, con la condizione di normalizzazione (2.30) lo stato di un computer quantistico a n qubit é determinato da $2(2^n - 1)$parametri indipendenti.

Un qualsiasi operazione logica applicata ad un qubit deve preservarne la condizione di normalizzazione, perció le operazioni sui singoli qubit sono definite da matrici unitarie 2 x 2. L'operatore di Hadamard é definito come:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.31}$$

e costituisce un operatore di importanza fondamentale per la computazione quantistica, in quanto trasforma la base computazionale $\{|0\rangle, |1\rangle\}$ in una nuova base $\{|+\rangle, |-\rangle\}$, composta dalla sovrapposizione degli stati della base computazionale:

$$\begin{aligned} H|0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \equiv |+\rangle, \\ H|1\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \equiv |-\rangle. \end{aligned} \tag{2.32}$$

Il gate di Hadamard, oltre ad essere unitario, risulta essere anche hermitiano e verifica $H^2 = \mathbb{I}$, da cui $H^{-1} = H$. Un altro operatore di interesse é l'operatore di fase:

$$R_z(\delta) = \begin{bmatrix} 1 & 0 \\ 0 & e^{\imath\delta} \end{bmatrix} \tag{2.33}$$

la sua azione é quella di lasciare invariato lo stato $|0\rangle$ e di aggiungere un fattore di fase $e^{\imath\delta}$ allo stato $|1\rangle$. Dal postulato 1 sappiamo che le funzioni d'onda devono avere norma unitaria, dunque la moltiplicazione per un fattore di fase $e^{\imath\delta}$ a modulo uno non influenza la condizione di normalizzazione; d'altra parte $R_z(\delta)$ é per definizione unitario. L'azione di (2.33) sullo stato (2.25) ad esempio vale:

$$R_z(\delta)|\psi\rangle = \begin{bmatrix} 1 & 0 \\ 0 & e^{\imath\delta} \end{bmatrix} \begin{bmatrix} \cos\frac{\theta}{2} \\ e^{\imath\phi}\sin\frac{\theta}{2} \end{bmatrix} = \begin{bmatrix} \cos\frac{\theta}{2} \\ e^{\imath(\phi+\delta)}\sin\frac{\theta}{2} \end{bmatrix}. \tag{2.34}$$

e non é nient'altro che una rotazione di un angolo δ in senso antiorario attorno all'asse z della sfera di Bloch. In particolare é possibile rappresentare una qualsiasi trasformazione unitaria da un punto ad un altro della sfera di Bloch utilizzando solamente operatori di Hadamard e di fase.[6]

Per far apparire l'entanglement, il fenomeno quantistico piú interessante da sfruttare, occorre operare con almeno due qubits; in virtú del quarto postulato uno stato generico a due qubit puó essere scritto nella base computazionale come:

$$|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle, \tag{2.35}$$

con α, β, γ e δ coefficienti complessi. Considerando la condizione di normalizzazione $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$ e il fatto che ogni stato é definito a meno di un fattore di fase, come accennato prima, si ottiene che $|\psi\rangle$ ha, nel caso piú generale 6 gradi di libertá; se invece lo stato é separabile, come ad esempio (2.20), é possibile descrivere i due qubit sulla sfera di Bloch con due parametri ciascuno, quindi questo sistema ha solo 4 gradi di libertá. Un tipico operatore a 2 qubit in grado di generare entanglement é il controlled-not gate, o $CNOT$ gate. Il primo qubit in un CNOT gate agisce come bit di controllo, mentre il secondo qubit funge da bersaglio; l'operatore scambia lo stato del qubit bersaglio solamente se il qubit di controllo si trova nello stato $|1\rangle$, mentre lo lascia inalterato quando lo stato del qubit di controllo é $|0\rangle$. I vettori della base computazionale possono essere rappresentati come vettori colonna, ottenendo:

$$|0\rangle \equiv |00\rangle = \begin{bmatrix}1\\0\\0\\0\end{bmatrix}, |1\rangle \equiv |01\rangle = \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, |2\rangle \equiv |10\rangle = \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, |3\rangle \equiv |11\rangle = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}, \tag{2.36}$$

da cui segue la rappresentazione matriciale del CNOT gate

$$CNOT = \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{bmatrix} \tag{2.37}$$

dove le componenti $(CNOT)_{ij} = \langle i|CNOT|j\rangle$; come il gate di Hadamard, $CNOT$ verifica ($CNOT^2 = \mathbb{I}$. Sfruttando uno sovrapposizione degli stati della base computazionale risulta possibile generare stati non separabili, ad esempio,

$$CNOT(\alpha|0\rangle + \beta|1\rangle)|0\rangle = \alpha|00\rangle + \beta|11\rangle, \tag{2.38}$$

é entangled quando $\alpha, \beta \neq 0$. Il gate CNOT risulta fondamentale nell'ambito della computazione quantistica per la sua proprietá di essere universale[3], proprio come la porta NAND per la logica classica.

2.6 Il teorema No-Cloning

Operando con un sistema classico non é difficile costruire un circuito in grado di duplicare uno o piú bit, il circuito viene detto di FAN-OUT e si realizza semplicemente sdoppiando i fili di collegamento. Invece, come giá affermato a partire dal terzo postulato (2.14), le misure quantistiche sono tipicamente distruttive, ovvero alterano lo stato del sistema misurato. Di conseguenza non é possibile costruire un gruppo di copie identiche (cloni) di uno stato quantistico sconosciuto; in aggiunta se per assurdo si potesse costruire una macchina quantistica clonatrice la meccanica quantistica sparirebbe. Questo risultato é stato provato per la prima volta nel 1982 da Wootters, Zurek e Dieks[10] ed é noto col nome di Teorema No-Cloning.

Supponiamo di avere una macchina quantistica con due compartimenti A e B. Lo stato iniziale della macchina vale $|\psi\rangle \otimes |s\rangle$, dove $|\psi\rangle$ é lo stato da copiare, contenuto in A e $|s\rangle$ é lo stato bersaglio. L'azione della macchina risulta definita da un operatore unitario U per cui:

$$|\psi\rangle \otimes |s\rangle \rightarrow U(|\psi\rangle \otimes |s\rangle) = |\psi\rangle \otimes |\psi\rangle. \tag{2.39}$$

allora, per due stati puri $|\psi\rangle$ e $|\phi\rangle$:

$$\begin{aligned} U(|\psi\rangle \otimes |s\rangle) &= |\psi\rangle \otimes |\psi\rangle \\ U(|\phi\rangle \otimes |s\rangle) &= |\phi\rangle \otimes |\phi\rangle. \end{aligned} \tag{2.40}$$

e, prendendo il prodotto scalare delle (2.40),

$$\langle\psi|\phi\rangle = (\langle\psi|\phi\rangle)^2. \tag{2.41}$$

che ha soluzione solo per $\langle\psi|\phi\rangle = 0, 1$, per cui o i due stati coincidono oppure sono ortogonali; ne consegue che una macchina clonatrice é in grado di replicare stati fra loro ortogonali, mentre fallisce con una loro qualsiasi combinazione lineare (2.23 ad esempio).

Il teorema No-Cloning sancisce dunque l'impossibilitá di sfruttare i fenomeni quantistici come l'entanglement per trasmettere informazione fra due sistemi a velocitá superluminali, in completo accordo con la teoria della relativitá; questo risultato é fondamentale nell'ambito della teoria quantistica e non ha nessun tipo di analogia con una trattazione di tipo classico. Occorre quindi sviluppare strategie piú fini in grado di sfruttare al meglio le peculiaritá dei sistemi quantistici per ottenere qualche risultato nello studio della computazione quantistica.

Capitolo 3

Implementazione di algoritmi quantistici

"If computers that you build are quantum,
Then spies everywhere will all want 'em.
Our codes will all fail,
And they'll read our email,
Till we get crypto that's quantum, and daunt 'em"
– Jennifer e Peter Shor

Il computer classico nasce dal progetto teorico di Von Neumann[11] che negli anni '40 concepí un modello generale per costruire una MdT universale; nel 1947, con l'invenzione del transistor iniziarono ad essere realizzati i primi componenti hardware. Da allora il progresso nella produzione tecnologica di elementi microelettronici ha avuto un andamento esponenziale nel tempo, come sintetizzato dalla legge di Moore (1965)[9]:

Il numero di transistor per circuito integrato raddoppia circa ogni due anni.

Ne consegue che, ad un certo punto, la continua miniaturizzazione delle componenti elettroniche dei circuiti integrati porterá inevitabilmente al raggiungimento di una scala di grandezze in cui i fenomeni quantistici non risulteranno piú trascurabili; in questo senso risulta comodo e sicuramente innovativo ragionare secondo i paradigmi della computazione quantistica. Il paradosso EPR sintetizza in modo piuttosto efficace la sostanziale differenza fra l'informazione contenuta in un sistema quantistico rispetto ad un sistema classico, é possibile peró sfruttare i fenomeni quantistici per risolvere problemi piú velo-

cemente di un computer classico? Cercheremo di dare una risposta a questa domanda nei prossimi paragrafi.

Non é sorprendente verificare che un circuito composto da gate quantistici é in grado di simulare un circuito logico classico; questo é principalmente dovuto all'unitarietá delle trasformazioni quantistiche, di conseguenza un circuito classico puó essere rimpiazzato da un circuito equivalente contenente solo elementi invertibili, usando semplicemente una porta, detta porta di Toffoli (T). Una porta T ha 3 bit in input e altrettanti in output, i primi due bit sono di controllo e rimangono inalterati dall'azione della porta, mentre il terzo bit viene cambiato di segno quando entrambi i bit di controllo sono posti uguali ad 1.

Input	Output
A,B,C	A',B',C'
0 , 0 , 0	0 , 0 , 0
0 , 0 , 1	0 , 0 , 1
0 , 1 , 0	0 , 1 , 0
0 , 1 , 1	0 , 1 , 1
1 , 0 , 0	1 , 0 , 0
1 , 0 , 1	1 , 0 , 1
1 , 1 , 0	1 , 1 , 1
1 , 1 , 1	1 , 1 , 0

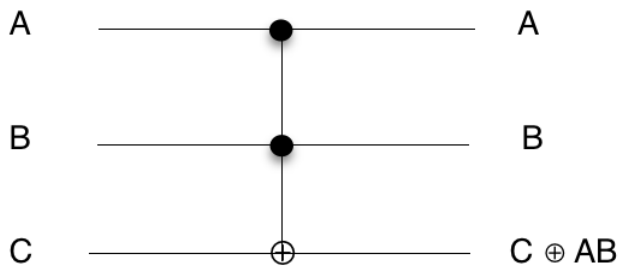

Figura 3.1: Rappresentazione circuitale del gate di Toffoli.

Usando una porta T é possibile realizzare una porta NAND[3] (figura 3.2) e un circuito di duplicazione (FANOUT), di conseguenza risulta possibile simulare qualsiasi altro tipo di circuito classico, secondo quanto affermato all'inizio della sezione 2.5; infine l'implementazione quantistica della porta T si ottiene utilizzando la base computazionale (2.23), in

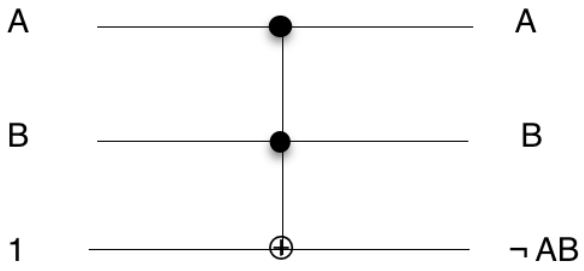

Figura 3.2: Esempio di porta NAND costruita a partire da un gate di Toffoli.

questo caso la sua azione sará definita da una matrice unitaria 8 x 8. Chiaramente non vi é tanto interesse nell'impiegare le proprietá della meccanica quantistica per ottenere il limite classico, di conseguenza qubit e porte logiche devono essere impiegati in modo da sfruttare al meglio tutte le proprietá che derivano dai 4 postulati del capitolo 2.

3.1 Il problema di Deutsch-Josza

Il problema di Deutsch-Jozsa illustra chiaramente il potere computazionale dell'interferenza quantistica; come la maggior parte degli algoritmi quantistici sfrutta il fenomeno del parallelismo quantistico per valutare il valore di una funzione f(x) per differenti valori di x simultaneamente, ottenendo risultati sconosciuti ad un normale computer. Consideriamo una scatola nera in grado di applicare una funzione booleana ad 1 bit f: $\{0,1\} \rightarrow \{0,1\}$ come rappresentato in tabella, fra le quattro funzioni fatte in questo modo due sono costanti e due bilanciate.

x	f_0 , f_1 , f_2 , f_3
0	0 , 0 , 1 , 1
1	0 , 1 , 0 , 1

Per risolvere il problema occorre determinare che funzione viene applicata dalla macchina interrogando il sistema il minimo numero di volte. Per effettuare una trasformazione reversibile occorre utilizzare una trasformazione unitaria U_f su due qubit del tipo

$|x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$, dove $\oplus$ indica l'addizione modulo 2, che classicamente ha l'effetto di cambiare il valore del secondo qubit quando $f(x) = 1$;

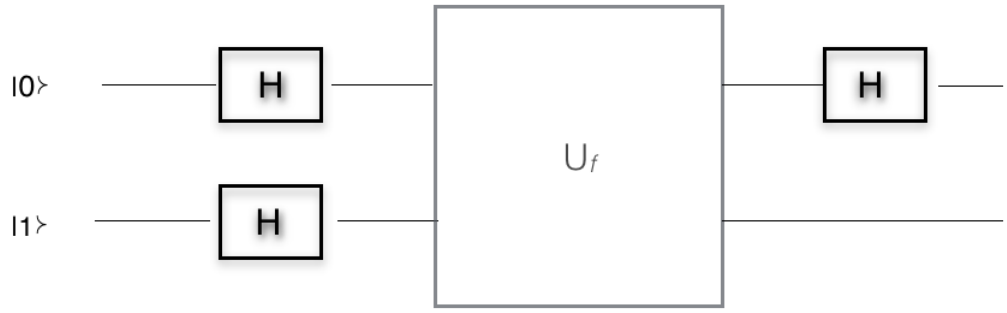

Figura 3.3: Rappresentazione circuitale del problema utilizzando il gate unitario U_f

Applicando U_f agli stati $|+\rangle$ e $|-\rangle$ in (2.32) si ottiene:

$$\begin{aligned} &U_f|+\rangle|-\rangle = |+\rangle\frac{1}{\sqrt{2}}(|f(+)\rangle - |1 \oplus f(+)\rangle = \\ &\frac{1}{2}(|0\rangle(|f(0) \oplus 0\rangle - |f(0) \oplus 1\rangle) + |1\rangle(|f(1) \oplus 0\rangle - |f(1) \oplus 1\rangle)) = \\ &= \begin{cases} \pm\frac{(|0\rangle+|1\rangle)}{\sqrt{2}}\frac{(|0\rangle-|1\rangle)}{\sqrt{2}} & se\ f(0) = f(1) \\ \pm\frac{(|0\rangle-|1\rangle)}{\sqrt{2}}\frac{(|0\rangle-|1\rangle)}{\sqrt{2}} & se\ f(0) \neq f(1) \end{cases} \end{aligned} \tag{3.1}$$

applicando poi un gate di Hadamard al primo qubit si ottiene lo stato finale

$$|\psi\rangle = \begin{cases} \pm|0\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}} & se\ f(0) = f(1) \\ \pm|1\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}} & se\ f(0) \neq f(1). \end{cases} \tag{3.2}$$

e ricordando le proprietá dell'addizione modulo 2, ovvero $f(0) \oplus f(1) = 0 \rightarrow f(0) = f(1)$, la ψ puó essere riscritta come:

$$|\psi\rangle = \pm|f(0) \oplus f(1)\rangle\frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{3.3}$$

per cui risulta possibile risalire ad una proprietá globale di f, $f(0) \oplus f(1)$ effettuando la misura di un singolo qubit; nessun apparecchio classico risulta in grado di lavorare cosí velocemente.

A questo punto é possibile generalizzare il procedimento per ottenere l'algoritmo di Deutsch-Jozsa[12]: Alice si trova ad Amsterdam e viene sfidata ad un gioco da Bob, che si trova a Boston per lavoro; Bob chiede ad Alice di inviargli un numero x a n bit compreso da 0 a $2^n - 1$, dopodiché Bob applicherá la sua funzione ad x e risponderá ad Alice uno 0 o un 1 a seconda del numero da lei inviato. Bob ha promesso che la sua funzione é costante oppure é uguale a 0 esattamente per lo stesso numero di valori per il quale é uguale a 1 (bilanciata). Ci chiediamo quanti numeri debba inviare Alice per riuscire a determinare la funzione utilizzata da Bob. L'algoritmo classico e deterministico piú efficiente utilizzabile da Alice richiederebbe comunque nel caso peggiore $\frac{2^n}{2} + 1$ interrogazioni, in quanto potrebbe ricevere $\frac{2^n}{2}$ zeri prima di ottenere un 1 e stabilire che la funzione non é costante. Utilizzando il metodo visto appena prima, peró Alice potrebbe usare un registro quantistico a n qubit e risolvere il problema con un solo tentativo, dando a Bob un solo qubit in cui immagazzinare la risposta. Prima di tutto Alice prepara il suo stato fiduciario, composto dal registro di n qubit, tutti preparati nello stato $|0\rangle$ e un qubit ausiliario nello stato $|1\rangle$ necessario per l'applicazione di U_f:

$$|\psi_0\rangle = |0\rangle^{\otimes n}|1\rangle. \tag{3.4}$$

dove $^{\otimes n}$ indica il prodotto tensoriale di n spazi identici. Risulta poi necessario applicare al registro n porte di Hadamard in parallelo, ovvero

$$H^{\otimes n} = H \otimes H \otimes \cdots \otimes H. \tag{3.5}$$

e, per uno stato $|x\rangle$ nella base computazionale

$$H^{\otimes n}|x\rangle = \prod_{i=0}^{n-1}(\frac{1}{\sqrt{2}}\sum_{y_i=0}^{1}(-1)^{x_i y_i}|y_i\rangle) = \frac{1}{2^{\frac{n}{2}}}\sum_{y=0}^{2^n-1}(-1)^{x\cdot y}|y\rangle, \tag{3.6}$$

dove $x \cdot y$ é il prodotto scalare modulo 2 di due parole a n-bit:

$$x \cdot y = x_{n-1}y_{n-1} \oplus x_{n-2}y_{n-2} \oplus \cdots \oplus x_0 y_0 \tag{3.7}$$

Si puó adesso notare che la (3.1) implica che, quando $|y\rangle = |-\rangle$, $U_f|x\rangle|y\rangle = -1^{f(x)}|x\rangle|y\rangle$; allora se Alice utilizza la trasformazione

$$(H^{\otimes n} \otimes \mathbb{I})U_f(H^{\otimes n} \otimes H) \tag{3.8}$$

sul suo stato iniziale (3.4) otterrá come risultato lo stato

$$\psi_1 = \left(\frac{1}{2^n} \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^n-1} (-1)^{f(x)+x\cdot y} |y\rangle \right) \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle). \tag{3.9}$$

La (3.9) potrebbe anche sembrare criptica ad un primo approccio, tuttavia se osserviamo quanto vale la probabilitá di ottenere come risultato lo stato $|0\rangle^{\otimes n}$ abbiamo che questa é uguale a:

$$|\langle 0^{\otimes n}|\psi_1\rangle|^2 = |\sum_{x=0}^{2^n-1} \frac{(-1)^{f(x)}}{2^n}|^2; \tag{3.10}$$

perció quando f é costante la probabilitá di ottenere $|0\rangle^{\otimes n}$ é uno, di conseguenza qualsiasi misura effettuata su un qubit nel registro dará come risultato lo stato $|0\rangle$. Al contrario, se f é bilanciata, le componenti nella sommatoria (3.10) si cancellano vicendevolmente portando ad una probabilitá nulla di avere un qualsiasi qubit del registro nello stato $|0\rangle$. A questo punto Alice puó misurare un qualsiasi qubit e indovinare la funzione di Bob effettuando una sola operazione! Un risultato sorprendente rispetto alle richieste di un computer classico. Sfortunatamente il problema di Deutsch non costituisce un problema d'interesse per le teorie dell'informazione quantistica in quanto non possiede applicazioni pratiche note, ciononostante costituisce un importante esempio della potenza di calcolo quantistica e contiene una sintesi delle proprietá quantistiche di base che devono essere sfruttate da un algoritmo per essere efficiente.

3.2 La trasformata di Fourier quantistica

La trasformata discreta di Fourier di un vettore complesso ad N componenti $\{f(0), f(1), \cdots, f(N-1)\}$ é un nuovo vettore complesso con le componenti $\{\tilde{f}(0), \tilde{f}(1), \cdots, \tilde{f}(N-1)\}$ definite da:

$$\tilde{f}(k) = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi\imath \frac{kj}{N}} f(j). \tag{3.11}$$

La trasformata quantistica di Fourier (QFT) é definita per un registro quantistico a n qubits (N = 2^n), agisce esattamente allo stesso modo e puó essere rappresentata dall'operatore unitario F:

$$F(|j\rangle) = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi\imath \frac{kj}{2^n}} |k\rangle \tag{3.12}$$

la cui azione non é nient'altro che un cambio di base

$$|\tilde{\psi}\rangle = F|\psi\rangle = F\sum_{j=0}^{2^n-1} f(j)|j\rangle = \sum_{k=0}^{2^n-1} \tilde{f}(k)|k\rangle. \tag{3.13}$$

Applicando un po' di algebra alla (3.12) é possibile ottenere una definizione alternativa della trasformata di Fourier quantistica; questa rappresentazione, detta anche rappresentazione per prodotti, risulta utile per costruire un circuito quantistico efficiente in grado di applicare la QFT. Considerando la rappresentazione binaria per uno stato $|j\rangle$, come in (2.36)

$$j = j_{n-1}j_{n-2}\cdots j_0 = j_{n-1}2^{n-1} + j_{n-2}2^{n-2} + \cdots + j_0, \tag{3.14}$$

e, analogamente, la rappresentazione di frazione binaria

$$0._{j_l j_{l+1}\cdots j_m} = \frac{1}{2}j_l + \frac{1}{4}j_{l+1} + \cdots + \frac{1}{2^{m-l+1}}j_{m.} \tag{3.15}$$

Applicando la (3.14) alla (3.12) si ha:

$$\begin{aligned}
|j_1\cdots j_n\rangle = |j\rangle \to \frac{1}{2^{\frac{n}{2}}}\sum_{k=0}^{2^n-1} e^{2\pi\imath\frac{kj}{2^n}}|k\rangle = \\
= \frac{1}{2^{\frac{n}{2}}}\sum_{k_1=0}^{1}\cdots\sum_{k_n=0}^{1} e^{2\pi\imath j\left(\sum_{l=1}^{n} k_l 2^{-l}\right)}|k_1\cdots k_n\rangle = \\
= \frac{1}{2^{\frac{n}{2}}}\sum_{k_1=0}^{1}\cdots\sum_{k_n=0}^{1}\bigotimes_{l=1}^{n} e^{2\pi\imath j k_l 2^{-l}}|k_l\rangle = \\
= \frac{1}{2^{\frac{n}{2}}}\bigotimes_{l=1}^{n}\left[\sum_{k_l=0}^{1} e^{2\pi\imath j k_l 2^{-l}}|k_l\rangle\right] = \\
= \frac{1}{2^{\frac{n}{2}}}\bigotimes_{l=1}^{n}\left[|0\rangle + e^{2\pi\imath j 2^{-l}}|1\rangle\right]
\end{aligned} \tag{3.16}$$

da cui, per la (3.15), si ottiene:

$$|j_1,\cdots,j_n\rangle \to \frac{\left(|0\rangle + e^{2\pi\imath 0._{j_n}}|1\rangle\right)\left(|0\rangle + e^{2\pi\imath 0._{j_{n-1}j_n}}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi\imath 0._{j_1 j_2\cdots j_n}}|1\rangle\right)}{2^{\frac{n}{2}}}. \tag{3.17}$$

Come giá annunciato prima, non é difficile fornire una rappresentazione circuitale per la (3.17): questa é composta da gate di Hadamard (2.31), di fase (2.33) e da operazioni extra di scambio fra qubit.

Ricordando le (2.32) l'applicazione dell'operatore di Hadamard al primo qubit dello stato $|j\rangle$ porta a

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi\imath 0.j_1}|1\rangle\right)|j_2\cdots j_n\rangle, \tag{3.18}$$

siccome $e^{2\pi\imath 0.j_1} = -1$ quando $j_1 = 1$ ed é uguale ad 1 altrimenti. Definiamo ora una collezione di operatori di fase unitari R_k:

$$R_k \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi\imath}{2^k}} \end{bmatrix} \tag{3.19}$$

Prendendo come riferimento la base computazionale, gli R_k hanno come effetto quello di lasciare inalterato lo stato $|0\rangle$, mentre lo stato $|1\rangle$ viene moltiplicato per un fattore di fase $e^{\frac{2\pi\imath}{2^k}}$. Applicando ad esempio R_2 allo stato (3.18) si ottiene:

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi\imath 0.j_1 j_2}|1\rangle\right)|j_2\cdots j_n\rangle, \tag{3.20}$$

e applicando in successione $R_3, R_4, \cdots, R_n$ lo stato finale del sistema sará

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi\imath 0.j_1 j_2\cdots j_n}|1\rangle\right)|j_2\cdots j_n\rangle, \tag{3.21}$$

Ora non é difficile convincersi che, ragionando analogamente con i rimanenti n-1 qubits, si puó arrivare allo stato finale

$$\frac{1}{2^{\frac{n}{2}}}\left(|0\rangle + e^{2\pi\imath 0.j_1 j_2\cdots j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi\imath 0.j_2\cdots j_n}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi\imath 0.j_n}|1\rangle\right) \tag{3.22}$$

il quale corrisponde, a meno di operazioni di scambio fra i qubit, alla (3.17). Quanti gates sono impiegati dal circuito? Sul primo qubit agiscono un gate di Hadamard e n - 1 gates di fase, per un totale di n gates; il secondo qubit é soggetto a un gate di Hadamard e n - 2 gates di fase. Nel processo totale sono dunque impiegati $\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$. Considerando anche le operazioni di scambio fra qubit si puó affermare che l'algoritmo accennato qua sopra calcola la QFT in $\Theta(n^2)$ operazioni. In contrasto, i migliori algoritmi noti per applicare la trasformata discreta di Fourier, come ad esempio la Fast Fourier Transform (FFT) necessitano almeno $\Theta(n2^n)$ operazioni[3]; questo speed-up esponenziale potrebbe essere cruciale per le operazioni che richiedono applicazioni in massa di questo strumento, come ad esempio la correzione e il ritocco di immagini digitali o il riconoscimento dei fonemi di base nei campioni di suoni digitalizzati. Sfortunatamente, le informazioni della QFT sono contenute all'interno delle coordinate dei qubit rispetto alla base computazionale e non é possibile accedervi con una misura diretta. Risulta quindi necessario impiegare questo nuovo strumento in maniera intelligente, in modo da sfruttare tutte le sue potenzialitá.

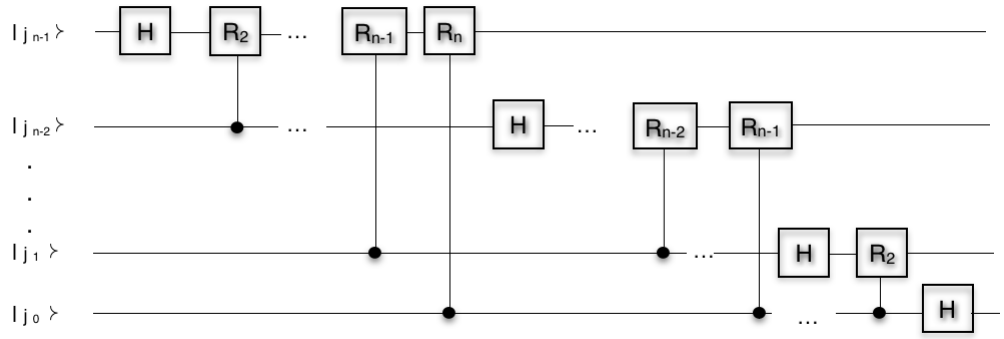

Figura 3.4: Rappresentazione circuitale della QFT

3.3 L'algoritmo di Shor e la scomposizione in fattori primi

Al di lá delle fenomenali e per nulla intuitive proprietá quantistiche della materia, la piú grande scoperta nell'ambito della computazione quantistica é sicuramente l'algoritmo di Shor[13]: un algoritmo probabilistico in grado di risolvere efficientemente, almeno in linea teorica, il problema della scomposizione in fattori primi che, come giá accennato nel capitolo 1, pare appartenere alla classe di complessitá $\boldsymbol{NP}$ e risulta quindi inattaccabile da una MdT classica. La maggior parte dei sistemi crittografici nel mondo utilizzano estensivamente un protocollo di sicurezza denominato RSA, basato su certe proprietá particolari dei numeri primi; ad oggi il migliore algoritmo classico per la scomposizione, detto anche crivello a campo numerico generale, impiega $\exp(O(n^{\frac{1}{3}}(\log n)^{\frac{2}{3}}))$ operazioni[14], quindi un tempo super-polinomiale per un input di lunghezza n. L'algoritmo di Shor é in grado di risolvere lo stesso problema in $O(n^2 \log n \log\log n)$ [6] operazioni, realizzando uno speed-up esponenziale rispetto a qualsiasi altro algoritmo noto.

La complessitá del problema nasce da un'asimmetria di base della funzione potenza discreta $f(x) = a^x \pmod N$, ovvero $f(x)$ é uguale al resto della divisione fra a^x e N; é facile calcolare $f(x)$ noti gli altri due parametri, ad esempio per a = 4 e N = 18:

$$\begin{aligned} f(2) = 4^2 = 16 \equiv 16 \pmod{18} \\ f(4) = 4^4 = 256 \equiv 4 \pmod{18} \\ f(5) = 4^5 = 1024 \equiv 16 \pmod{18} \end{aligned} \tag{3.23}$$

Tuttavia non é altrettanto facile ricavare x a partire da $f(x)$; per di piú la funzione f é periodica, per cui il problema inverso ha addirittura infinite soluzioni. Il problema della fattorizzazione puó essere ridotto al problema di determinare il periodo della funzione $f(x)$, ovvero il piú piccolo intero che verifica $f(x+r) = f(x)$, dove N é il numero da scomporre e a < N viene scelto casualmente. Difatti, il noto teorema di Eulero per l'aritmetica modulare afferma che, se N ed a sono coprimi (ovvero non hanno divisori in comune), allora esiste un $r > 0$ tale che:

$$a^r \equiv 1 \pmod{N} \tag{3.24}$$

il piú piccolo valore di r con questa proprietá viene anche detto ordine moltiplicativo di a modulo N[15] e corrisponde al valore di $\varphi(N)$, la funzione di Eulero, che associa ad ogni intero positivo N il numero degli interi positivi compresi tra 1 e n coprimi con esso. La (3.24) mette in evidenza che r é proprio il periodo della $f(x)$ che stiamo cercando, infatti:

$$f(x+r) = a^{x+r} \pmod{N} = a^x a^r \pmod{N} = a^x \pmod{N} = f(x). \tag{3.25}$$

quando r é pari e $a^{\frac{r}{2}} \not\equiv -1 \pmod{N}$, allora vale:

$$a^r \equiv 1 \pmod{N} \rightarrow a^r - 1 \equiv 0 \pmod{N} \rightarrow (a^{\frac{r}{2}})^2 - 1 \equiv 0 \pmod{N} \tag{3.26}$$

da cui

$$(a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1) \equiv 0 \pmod{N} \tag{3.27}$$

chiamando quindi $\alpha = (a^{\frac{r}{2}} - 1)$ e $\beta = (a^{\frac{r}{2}} + 1)$ si ottiene che N divide il prodotto $\alpha\beta$; per cui, a meno di multipli di N, α e β contengono la scomposizione in fattori primi di N. In particolare, considerando il caso piú semplice N = pq, con p,q primi si ha:

$$\begin{aligned} p &= MCD(a^{\frac{r}{2}} - 1, N) \\ q &= MCD(a^{\frac{r}{2}} + 1, N). \end{aligned} \tag{3.28}$$

Nel caso particolare in cui N = pq, la funzione di Eulero $\varphi(N) = (p-1)(q-1)$ e il nucleo centrale di RSA sfrutta questa proprietá per generare a partire da N una chiave pubblica (a, N) condivisa apertamente ed una chiave privata (p,q) nota esclusivamente al possessore del messaggio, necessarie rispettivamente per la codifica e la decodifica dell'informazione. L'efficacia dell'algoritmo risiede nel fatto che, per ottenere la chiave di decodifica, occorre determinare p e q o analogamente $\varphi(N)$, scomponendo N oppure risolvendo (3.24), ma entrambi i problemi risultano intrattabili per un normale computer quando il numero di cifre di N diventa abbastanza elevato.

Per applicare l'algoritmo di Shor consideriamo lo stato fiduciario

$$|\psi_0\rangle = |0^{\otimes n}0\rangle \tag{3.29}$$

e applichiamo un gate di Hadamard ai primi n qubits per ottenere:

$$|\psi_1\rangle = (H^{\otimes n}|0^{\otimes n}\rangle)|0\rangle = \left(\frac{1}{2^{\frac{n}{2}}}\sum_{x=0}^{2^n-1}|x\rangle\right)|0\rangle; \tag{3.30}$$

A questo punto applichiamo la trasformazione U_f definita nella sezione (3.1) per ottenere nel registro una sovrapposizione di tutti i valori di f:

$$|\psi_2\rangle = \frac{1}{2^{\frac{n}{2}}}\sum_{x=0}^{2^n-1}|x\rangle|f(x)\rangle. \tag{3.31}$$

E sfruttando il parallelismo quantistico si ottiene che f(x) viene valutata per tutti i valori di x in un solo passaggio; tuttavia in questo modo i due registri diventano entangled e non risulta possibile accedere a tutti i valori di f con una sola misura. Supponiamo di effettuare a questo punto una misura del secondo registro e ottenere il valore $f(x_0)$, la funzione d'onda del sistema quindi diventa:

$$|\psi_2\rangle^M = \frac{1}{\sqrt{m}}\sum_{j=0}^{m-1}|x_0 + jr\rangle|f(x_0)\rangle \qquad (0 \leq x_0 < r-1) \tag{3.32}$$

dove r é il periodo della funzione f(x) e $m = \frac{2^n}{r}$ é il numero di valori di x per cui si ha $f(x) = f(x_0)$; per semplicitá ammettiamo che il periodo della funzione divida esattamente il numero di bit 2^n su cui viene valutata, il caso generale presenta qualche complicazione aggiuntiva ma ha le stesse proprietá. Dopo il collasso in (3.32) i due stati risultano separabili e l'applicazione della QFT (3.12) al primo registro $|\psi_3\rangle = \frac{1}{\sqrt{m}}\sum_{j=0}^{m-1}|x_0 + jr\rangle$

porta a:

$$\begin{aligned}|\tilde{\psi}_3\rangle &= \frac{1}{\sqrt{m}\sqrt{r}}\sum_{k=0}^{r-1}\sum_{j=0}^{m-1} e^{\frac{2\pi\imath k}{r}(x_0+jr)}|k\rangle = \\ &= \frac{1}{\sqrt{m}\sqrt{r}}\sum_{k=0}^{r-1} e^{\frac{2\pi\imath k x_0}{r}}\sum_{j=0}^{m-1} e^{2\pi\imath kj}|k\rangle = \\ &= \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{\frac{2\pi\imath k x_0}{r}}\frac{1}{\sqrt{m}}\sum_{j=0}^{m-1} e^{2\pi\imath kj}|k\rangle = \\ &= \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{\frac{2\pi\imath k x_0}{r}}|k\frac{2^n}{r}\rangle.\end{aligned} \tag{3.33}$$

a questo punto l'interferenza quantistica ha selezionato solo un certo tipo di frequenze e una misura della funzione d'onda $|\tilde{\psi}_3\rangle$ fornisce con uguale probabilitá uno dei possibili valori $\frac{k2^n}{r}$ con k=0,1,$\cdots$, r-1; per cui, indicando con c il valore misurato si ha $c = \frac{k2^n}{r}$ e, considerando che sia k che $m = \frac{2^n}{r}$ sono numeri interi, vale $\frac{c}{2^n} = \frac{\lambda}{r}$, con λ intero. Di conseguenza, quando λ e r non hanno fattori in comune (per la teoria dei numeri questo avviene con una probabilitá $\geq \frac{1}{loglogr}$) é possibile ridurre $\frac{c}{2^n}$ ad una frazione irriducibile per ottenere simultaneamente sia λ che r; altrimenti l'algoritmo fallisce e deve essere ripetuto.

Consideriamo la funzione $f(x) = \frac{1}{2}(cos(\pi x) + 1)$ e proviamo a determinarne il periodo usando l'algoritmo di Shor e un registro a n=3 qubit per la variabile d'ingresso x. Partendo dallo stato fiduciario $|000\rangle|0\rangle$ e costruendo (3.31) si ottiene:

$$|\psi\rangle = \frac{1}{\sqrt{8}}(|0\rangle|f(0)\rangle+|1\rangle|f(1)\rangle+|2\rangle|f(2)\rangle+|3\rangle|f(3)\rangle+|4\rangle|f(4)\rangle+|5\rangle|f(5)\rangle+|6\rangle|f(6)\rangle+|7\rangle|f(7)\rangle). \tag{3.34}$$

é banale vedere che, per costruzione, $f(x) = 0$ quando x é dispari e $f(x) = 1$ quando x é pari; il suo periodo quindi é r=2. Supponiamo ora di misurare il secondo registro ed ottenere come risultato 0, la funzione d'onda totale collassa quindi nello stato

$$|\psi\rangle^M = \frac{1}{2}(|1\rangle + |3\rangle + |5\rangle + |7\rangle)|0\rangle, \tag{3.35}$$

e applicando la QFT, dalla (3.33) la funzione d'onda del primo registro diventa:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |4\rangle) \tag{3.36}$$

per cui, misurandone lo stato, si puó ottenere $|0\rangle$ o $|4\rangle$ con eguale probabilitá. Nel primo caso non si é in grado di determinare il periodo r e l'algoritmo deve essere ripetuto; invece nel secondo caso si ottiene $\frac{c}{2^n} = \frac{4}{2^3} = \frac{1}{2} = \frac{\lambda}{r} \rightarrow r = 2$, risolvendo cosí il problema.

Capitolo 4

Postfazione

"Uno degli strumenti piú importanti della fisica é il cestino della carta straccia."
– Richard P. Feynman

4.1 Qualche conclusione

Lo sviluppo di algoritmi quantistici come quello di Deutsch o quello di Shor ha portato all'introduzione di una nuova classe di complessitá, denotata con $\boldsymbol{BQP}$, che contiene tutti i problemi che sono risolvibili efficientemente da una macchina quantistica[16]; si spera che una piú lucida comprensione delle meccaniche che regolano algoritmi, gates e registri quantistici possa permetterci di stabilire con precisione dove si colloca questa classe rispetto alle due giá definite in precedenza.

In questo elaborato ho voluto trattare le particolaritá e le potenzialitá degli algoritmi quantistici per analizzare ed approfondire tematiche piuttosto attuali e per delineare un confronto fra quelli che possono essere definiti il computer di ieri e il computer del domani. Abbiamo visto la potenza di calcolo e i limiti intrinseci di una MdT, sviluppando poi una trattazione quantistica per provare ad oltrepassare questi limiti, riuscendoci anche talvolta. I risultati ottenuti sembrano inequivocabili: una macchina quantistica é in grado non solo di replicare ogni tipo di macchina classica, bensí riesce a risolvere certi problemi molto piú velocemente! Esiste peró un fondamentale problema, legato alla natura intrinsecamente probabilistica della meccanica quantistica. Grazie ai fenomeni di interferenza quantistica un qubit di un registro puó trovarsi in una qualsiasi sovrapposizione di $|0\rangle$ e $|1\rangle$ come in (2.23), tuttavia una misura dello stato del qubit fornirá il valore $|0\rangle$ o il valore $|1\rangle$ con una determinata probabilitá, per cui risulta necessario

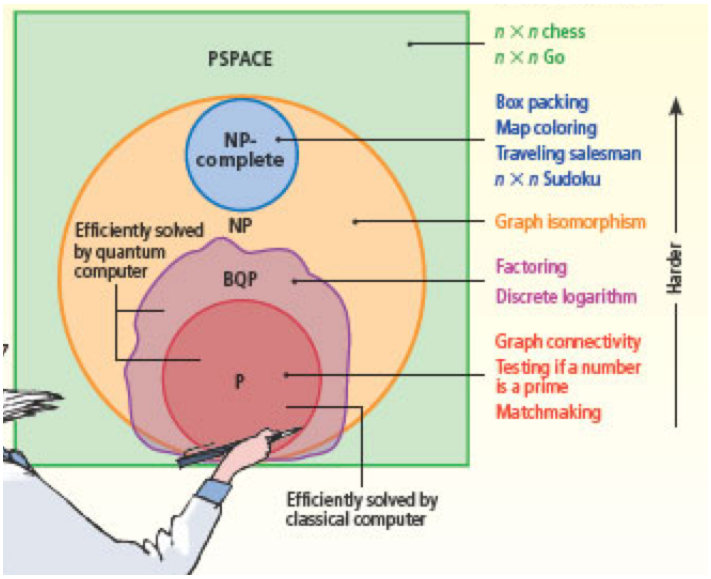

Figura 4.1: Classi di complessitá. (Adattata da Scientific American)

sviluppare tecniche di risoluzione specifiche diverse per ogni problema. Inoltre vi sono alcuni problemi di carattere applicativo, come ad esempio la scelta del miglior sistema adatto a rappresentare i qubits e i fenomeni di decoerenza quantistica dovuti all'interazione del registro con l'ambiente circostante. Per M.Nielsen e I.Chuang confrontare l'ambito classico con quello quantistico é un po' come paragonare fra loro delle mele e delle arance[3], tuttavia solo ponendo la MdT classica come termine di riferimento si riesce a capire quali sono i vantaggi di una macchina quantistica. Ad oggi qualche computer quantistico é in costruzione e uno dei prototipi D-Wave con un processore a 512 qubits é stato acquistato dalla NASA in collaborazione con Google. Al di lá delle possibili agghiaccianti applicazioni nel ramo della crittografia, si pensa che i fenomeni quantistici possano essere sfruttati per realizzare macchine in grado di apprendere, incredibilmente utili nella risoluzione di problemi complessi come il riconoscimento di un immagine o di un discorso, la traduzione linguistica o l'analisi genomica[17].

4.2 Ringraziamenti

Vorrei ringraziare innanzitutto la mia relatrice, la prof.ssa Elisa Ercolessi, per la cortesia, la disponibilitá e la professionalitá dimostrate nel periodo di lavoro insieme; un

affettuoso ringraziamento va anche a tutta la mia famiglia per l'aiuto e la fiducia che mi hanno sempre dimostrato. Merita una menzione speciale la famiglia dei Sempreverde, in particolare Francesco, Martina e Rosa con i quali ho condiviso due anni di coinquilinaggio fantastici. Devo un doveroso ringraziamento anche a Giuseppe per l'aiuto che mi ha dato nei momenti di crisi con LaTeX. Un ultimo pensiero va a tutte le persone che mi hanno supportato ed ispirato in questo percorso complicato ed avvincente: a loro va il mio piú sincero ringraziamento.

Bibliografia

[1] Richard P. Feynman, *Quantum mechanical computers, 1985*

[2] Alan Turing, *On computable numbers, with an application to the Entscheidungsproblem, 1936*

[3] Michael A. Nielsen, Isaac L. Chuang, *Quantum Computation and Quantum Information, 2000*

[4] David J. Griffiths, *Introduction to Quantum Mechanics, 2005*

[5] N. David Mermin, *Is the moon there where nobody looks?, Physics Today 1985*

[6] G.Benenti, G.Casati, G.Strini, *Principles of Quantum Computation, 2004*

[7] A. Einstein, B. Podolsky, N.Rosen, *Can quantum-mechanical description of physical Reality be considered complete?, 1935*

[8] Charles Kime, *Reti logiche*, Pearson, 2002

[9] J.Millman, A.Grabel, *Microelettronica*

[10] W.K. Wootters, W. H. Zurek, *Nature 299, 802(1982)*

[11] John von Neumann, *First Draft of a Report on the EDVAC, 1945*

[12] David Deutsch, Richard Jozsa *Rapid Solution of Problems by Quantum Computation, 1992*

[13] P. W. Shor *Algorithms for quantum computation: Discrete log and factoring, 1994*

[14] R. Crandall, C.Pomerance *Prime Numbers: A Computational perspective, 2001*

[15] Philip Kaye, Raymond Laflamme, Michele Mosca *An introduction to quantum computing*

[16] Scott Aaronson, *6.845 Quantum Complexity Theory, Fall 2010. (Massachusetts Institute of Technology: MIT OpenCourseWare)*

[17] Simon Benjamin, *oxfordquantum.org/stories/, 2014*