UNIVERSITÀ DEGLI STUDI DI ROMA TOR VERGATA

Corso di Laurea in Fisica

Percorso di Eccellenza Triennale

Computazione Quantistica e Algoritmo di Shor

Relatore
prof. Gaetano Salina

Laureando
Mattia Marzi

Dicembre 2020

Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy.

Richard Feynman

Ringraziamenti

Un grazie sentito al Professore Gaetano Salina per avermi supervisionato e guidato durante questo percorso. È doveroso inoltre ringraziare la Professoressa Alessandra Di Pierro dell'Univeristà di Verona Strada le Grazie per i suoi preziosi appunti sulla Computazione Quantistica [1] e il Professore Samuel J. Lomonaco Jr. dell'Univeristà del Maryland per il suo articolo [2] a cui questo lavoro è fortemente ispirato.

Sommario

In questo lavoro si intende fornire al lettore gli elementi fondamentali per comprendere le fondamenta alla base della computazione quantistica, mettendone in luce l'importanza e l'indispensabilità nel prossimo futuro e accompagnandolo fino alla scrittura di un vero e proprio algoritmo quantistico (l'algoritmo di Shor). A tale scopo, l'elaborato inizia con la presentazione delle proprietà generali degli spazi vettoriali in campo complesso, inquadrando i qubits come vettori in uno spazio di Hilbert bi-dimensionale. Si presentano poi i concetti fondamentali della meccanica quantistica che saranno utili per capire il funzionamento di alcuni circuiti e per comprendere il perché un computer quantistico possa risolvere problemi classicamente reputati irrisolvibili. Si procede poi con la descrizione delle porte logiche quantistiche fondamentali e della loro combinazione a formare circuiti semplici, fino ad arrivare all'implementazione circuitale della Trasformata di Fourier Quantistica (alla base di tutti gli algoritmi quantistici esponenzialmente più efficienti delle controparti classiche). Si può quindi procedere alla presentazione dell'algoritmo di Shor, stimando le sue probabilità di successo, il tempo necessario per la sua esecuzione e, infine, scrivendo il codice per implementarlo in un computer quantistico.

Indice

Elenco delle figure VII

Elenco dei listati VIII

1 Introduzione 1

2 Introduzione Matematica 2
- 2.1 Spazi vettoriali . . . 2
  - 2.1.1 Norma . . . 2
  - 2.1.2 Basi . . . 2
- 2.2 Prodotto tensore . . . 3
- 2.3 Autovalori e autovettori . . . 3
- 2.4 Operatori aggiunti . . . 4
  - 2.4.1 Operatori Hermitiani . . . 4
  - 2.4.2 Operatori unitari . . . 5
- 2.5 Rappresentazione di operatori . . . 5

3 Introduzione Fisica 7
- 3.1 I postulati della meccanica quantistica . . . 7
  - 3.1.1 Postulato 1 . . . 7
  - 3.1.2 Postulato 2 . . . 7
  - 3.1.3 Postulato 3 . . . 8
  - 3.1.4 Postulato 4 . . . 8
  - 3.1.5 Postulato 5 . . . 8
  - 3.1.6 Principio di sovrapposizione degli stati . . . 8
  - 3.1.7 Il problema della misurazione quantistica . . . 8
  - 3.1.8 Stati *entangled* . . . 9

4 Porte logiche e circuiti quantistici 10
- 4.1 Quantum bits . . . 10

4.2 Porte logiche quantistiche a un qubit 10
4.3 Porte logiche quantistiche a più qubits 11
4.4 Teorema no-cloning 12
4.5 Un'interessante applicazione... Il teletrasporto quantistico 13

5 Complessità computazionale e Trasformata di Fourier Quantistica 16
5.1 Complessità computazionale classica 16
5.2 Complessità computazionale quantistica 16
5.3 La Trasformata di Fourier Quantistica 17
5.4 Implementazione circuitale della Trasformata di Fourier Quantistica . . . 19

6 Algoritmo di Shor 22
6.1 Introduzione 22
6.2 Una panoramica dell'algoritmo 23
6.3 La parte quantistica dell'algoritmo di Shor 24
6.3.1 Introduzione e stima delle probabilità 24
6.3.2 Digressione sulle frazioni continue 28
6.3.3 Parte finale dello *Step 2* 30

7 Implementazione dell'Algoritmo di Shor 32
7.1 Preparazione dell'ambiente di sviluppo 32
7.2 Introduzione al problema 33
7.3 Quantum Phase Estimation 36
7.4 Implementazione circuitale 38

Riferimenti bibliografici 46

Elenco delle figure

4.1 Rappresentazione circuitale della porta CNOT 12
4.2 Rappresentazione circuitale della porta *controlled-U* 12
4.3 Rappresentazione circuitale per il teletrasporto di un qubit 14
4.4 Circuito per la creazione dello stato $|\beta_{00}\rangle$ 14

5.1 Rappresentazione circuitale della trasformata di Fourier quantistica 20

7.1 Rappresentazione visiva del periodo 34
7.2 Rappresentazione circuitale dell'algoritmo quantistico di stima della fase . 36
7.3 Rappresentazione circuitale dell'algoritmo di Shor 39
7.4 Rappresentazione circuitale dell'antitrasformata di Fourier quantistica che agisce su 4 qubits 39
7.5 Circuito utilizzato per l'implementazione dell'algoritmo di Shor 40
7.6 Istogramma rappresentativo delle misure effettuate dal simulatore quantistico 45
7.7 Tabelle riassuntive dei risultati ottenuti applicando l'algoritmo ripetutamente 45

Elenco dei listati

7.1 Rappresentazione visiva del Periodo 33
7.2 Implementazione dell'algoritmo di Shor 41
7.3 Iterazioni multiple dell'algoritmo 43

Capitolo 1

Introduzione

La computazione si è sempre basata sul concetto di Macchina di Turing Universale: un dispositivo meccanico con memoria potenzialmente infinita che obbedisce a leggi della fisica classica.

Tuttavia, se la legge di Moore, che prevede un raddoppio del numero di transistori nei microprocessori (con conseguente loro miniaturizzazione) ogni due anni, dovesse continuare a valere, allora l'influenza della meccanica quantistica sulla computazione diventa sempre più necessaria in quanto la meccanica classica non può spiegare i fenomeni fisici che avvengono a livello microscopico.

Già R. Feynman, negli anni '80, dimostrò come un "simulatore quantistico universale" potesse effettuare simulazioni di certi fenomeni fisici in modo estremamente più efficiente di una Macchina di Turing classica (che incorre in un rallentamento esponenziale delle prestazioni). Tutto ciò portò ad una ridefinizione del concetto di "trattabilità". Infatti, in prima approssimazione, possiamo definire trattabili quei problemi che possono essere risolti in un tempo polinomiale. Uno degli esempi più importanti di problemi trattabili con un computer quantistico, ma non classicamente, è costituito dalla fattorizzazione dei numeri primi.

Nel 1994 P. Shor (§6) sviluppò un algoritmo quantistico in grado di trattare il problema in un tempo polinomiale.

Per capire il funzionamento di un computer quantistico è necessario fare una breve introduzione matematica che ci porterà a formulare il *principio di sovrapposizione degli stati* (§3.1.6), il *principio di misurazione* (§3.1.7) e il fenomeno dell'*entanglement* (§3.1.8).

Capitolo 2

Introduzione Matematica

Come la computazione classica si basa sul bit, quella quantistica si basa sul qubit. Esso è un oggetto matematico astratto che vive nello spazio vettoriale complesso $\mathbb{C}^2$

2.1 Spazi vettoriali

2.1.1 Norma

Lo spazio vettoriale complesso $\mathbb{C}^2$ è l'insieme dei vettori colonna

$$v = \begin{pmatrix} a \\ b \end{pmatrix}, \tag{2.1}$$

con $a, b \in \mathbb{C}$. Utilizzando la *notazione di Dirac* si può indicare un vettore colonna con $|v\rangle$ (o *ket*). È possibile definire l'**aggiunto** di un vettore (o $\langle v|$, *bra*) come:

$$v^\dagger \coloneqq \begin{pmatrix} a^* & b^* \end{pmatrix}. \tag{2.2}$$

Si ha quindi che:

$$\|v\| = (v, v) = \langle v, v \rangle = \begin{pmatrix} a^* & b^* \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = |a|^2 + |b|^2. \tag{2.3}$$

2.1.2 Basi

Un insieme di vettori $v_i \in \mathbb{C}^2 \quad | \quad i = 1,2,...,k$ si dice **linearmente indipendente** se:

$$a_1 v_1 + a_2 v_2 + ... + a_k v_k = 0 \quad con \quad a_i \in \mathbb{R} \Leftrightarrow a_i = 0 \quad con \quad i = 1,2,...,k. \tag{2.4}$$

Si definisce **base** di $\mathbb{C}^2$ un qualsiasi insieme di vettori linearmente indipendenti tali che ogni vettore in $\mathbb{C}^2$ si può esprimere come combinazione dei vettori della base. La base

canonica è la base data dai vettori:

$$|0\rangle \coloneqq \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle \coloneqq \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{2.5}$$

Una base si dice ortonormale se:

$$\langle i, j \rangle = \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}. \tag{2.6}$$

Uno spazio vettoriale complesso in cui è definito un prodotto scalare viene detto **spazio di Hilbert**.

2.2 Prodotto tensore

Il prodotto tensore è quell'operazione che permette di combinare spazi vettoriali per ottenere spazi vettoriali più grandi. Dati due spazi vettoriali $\mathbb{C}^k$ e $\mathbb{C}^l$, si definisce **prodotto tensore** la funzione

$$\otimes : \mathbb{C}^k \times \mathbb{C}^l \to \mathbb{C}^{kl} \tag{2.7}$$

con

$$v \otimes w = \begin{pmatrix} v_1 w \\ v_2 w \\ \vdots \\ v_k w \end{pmatrix}. \tag{2.8}$$

Si può anche definire il prodotto tensore tra matrici nel modo seguente:

$$M : \mathbb{C}^k \mapsto \mathbb{C}^k \quad e \quad N : \mathbb{C}^l \mapsto \mathbb{C}^l, \tag{2.9}$$

$$M \otimes N : \mathbb{C}^{kl} \mapsto \mathbb{C}^{kl}, \tag{2.10}$$

$$M \otimes N = \begin{pmatrix} M_{11}N & M_{12}N & \dots & M_{1k}N \\ M_{21}N & M_{22}N & \dots & M_{2k}N \\ \vdots & \vdots & \vdots & \vdots \\ M_{k1}N & M_{k2}N & \dots & M_{kk}N \end{pmatrix}. \tag{2.11}$$

2.3 Autovalori e autovettori

Dato uno spazio vettoriale V e un operatore lineare $L : V \mapsto V$, si definisce **autovettore** di L un vettore non nullo $|v\rangle$ tale che

$$L\,|v\rangle = v\,|v\rangle\,, \tag{2.12}$$

con $v \in \mathbb{C}$ detto **autovalore**. Una base di autovettori con lo stesso autovalore prende il nome di **autospazio**. Nel caso in cui tale base contenga più di un autovettore linearmente indipendente allora l'autospazio si dice **degenere**.

2.4 Operatori aggiunti

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

$$(|v\rangle, L|w\rangle) = (L^\dagger|v\rangle, |w\rangle), \tag{2.13}$$

dove $L^\dagger$ si dice **aggiunto** di L.

2.4.1 Operatori Hermitiani

Un operatore L si dice **Hermitiano** se

$$L^\dagger = L. \tag{2.14}$$

La caratteristica principale che fa si, come vedremo in §3.1.2, che tutte le grandezze fisiche siano rappresentabili con operatori Hermitiani è il fatto che tali operatori hanno autovalori reali. Un'importante classe di operatori Hermitiani è costituita dagli operatori di proiezione. Dato uno spazio vettoriale V di dimensione d, si consideri un sottospazio di V di dimensione k. Se $|1\rangle, ..., |k\rangle$ è una base ortonormale del sottospazio, si definisce **operatore di proiezione** sul sottospazio:

$$P \coloneqq \sum_{i=1}^{k} |i\rangle\langle i|, \tag{2.15}$$

dove, l'operazione $|i\rangle\langle i|$, prende il nome di **prodotto esterno**:
dati $|v\rangle \in V \subseteq \mathbb{C}^k$ e $|w\rangle \in W \subseteq \mathbb{C}^l$ definiamo l'operatore lineare $|w\rangle\langle v| : V \mapsto W$ che associa ad ogni vettore $|v'\rangle \in V$ un vettore in W dato da:

$$|w\rangle\langle v|(|v'\rangle) \coloneqq |w\rangle(\langle v, v'\rangle). \tag{2.16}$$

Vale dunque: $|w\rangle\langle v| : V \mapsto W$. Notiamo infine che per una base ortonormale vale la cosiddetta *relazione di completezza*:

$$\sum_i |i\rangle\langle i| = I \tag{2.17}$$

(proprietà che si può verificare prendendo una qualsiasi base ortonormale e facendo il prodotto righe per colonne $|i\rangle\langle i|$).

Un operatore di proiezione, oltre ad essere Hermitiano, gode anche della proprietà di idempotenza:

$$P^n = P^2 = P. \tag{2.18}$$

2.4.2 Operatori unitari

Un operatore U si dice **unitario** se vale la relazione

$$U^\dagger U = \mathbb{1}. \tag{2.19}$$

2.5 Rappresentazione di operatori

Dato un vettore $v \in \mathbb{C}^2$, si dice **duale** di u la funzione $L_v : \mathbb{C}^2 \mapsto \mathbb{C}$ definita da $L_v(w) = (v, w) = v^\dagger w$. Si è soliti identificare L_v con $v^\dagger$ e nella notazione di Dirac il duale di un vettore $|\psi\rangle$ è indicato con $\langle\psi|$ (il *bra* è il duale del *ket*).

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

$$L(a_1 v_1 + a_2 v_2) = a_1 L(v_1) + a_2 L(v_2). \tag{2.20}$$

Un operatore di questo tipo è sempre rappresentabile con una matrice rispetto ad una data base.

Esempio 2.1

La matrice di rappresentazione di una funzione lineare L nella base $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ e $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ si definisce a partire da:

$$\begin{cases} L|0\rangle = a_{11}|0\rangle + a_{21}|1\rangle \\ L|1\rangle = a_{12}|0\rangle + a_{22}|1\rangle \end{cases}. \tag{2.21}$$

Possiamo quindi scrivere:

$$L = a_{11}|0\rangle\langle 0| + a_{21}|1\rangle\langle 0| + a_{12}|0\rangle\langle 1| + a_{22}|1\rangle\langle 1|, \tag{2.22}$$

e quindi la matrice rappresentativa di L nella base data da $|0\rangle$ e $|1\rangle$ è data da:

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}. \tag{2.23}$$

Definiamo infine matrice **unitaria** una matrice A $n \times n$ tale che:

$$A^\dagger = A^{-1} \tag{2.24}$$

e matrice **hermitiana** una matrice $n \times n$ tale che:

$$A^\dagger = A, \tag{2.25}$$

con $A^\dagger$ detta matrice **aggiunta** e con $A^\dagger = (A^T)^*$, $(A^T)_{ij} = (A)_{ij}$ e $(A^*)_{ij} = (A)^*_{ij}$. Notiamo che una matrice conserva il prodotto scalare e la norma se e solo se è unitaria. Infatti, data una matrice U e due vettori $v, w \in \mathbb{C}^n$, allora:

$$\begin{aligned} \langle Uv, Uw\rangle &= \\ \langle U^\dagger Uv, w\rangle &= \\ \langle U^{-1}Uv, w\rangle &= \\ \langle v, w\rangle, & \end{aligned}$$

dove la prima uguaglianza è giustificata dalla (2.13) e la seconda dalla (2.19). Un operatore hermitiano e uno unitario hanno come rappresentazione matriciale rispettivamente una matrice hermitiana e una matrice unitaria se e solo se la base di rappresentazione è ortonormale (per maggiori dettagli si faccia riferimento a [3]).

Capitolo 3

Introduzione Fisica

3.1 I postulati della meccanica quantistica

W. Pauli: *Non mi fu risparmiato lo shock che ogni fisco abituato al modo di pensare classico subiva quando sentiva parlare per la prima volta del postulato fondamentale della meccanica quantistica di Bohr* [4]
N. Bohr: *Quelli che non rimangono scioccati la prima volta che si imbattono nella meccanica quantistica non possono averla compresa* [5]

La parola "postulato" deriva dal latino *postulatum* e significa: ciò che è richiesto, proposizione che senza essere evidente o dimostrata si assume per una teoria. Al contrario, "assioma", viene da *axioma* e indica proposizioni così evidenti che si assumano vere senza dimostrarle. Questa distinzione etimologica si è persa nella lingua moderna ma è utile per inquadrare il ruolo che hanno i postulati della meccanica quantistica.

3.1.1 Postulato 1

Ad ogni sistema fisico è associato uno spazio di Hilbert e gli stati del sistema sono i vettori di tale spazio, definiti a meno di una costante complessa moltiplicativa.

3.1.2 Postulato 2

Ad ogni grandezza fisica misurabile è associato un operatore lineare Hermitiano. L'insieme dei possibili risultati della misura è costituito dallo spettro degli autovalori dell'operatore. tali valori devono essere reali e quindi l'operatore Hermitiano

3.1.3 Postulato 3

La probabilità di ottenere da una certa misura un determinato autovalore a_i è data da

$$|\langle a_i, \psi \rangle|^2. \tag{3.1}$$

Da qui la necessità che gli stati $|\psi\rangle$ di uno spazio di Hilbert siano normalizzati

3.1.4 Postulato 4

Dopo aver effettuato una misura lo stato collassa nell'autovettore corrispondente all'autovalore ottenuto. Nel caso in cui sia associato all'autovalore un autospazio degenere, allora lo stato collassa in tale autospazio.

3.1.5 Postulato 5

Lo stato evolve nel tempo secondo l'**Equazione di Schrödinger**

$$i\hbar \frac{\partial |\psi(t)\rangle}{\partial t} = \hat{H}(t) |\psi(t)\rangle . \tag{3.2}$$

3.1.6 Principio di sovrapposizione degli stati

Proiettando un *ket* su un particolare spazio di coordinate (come quello della posizione) si ottiene una funzione d'onda:

$$\psi(x,t) = \langle x, \psi(t) \rangle . \tag{3.3}$$

Due o più funzioni d'onda che differiscono solo per una costante moltiplicativa descrivono lo stesso stato quantistico (questo perché i rispettivi vettori dello spazio di Hilbert sono linearmente dipendenti). Se uno stato può essere descritto sia da una funzione d'onda ψ_1 che da una funzione d'onda ψ_2, allora lo stato può essere descritto nel modo più generale possibile da una qualsiasi combinazione lineare normalizzata dei due stati:

$$\psi = c_1\psi_1 + c_2\psi_2 \quad con \quad |c_1|^2 + |c_2|^2 = 1. \tag{3.4}$$

Tale principio prende il nome di *principio di sovrapposizione*

3.1.7 Il problema della misurazione quantistica

Secondo la formalizzazione di von Neumann il processo di misurazione avviene in due fasi: nella prima fase l'operatore $\hat{H}$ agisce sullo stato formandone uno *entangled* (§3.1.8). In una seconda fase avviene la *riduzione di stato*, ovvero il salto dallo stato *entangled* ad uno degli autovettori dell'osservabile. Tale riduzione può essere interpretata come un

disturbo della misura sullo stato su cui si vuole effettuare la misura ed è completamente non deterministica: non è possibile prevedere il valore misurato non per un limite dell'osservatore, ma per una caratteristica intrinseca del processo. L'*interpretazione statistica di Born* permette di stabilire la distribuzione probabilistica dei risultati come indicato nel Postulato 3. L'assenza di determinismo del processo è sempre stata rifiutata da A. Einstein e portò alla formulazione della celebre frase: *non credo per un solo istante che Lui giochi a dadi* [6]

3.1.8 Stati *entangled*

Un **registro quantistico** di n *ket* è un elemento dello spazio di Hilbert dato da

$$|i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_n\rangle \,, \tag{3.5}$$

dove il simbolo $\otimes$ indica il prodotto tensore (§2.2). Spesso, per alleggerire la notazione, si utilizza la scrittura compatta:

$$|i_1 i_2 \dots i_n\rangle \,. \tag{3.6}$$

Un registro quantistico a n vettori, per il principio di sovrapposizione, può essere espresso come somma di k registri quantistici a n vettori. Tuttavia, così facendo, non sempre è possibile esprimere il registro quantistico come prodotto tensore di n vettori. Vettori di questo tipo sono detti **entangled**. Vettori di questo tipo sono strettamente connessi l'uno all'altro e la misurazione di uno dei due ci dà informazioni anche sugli altri.

Esempio 3.1
Lo stato

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{3.7}$$

è uno stato entangled. Infatti non esistono a_1, b_1, a_2, b_2 tali che

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = (a_1 |0\rangle + b_1 |1\rangle) \otimes (a_2 |0\rangle + b_2 |1\rangle). \tag{3.8}$$

Inoltre, una misura sul primo stato porta ad una conoscenza esatta anche del secondo. Infatti, misurando λ_1 (dove con λ_1 si intende l'autovaslore dell'autostato $|1\rangle$) sul primo stato si avrebbe il collasso

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \to |11\rangle \,; \tag{3.9}$$

dunque anche il secondo stato collasserebbe nell'autostato $|1\rangle$.

Capitolo 4

Porte logiche e circuiti quantistici

4.1 Quantum bits

La computazione classica si basa sul concetto fondamentale di *bit* che può assumere lo stato 0 o lo stato 1. L'analogo quantistico è costituito dal *quantum bit* che può essere visto come un vettore unitario di uno spazio di Hilbert bi-dimensionale in cui i vettori $|0\rangle$ e $|1\rangle$ costituiscono una base ortonormale che viene detta **base computazionale standard**. Quindi un qubit generico $|\psi\rangle$ può essere rappresentato mediante l'equazione:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{4.1}$$

con $\alpha, \beta \in \mathbb{C}$ e $|\alpha|^2 + |\beta|^2 = 1$.

4.2 Porte logiche quantistiche a un qubit

Nella computazione classica esiste una sola porta logica a singolo bit non banale, la porta `NOT`, che implementa l'operazione logica di negazione:

$$0 \to 1 \tag{4.2}$$

$$1 \to 0. \tag{4.3}$$

Equivalentemente, l'analogo quantistico della porta `NOT` dovrebbe effettuare l'operazione:

$$\alpha\,|0\rangle + \beta\,|1\rangle \to \beta\,|0\rangle + \alpha\,|1\rangle \tag{4.4}$$

La matrice che permette questo passaggio viene chiamata X ed è definita come segue:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{4.5}$$

Dal momento che X mantiene la norma dei vettori a cui è applicata, allora X è unitaria (§2.5). Contrariamente al caso classico, nel caso quantistico ci sono più operazioni non banali. Per esempio, definiamo la porta Z

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{4.6}$$

che agisce solamente sulla componente $|1\rangle$ cambiandone il segno, e la *porta di Hadamard*

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{4.7}$$

che trasforma uno stato della base computazionale standard in una sovrapposizione di stati che risulta essere 0 o 1 con uguale probabilità effettuandone una misura. Infatti:

$$H \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \tag{4.8}$$

$$H \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{4.9}$$

$$\tag{4.10}$$

Notiamo che le matrici X, Z, insieme alla matrice

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \tag{4.11}$$

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

4.3 Porte logiche quantistiche a più qubits

Le più importanti porte logiche classiche a più bits sono: `AND`, `OR`, `XOR`, `NAND` e `NOR`. Le porte `NOT` e `AND` formano un **insieme universale** (tutte le altre possono essere scritte come combinazione di queste due).
L'analogo quantistico di `XOR` è costiuito dalla porta `CNOT`. Essa opera su un qubit di **controllo** e su un qubit **target**. Se il controllo è 0 allora il target non viene alterato; se il controllo è 1 allora il target viene negato:

$$|00\rangle \mapsto |00\rangle\,,\ |10\rangle \mapsto |11\rangle\,,\ |01\rangle \mapsto |01\rangle\,,\ |11\rangle \mapsto |10\rangle\,. \tag{4.12}$$

. Equivalentemente si può porre:

$$|A, B\rangle \mapsto |A, B \oplus A\rangle\,, \tag{4.13}$$

dove l'operazione $\oplus$ è la *somma modulo 2*: $B \oplus A = (B + A) mod 2$ e $xmody$ dà come risultato il resto della divisione $\frac{x}{y}$. Possiamo anche pensare alla porta `CNOT` con la sua rappresentazione matriciale:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{4.14}$$

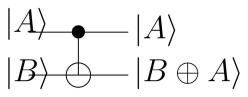

Figura 4.1. Rappresentazione circuitale della porta CNOT

Come è possibile notare da Fig. 4.1 la linea con il pallino nero indica il qubit di controllo, il target è indicato dalla linea sottostante Notiamo che il `CNOT`, essendo unitario, è invertibile (questo non è vero per le porte logiche classiche che in generale sono irreversibili). Si può aggiungere un numero maggiore di qubits di controllo e qubits target applicando a quest'ultimi l'operatore unitario U solo e soltanto nel caso in cui tutti i target siano $|1\rangle$. Una porta di questo tipo prende il nome di *controlled-U*. Prendiamo il circuito raffigurato in Fig. 4.2, assumiamo che U agisca su un singolo qubit e chiamiamo i qubits in input $|c\rangle |t\rangle$ (il primo sarà di controllo, il secondo sarà il target). Un circuito siffatto opererà la trasformazione:

$$|c\rangle |t\rangle \mapsto |c\rangle U^c |t\rangle . \tag{4.15}$$

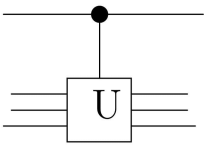

Figura 4.2. Rappresentazione circuitale della porta *controlled-U*

4.4 Teorema no-cloning

Prendendo una porta `CNOT` e ponendo come qubit di controllo un certo $|\psi\rangle$ e come qubit target $|0\rangle$, ci chiediamo se sia possibile trasformare $|0\rangle$ in `CNOT`. Nel caso in cui $|\psi\rangle$ sia

uno stato della base computazionale standard ciò è banalmente verificato. Tuttavia, se $|\psi\rangle$ si trova in una combinazione generica di stati

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

allora è impossibile trasformare il target in $|\psi\rangle$ e ciò è vero non soltanto per la porta `CNOT`, ma per qualsiasi altra porta logica a due qubit. Infatti, supponiamo per assurdo che esista una porta logica con rappresentazione matriciale A tale che

$$M\,|\psi\rangle\,|0\rangle = |\psi\rangle\,|\psi\rangle$$

per ogni $|\psi\rangle$. Dunque $\exists\,|\phi\rangle\,,|\eta\rangle$ tale che

$$M\,|\phi\rangle\,|0\rangle = |\phi\rangle\,|\phi\rangle \tag{4.16}$$

$$M\,|\eta\rangle\,|0\rangle = |\eta\rangle\,|\eta\rangle \tag{4.17}$$

con

$$1 < \langle\phi,\eta\rangle < 1. \tag{4.18}$$

Facendo il prodotto della prima della 4.16 per l'aggiunta della seconda e ricordando che M, dovendo preservare le norme, è unitaria e quindi vale (2.19) si ha:

$$\langle\phi,\eta\rangle\,\langle\phi,\eta\rangle = \langle\phi,\eta\rangle\,\langle 0{,}0\rangle = \langle\phi,\eta\rangle \tag{4.19}$$

e quindi $\langle\phi,\eta\rangle = \pm 1$ contraddicendo la (4.18). Tale teorema prende il nome di *no-cloning*

4.5 Un'interessante applicazione... Il teletrasporto quantistico

Il teletrasporto quantistico è una tecnica per trasferire l'informazione contenuta in un qubit da un punto ad un altro dello spazio attraverso la trasmissione di bits classici (senza perdere informazione come invece avverrebbe effettuando una misura del qubit con conseguente collasso).
Con riferimento alla Fig. 4.3 stabiliamo che

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

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

Dunque, i due qubits che costituiscono lo stato $|\beta_{00}\rangle$, sono entangled. Per costruire uno stato di questo tipo si può sfruttare il circuito riportato in Fig. 4.1 con $x = |0\rangle$ e $y = |0\rangle$.

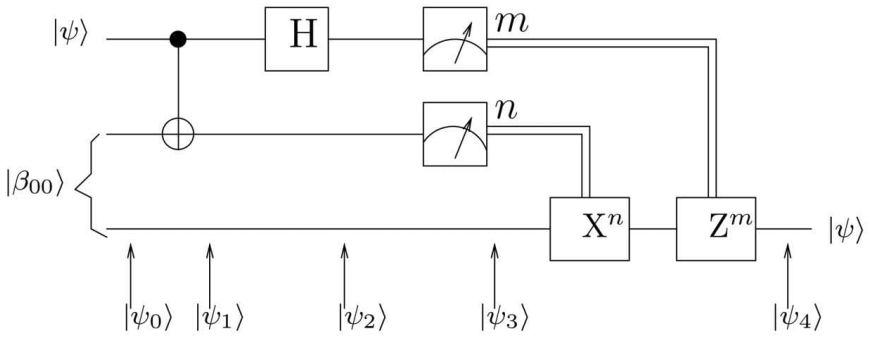

Figura 4.3. Rappresentazione circuitale per il teletrasporto di un qubit

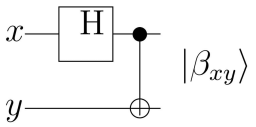

Figura 4.4. Circuito per la creazione dello stato $|\beta_{00}\rangle$

Infatti, dopo aver applicato H su x si ha:

$$x = |0\rangle \mapsto Hx = \frac{|0\rangle + |1\rangle}{\sqrt{2}}. \tag{4.20}$$

Quindi, sullo stato Hx, si misurerà λ_0 (con λ_0 autovalore associato allo stato $|1\rangle$ di un ipotetico operatore O con cui si effettua la misura) con probabilità $p = \frac{1}{2}$ (in questo caso la y viene lasciata invariata e quindi lo stato finale è $|00\rangle$) e λ_1 (autovalore associato a $|1\rangle$) con probabilità $p = \frac{1}{2}$ (in questo caso la y viene trasformata in 1 e quindi lo stato finale sarà $|11\rangle$). Quindi, uno stato che dia tali misure con le probabilità indicate, è proprio $|\beta_{00}\rangle$.

Immaginiamo che una persona, che chiamiamo Alice, voglia far conoscere lo stato $|\psi\rangle$ ad una seconda persona che chiamiamo Bob. Sia Alice che Bob possiedono un qubit della coppia entangled $|\beta_{00}\rangle$. Alice non può fare una copia del suo qubit a causa del teorema no-cloning (§4.4). Inoltre, i canali di trasmissione in suo possesso, possono trasferire solamente informazione classica. Sembrerebbe quindi che, per Alice, non sia possibile far conoscere a Bob il suo qubit senza prima farlo collassare per effetto di una misura. Vediamo tuttavia come ciò, in realtà, sia possibile, grazie alla coppia entangled in possesso sia di Alice che di Bob. Alice applica una porta `CNOT` con $|\psi\rangle$ come controllo e il qubit in

suo possesso della coppia entangled come target. Si ha:

$$|\psi\rangle\,|\beta_{00}\rangle \mapsto \frac{|000\rangle + |011\rangle}{\sqrt{2}} \quad con \quad p = |\alpha|^2 \tag{4.21}$$

$$|\psi\rangle\,|\beta_{00}\rangle \mapsto \frac{|110\rangle + |101\rangle}{\sqrt{2}} \quad con \quad p = |\beta|^2. \tag{4.22}$$

Quindi

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}[\alpha\,|0\rangle\,(|00\rangle + |11\rangle) + \beta\,|1\rangle\,(|10\rangle + |01\rangle)]. \tag{4.23}$$

Applicando Hadamard al primo qubit, da (4.8), si ha:

$$|\psi_2\rangle = \frac{1}{2}[\alpha(|0\rangle + |1\rangle)(|00\rangle + |11\rangle) + \beta(|0\rangle - |1\rangle\,(|10\rangle + |01\rangle)] = \tag{4.24}$$

$$= \frac{1}{2}[|00\rangle\,(\alpha\,|0\rangle + \beta\,|1\rangle) + |01\rangle\,(\alpha\,|1\rangle + \beta\,|0\rangle) + |10\rangle\,(\alpha\,|0\rangle - \beta\,|1\rangle) + |11\rangle\,(\alpha\,|1\rangle - \beta\,|0\rangle)]. \tag{4.25}$$

A questo punto Alice effettua una misura sul primo qubit e sul secondo qubit; si ha:

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

Comunica i due bit mn ottenuti e Bob riottiene lo stato $|\psi\rangle$ di partenza applicando le porte $X^n Z^m$. Infatti, immaginiamo, per esempio, che Alice misuri 11. Allora:

$$XZ(\alpha\,|1\rangle - \beta\,|0\rangle) = X(-\alpha\,|1\rangle - \beta\,|0\rangle) = (-\alpha\,|0\rangle - \beta\,|1\rangle) = -\,|\psi\rangle \equiv |\psi\rangle$$

dove l'ultima uguaglianza è giustificata dal Postulato 1 (§3.1.1).

Capitolo 5

Complessità computazionale e Trasformata di Fourier Quantistica

5.1 Complessità computazionale classica

Definiamo con **P** la classe dei problemi risolvibili in tempo polinomiale da una Macchina di Turing deterministica e con **BPP** tutti quei problemi per i quali esistono algoritmi risolutivi probabilistici che li risolvono in tempo polinomiale e con probabilità di errore piccola (più precisamente, è necessario che si riconosca una preposizione falsa con probabilità di successo $\geq \frac{2}{3}$ e una preposizione vera con probabilità di successo $\geq \frac{2}{3}$). La classe **BPP** contiene tutti quei problemi considerati trattabili. Infine, la classe **NP**, è definita come l'insieme dei problemi le cui ipotetiche soluzioni impiegano un tempo polinomiale per essere verificate su una macchina di Turing non deterministica (ovvero una Macchina di Turing che sceglie a caso tra le transizioni disponibili in ogni fase secondo una determinata distribuzione di probabilità). I problemi in quest'ultima classe sono considerati classicamente intrattabili.

5.2 Complessità computazionale quantistica

Grazie alla risoluzione, mediante algoritmi quantistici, di problemi ritenuti classicamente intrattabili (vedi §6) si è resa necessaria la ridefinizione di classi di trattabilità. Il corrispondente quantistico della classe **BPP** è costituito dall'insieme dei problemi risolubili in tempo polinomiale su una Macchina di Turing quantistica (**BQP**). Dal momento che una Macchina di Turing probabilistica si può simulare con una Macchina di Turing quantistica

allora vale la relazione:

$$\mathbf{P} \sqsubseteq \mathbf{BPP} \sqsubseteq \mathbf{BQP}. \tag{5.1}$$

Non ci sono prove certe che $\mathbf{BPP} \neq \mathbf{BQP}$ e quindi della superiorità del modello quantistico. Tuttavia, grazie alla tecnica nota come **Trasformata di Fourier Quantistica** è stato possibile formulare alcuni algoritmi quantistici con prestazioni esponenzialmente migliori delle controparti classiche riuscendo a risolvere alcuni problemi classicamente ritenuti intrattabili.

5.3 La Trasformata di Fourier Quantistica

La **Trasformata di Fourier Discreta** (DFT) è una trasformata che converte una collezione finita di N punti equispaziati di una funzione in una collezione di coefficienti di una combinazione lineare di sinusoidi complesse, ordinate al crescere della frequenza. poniamoci nell'intervallo $[0, 2\pi]$ e prendiamo i punti definiti dall'equazione $x_k = k\frac{2\pi}{N}$ con $k = 0, \ldots, N-1$. Dato l'insieme $G_N \coloneqq x_k \quad | \quad k = 0, \ldots, N-1$ sia definita la funzione $f : G_N \mapsto \mathbf{C}$. Dati

$$e_n(x_k) = \begin{cases} 1 & k = n \\ 0 & k \neq n \end{cases} \tag{5.2}$$

$\{e_n\}_{n=0}^{N-1}$ è una base di G_N; infatti per ogni f posso scrivere $f = \sum_{k=0}^{N-1} f(x_k)e_k$ e quindi la dimensione (ovvero il numero massimo di vettori linearmente indipendenti) di G_N è N. In tale spazio definiamo il prodotto scalare come:

$$\langle f, g \rangle = \sum_{k=0}^{N-1} f^*(x_k)g(x_k); \tag{5.3}$$

secondo tale definizione la base $\{e_n\}_{n=0}^{N-1}$ è ortonormale. Poniamo $\omega \equiv e^{\frac{2\pi i}{N}}$, $\phi_n(x_k) = e^{\frac{2\pi ink}{N}} = \omega^{nk}$, $\phi_n^*(x_k) = \omega^{-nk}$. Dimostriamo che $\{\phi_n\}_{n=0}^{N-1}$ costituisce una base di G_N. Il numero di vettori è N come vogliamo, affinché essa sia una base è sufficiente dimostrare che i vettori ϕ_n siano ortogonali: sia $0 \leq m, n \leq N-1$

$$\langle \phi_m, \phi_n \rangle = \sum_{k=0}^{N-1} \omega^{-mk}\omega nk = \sum_{k=0}^{N-1} \omega^{(n-m)k} = \begin{cases} N & m = n, \\ \frac{1-\omega^{(n-m)N}}{1-\omega^{n-m}} \end{cases} \tag{5.4}$$

dove l'ultimo caso si è ottenuto applicando la formula per le somme geometriche. Ma $\omega^{(n-m)N} = e^{\frac{2\pi i(n-m)N}{N}} = 1$. Quindi $\langle \phi_m, \phi_n \rangle = 0$ se $n \neq m$ e quindi le ϕ_n sono ortogonali.

Poniamo $f = \sum_n c_n \phi_n$ con $c_n = \frac{\langle f,\phi_n\rangle}{\langle \phi_n,\phi_n\rangle} = \frac{1}{N}\langle f,\phi_n\rangle$ Definiamo:

$$\hat{f}(n) = \sqrt{N}c_n = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} f(x_k)e^{-\frac{i2\pi kn}{N}} \tag{5.5}$$

$$f = \frac{1}{\sqrt{N}}\sum_{n=0}^{N-1} \hat{f}(n)e^{\frac{i2\pi kn}{N}}. \tag{5.6}$$

La Trasformata di Fourier Quantistica (QFT) è una variante della DFT dove N è una potenza di 2. In particolare definiamo la QFT come l'operatore F che trasforma gli stati della base nel modo seguente:

$$|j\rangle \mapsto \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} e^{\frac{2\pi ijk}{N}}|k\rangle. \tag{5.7}$$

Assumiamo ora che $N = 2^n$ e che la base $|0\rangle, \ldots, |2^n - 1\rangle$ sia la base computazionale per gli stati su n qubits. Si è soliti rappresentare lo stato generico $|k\rangle$ come prodotto tensore $|k_1, \ldots, k_n\rangle$ dei bits della rappresentazione binaria di k: $k = k_1 2^{n-1} + \cdots + k_n 2^0 = \sum_{j=1}^{n} k_j 2^{n-j}$. Si ha quindi:

$$\begin{aligned}
QFT(|x\rangle) &= \frac{1}{\sqrt{N}}\sum_{k=0}^{2^n-1} e^{\frac{2\pi ixk}{N}}|k\rangle \\
&= \frac{1}{\sqrt{N}}\sum_{k_1\in\{0,1\}}\sum_{k_2\in\{0,1\}}\cdots\sum_{k_n\in\{0,1\}} e^{\frac{2\pi ix}{N}\sum_{j=1}^{n} k_j 2^{n-j}}|k_1 k_2 \ldots k_n\rangle \\
&= \frac{1}{\sqrt{N}}\sum_{k_1\in\{0,1\}}\cdots\sum_{k_n\in\{0,1\}}\bigotimes_{j=1}^{n} e^{\frac{2\pi ix}{N}k_j 2^{n-j}}|k_j\rangle \\
&= \frac{1}{\sqrt{N}}\sum_{k_1\in\{0,1\}}\cdots\sum_{k_n\in\{0,1\}} e^{\frac{2\pi ix}{N}k_1 2^{n-1}}|k_1\rangle \otimes \bigotimes_{j=2}^{n} e^{\frac{2\pi ix}{N}k_j 2^{n-j}}|k_j\rangle \\
&= \frac{1}{\sqrt{N}}\Big(\sum_{k_1\in\{0,1\}} e^{\frac{2\pi ix}{N}k_1 2^{n-1}}|k_1\rangle\Big) \otimes \sum_{k_2\in\{0,1\}}\cdots\sum_{k_n\in\{0,1\}} e^{\frac{2\pi ix}{N}k_2 2^{n-1}}|k_2\rangle \otimes \bigotimes_{j=3}^{n} e^{\frac{2\pi ix}{N}k_j 2^{n-j}}|k_j\rangle \\
&= \frac{1}{\sqrt{N}}\bigotimes_{j=1}^{n}\sum_{k_j\in\{0,1\}} e^{\frac{2\pi ix}{N}k_j 2^{n-j}}|k_j\rangle \\
&= \frac{1}{\sqrt{N}}\bigotimes_{j=1}^{n}(|0\rangle + e^{\frac{2\pi ix}{N}2^{n-j}}|1\rangle)
\end{aligned}$$

Ponendo $N = 2^n$ si ha

$$QFT(|x\rangle) = \frac{1}{\sqrt{2^n}}\bigotimes_{j=1}^{n}(|0\rangle + e^{2\pi i(x\cdot 2^{-j})}|1\rangle). \tag{5.8}$$

5.4 Implementazione circuitale della Trasformata di Fourier Quantistica

Per procedere con la rappresentazione circuitale è prima necessario modificare ulteriormente la (5.8) appena ricavata. Scriviamo x in espansione binaria:

$$x = x_1 2^{n-1} + x_2 2^{n-2} + \dots x_{n-1} 2^1 + x_n 2^0 = \sum_{r=1}^{n} x_r 2^{n-r}. \tag{5.9}$$

Possiamo riscrivere

$$x2^{-j} = 2^{-j} \sum_{r=1}^{n} x_r 2^{n-r} = \sum_{r=1}^{n} x_r 2^{n-j-r} = \sum_{r=1}^{n-j} x_r 2^{n-j-r} + \sum_{r=n-j+1}^{n} x_r 2^{n-j-r}. \tag{5.10}$$

$\sum_{r=1}^{n-j} x_r 2^{n-j-r} \in \mathbf{N}$, in quanto $n-j-r \geq 0 \; \forall \; r$ nella sommatoria. Da tale considerazione segue che

$$e^{2\pi i (\sum_{r=1}^{n-j} x_r 2^{n-j-r})} = 1. \tag{5.11}$$

Introduciamo la notazione binaria decimale secondo la quale

$$[0.x_1 \dots x_m] = x_1 2^{-1} + \dots x_m 2^{-m} = \sum_{k=1}^{m} x_k 2^{-k}. \tag{5.12}$$

Si può quindi scrivere

$$\sum_{r=n-j+1}^{n} x_r 2^{n-j-r} = [0.x_{n-j+1} x_{n-j+2} \dots x_n]. \tag{5.13}$$

Possiamo quindi riscrivere la (5.8) nel modo seguente:

$$QFT(|x\rangle) = \frac{1}{\sqrt{2^n}} \bigotimes_{j=1}^{n} (|0\rangle + e^{2\pi i [0.x_{n-j+1} \dots x_n]} |1\rangle). \tag{5.14}$$

Costruiamo quindi un circuito che implementi la trasformata di Fourier quantistica ricordando l'espressione della porta di Hadamard definita in (4.7) e definendo la *a cambio di fase controllata* R_m data da

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{pmatrix}. \tag{5.15}$$

Per capire il funzionamento del circuito rappresentato in Fig 5.1 iniziamo innanzitutto col notare che le x_i possono avere i valori 0 o 1, in quanto il ket $|x_1\rangle \otimes |x_2\rangle \otimes \dots \otimes |x_n\rangle = |x_1 x_2 \dots x_n\rangle$ rappresenta la scrittura in binario di un qualsiasi numero della base

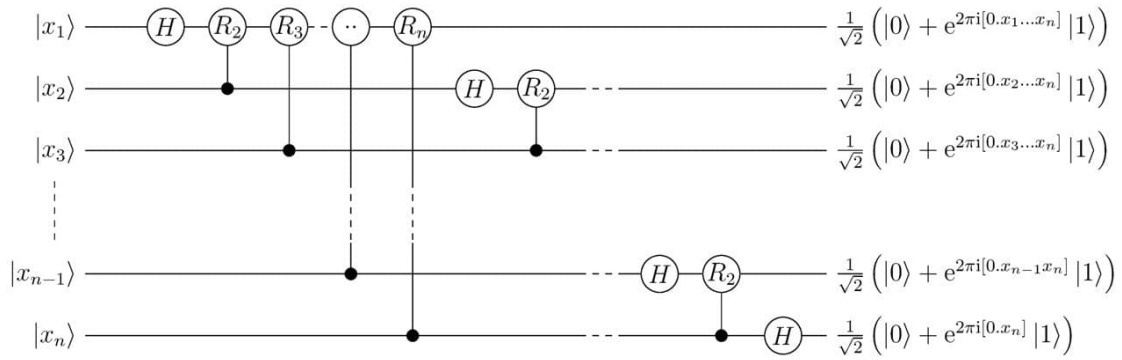

Figura 5.1. Rappresentazione circuitale della trasformata di Fourier quantistica

$|0\rangle, \ldots, |2^n - 1\rangle$. Per rappresentare l'azione della porta H (già descritta in (4.8)) possiamo sfruttare il fatto che $e^{\pi i} = -1$ e quindi scrivere

$$H|x_k\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{2\pi i}{2}x_k}|1\rangle), \tag{5.16}$$

in modo tale che, se $x_k = 0$ si ha $H|x_k\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, se $x_k = 1$ $H|x_k\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Notiamo infine che la porta R_k lascia invariato $|0\rangle$ (in quanto $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$, mentre agisce su $|1\rangle$ (essendo $|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$) ed è controllata in quanto, se $x_i = 0$, allora la fase aggiunta sarebbe nulla e tanto vale non effettuare l'operazione risparmiando risorse.
Cominciamo quindi a risolvere il circuito per verificare che l'output sia effettivamente quello riportato in figura. Indicando con H_1 la porta di Hadamard applicata al primo qubit, segue che

$$H_1|x_1x_2\ldots x_n\rangle = \frac{1}{\sqrt{2}}[|0\rangle + e^{\frac{2\pi i}{2}x_1}|1\rangle] \otimes |x_2x_3\ldots x_n\rangle\,. \tag{5.17}$$

Applicando successivamente la porta R_2 si ottiene

$$\frac{1}{\sqrt{2}}[|0\rangle + e^{\frac{2\pi i}{2}x_1 + \frac{2\pi i}{2^2}x_2}|1\rangle] \otimes |x_2x_3\ldots x_n\rangle \tag{5.18}$$

. Applicando tutte le R_k porte al primo qubit otteniamo

$$\frac{1}{\sqrt{2}}[|0\rangle + e^{\frac{2\pi i}{2}x_1 + \frac{2\pi i}{2^2}x_2 + \cdots + \frac{2\pi i}{2^{n-1}}x_{n-1} + \frac{2\pi i}{2^n}x_n}|1\rangle] \otimes |x_2x_3\ldots x_n\rangle \tag{5.19}$$

$$= \frac{1}{\sqrt{2}}[|0\rangle + e^{2\pi i[0.x_1\ldots x_n]}] \otimes |x_2x_3\ldots x_n\rangle\,. \tag{5.20}$$

Applicando lo stesso ragionamento per i qubit successivi otteniamo proprio l'output indicato in Fig. 5.1.
Notiamo che, per ricondurci esattamente alla (5.14) bisogna riordinare i qubit dall'ultimo a primo. Per operare tale riordinamento si possono effettuare $\frac{n}{2}$ inversioni di due qubit. Chiaramente, grazie alla reversibilità delle porte quantistiche, il circuito si può leggere sia da sinistra a destra, che da destra a sinistra, realizzando in quest'ultimo caso l'antitrasformata (che indicheremo con $QFT^{\dagger}$).

Capitolo 6

Algoritmo di Shor

6.1 Introduzione

Numerosi sistemi crittografici moderni si basano sulla congettura seconda la quale la fattorizzazione di numeri primi è un'operazione computazionalmente più complessa rispetto alla moltiplicazione di numeri primi. L'algoritmo classico più efficiente (il *number theoretic sieve* [7]) fattorizza un intero N in un tempo $O(e^{(logN)^{\frac{1}{3}}(loglogN)^{\frac{2}{3}}})$. Questo tempo è super-polinomiale (ovvero non è limitato superiormente da nessun polinomio) nel numero delle cifre (N.B.: per rappresentare un intero N in binario ho bisogno di $log_2 N$ bits). Nel 1994, tuttavia, P. Shor sviluppò un algoritmo quantistico in grado di risolvere il problema della fattorizzazione di numeri primi in un tempo $O((logN)^2(loglogN)(logloglogN))$, che è polinomiale nel numero delle cifre.

Fattorizzare un numero N coincide con lo scegliere un numero intero casuale m coprimo (ovvero che non abbia nessun divisore in comune) con N e trovare l'**ordine moltiplicativo** P, ovvero trovare P tale che:

$$m^P \equiv 1(modN), \tag{6.1}$$

dove il simbolo $\equiv$ sta ad indicare la **congruenza in modulo**. Quindi, $\exists k \in \mathbf{N}$ tale che

$$m^P - 1 = kN. \tag{6.2}$$

Se P è pari, allora significa che almeno un fattore di N si trova tra:

$$MCD((m^{\frac{P}{2}} + 1), N) \quad e \quad MCD((m^{\frac{P}{2}} - 1), N). \tag{6.3}$$

Infatti

$$m^P \equiv 1(modN) \Rightarrow (m^{\frac{P}{2}})^2 - 1 \equiv 0(modN) \tag{6.4}$$

$$\Rightarrow (m^{\frac{P}{2}} + 1)(m^{\frac{P}{2}} - 1) \equiv 0(modN). \tag{6.5}$$

Esempio 6.1
Sia $N = 143$ e $m = 21$.
Si vede che per $p = \{1,2,3\}$ $21^p \not\equiv 1(mod143)$
Per $p = 4$ si ha invece $21^4 - 1 = 194481$ e $\frac{194481}{143} = 1360$, quindi $p = 4$. Calcoliamo allora:

$$21^{\frac{4}{2}} - 1 = 440 \quad e \quad 21^{\frac{4}{2}} + 1 = 442$$
$$MCD(440,143) = 11, \quad MCD(442,143) = 13.$$

In questo particolare esempio sia 11 che 13 sono fattori primi di 143.

6.2 Una panoramica dell'algoritmo

L'algoritmo di Shor si compone di 5 *Step*:

1. Si scelga un numero intero positivo $m > 1$. Utilizzare un algoritmo classico ([8]) per calcolare il massimo comun divisore $gcd(m, N)$ di m e N. Se $gcd(m, N) \neq 1$ allora abbiamo trovato un fattore primo di N e abbiamo finito; altrimenti si proceda allo *Step* successivo.

2. Si utilizzi un computer quantistico per determinare il periodo incognito della funzione
$$f_N : \mathbf{N} \mapsto \mathbf{N} \tag{6.6}$$
definita da
$$a \mapsto m^a modN. \tag{6.7}$$
Si noti che la P definita nella (6.1) è proprio il periodo di tale funzione. Infatti:
$$m^{a+P} modN = m^a modN m^P modN = m^a modN$$
$$\Leftrightarrow m^P modN = 1 \Leftrightarrow m^P \equiv 1(modN).$$

3. Se P è dispari ricominciare dallo *Step 1*, se è pari procedere allo *Step 4*.

4. Se $m^{\frac{P}{2}} + 1 \equiv 0(modN)$ allora si utilizzi lo stesso algoritmo utilizzato nello *Step 1* per trovare il fattore primo $d_1 = \text{gcd}(m^{\frac{P}{2}} + 1, \text{N})$.

5. Se $m^{\frac{P}{2}} - 1 \equiv 0(modN)$ allora si utilizzi lo stesso algoritmo utilizzato nello *Step 1* per trovare il fattore primo $d_2 = \text{gcd}(m^{\frac{P}{2}} - 1, \text{N})$.

6.3 La parte quantistica dell'algoritmo di Shor

6.3.1 Introduzione e stima delle probabilità

Analizziamo più nel dettaglio lo *Step 2*. Scegliamo innanzitutto una potenza di 2, in particolare prendiamo $Q = 2^L$ tale che

$$N^2 \leq Q = 2^L \leq 2N^2, \tag{6.8}$$

e si consideri la funzione f_N (che da ora chiameremo semplicemente f) definita in (6.6) con $\mathbf{N} \equiv S_Q$ e con

$$S_Q = \{0, 1, \ldots, Q-1\}. \tag{6.9}$$

Si noti che la funzione f ha dominio e codominio coincidenti. Infatti, l'operazione $m^a mod N$, dà come risultato il resto di $\frac{m^a}{N}$ che sarà, con m intero positivo, un numero n intero tale che $0 \leq n \leq N$. Dalla (5.7) possiamo introdurre la trasformata di Fourier a Q punti

$$\hat{f}(y) = \frac{1}{\sqrt{Q}} \sum_{x \in S_Q} f(x)\omega^{xy}, \tag{6.10}$$

con $\omega = e^{\frac{2\pi i}{Q}}$. Prepariamo due qubit quantistici $|\phi\rangle$ e $|\eta\rangle$ e inizializziamoli:

$$|\psi_0\rangle = |\phi\rangle\,|\eta\rangle = |0\rangle\,|0\rangle\,. \tag{6.11}$$

Applichiamo la QFT su $|\phi\rangle$:

$$|\psi_0\rangle \mapsto |\psi_1\rangle = \frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} \omega^{0\cdot x}\,|x\rangle\,|0\rangle = \frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |x\rangle\,|0\rangle\,. \tag{6.12}$$

Chiamiamo con U_f la trasformazione unitaria che porta $|x\rangle\,|0\rangle$ a $|x\rangle\,|f(x)\rangle$. Applicando tale trasformazione si ha

$$|\psi_1\rangle = \frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |x\rangle\,|0\rangle \mapsto |\psi_2\rangle = \frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |x\rangle\,|f(x)\rangle\,. \tag{6.13}$$

In questo modo i due stati sono entangled. A questo punto applichiamo nuovamente la QFT sul primo qubit ottenendo

$$|\psi_2\rangle \mapsto |\psi_3\rangle = \frac{1}{Q} \sum_{x=0}^{Q-1} \sum_{y=0}^{Q-1} \omega^{xy}\,|y\rangle\,|f(x)\rangle \tag{6.14}$$

$$= \frac{1}{Q} \sum_{y=0}^{Q-1} \||\Gamma(y)\rangle\|\,|y\rangle\,\frac{||\Gamma(y)\rangle\rangle}{\||\Gamma(y)\rangle\|}, \tag{6.15}$$

con

$$|\Gamma(y)\rangle = \sum_{x=0}^{Q-1} \omega^{xy} |f(x)\rangle . \tag{6.16}$$

Effettuando una misura sul primo qubit mediante gli operatori di proiezione ortogonale $|0\rangle\langle 0| \otimes \mathbf{1}, |1\rangle\langle 1| \otimes \mathbf{1}, \ldots, |Q-1\rangle\langle Q-1| \otimes \mathbf{1}$ si ottiene il valore

$$y_0 \in \{0,1,\ldots,Q-1\} \tag{6.17}$$

con probabilità

$$Prob(y_0) = \frac{\||\Gamma(y_0)\rangle\|^2}{Q^2} \tag{6.18}$$

e il sistema collassa nello stato

$$|y_0\rangle \frac{|\Gamma(y_0)\rangle}{\||\Gamma(y_0)\rangle\|}. \tag{6.19}$$

Ricaviamo ora un'espressione più comoda per $|\Gamma(y)\rangle$ definendo due interi non-negativi q e r tali che:

$$Q = Pq + r, \quad 0 \leq r < P \tag{6.20}$$

e poniamo

$$Q_0 = Pq. \tag{6.21}$$

$$|\Gamma(y)\rangle = \sum_{x=0}^{Q-1} \omega^{xy} |f(x)\rangle = \sum_{x=0}^{Q_0-1} \omega^{xy} |f(x)\rangle + \sum_{x=Q_0}^{Q-1} \omega^{xy} |f(x)\rangle \tag{6.22}$$

$$= \sum_{x_0=0}^{P-1} \sum_{x_1=0}^{\frac{Q_0}{P}-1} \omega^{(Px_1+x_0)y} |f(Px_1+x_0)\rangle + \sum_{x_0=0}^{r-1} \omega^{[P(\frac{Q_0}{P})+x_0]y} |f(Px_1+x_0)\rangle \tag{6.23}$$

$$= \sum_{x_0=0}^{P-1} \omega^{x_0 y} \cdot (\sum_{x_1=0}^{\frac{Q_0}{P}-1} \omega^{Px_1 y}) |f(x_0)\rangle + \sum_{x_0=0}^{r-1} \omega^{x_0 y} \cdot \omega^{Py(\frac{Q_0}{P})} |f(x_0)\rangle \tag{6.24}$$

$$= \sum_{x_0=r}^{P-1} \omega^{x_0 y} \cdot (\sum_{x_1=0}^{\frac{Q_0}{P}-1} \omega^{Px_1 y}) |f(x_0)\rangle + \sum_{x_0=0}^{r-1} \omega^{x_0 y} \cdot (\sum_{x_1=0}^{\frac{Q_0}{P}} \omega^{Px_1 y}) |f(x_0)\rangle , \tag{6.25}$$

dove:

- in (6.19), nella prima sommatoria, si è fatta la sostituzione $xmapsto Px_1+x_0$ notando che, in tale sostituzione, $\sum_{x=0}^{Q_0-1} \equiv \sum_{x_0=0}^{P-1} \sum_{x_1=0}^{\frac{Q_0}{P}-1}$; nella seconda sommatoria si è fatta la sostituzione $x \mapsto Q_0 + x_0$ cosicché $\sum_{x=Q_0}^{Q-1} \equiv \sum_{x_0=0}^{r-1}$ con r definito in (6.20) in modo tale che $Q_0 + r = Q$.

- In (6.20) si è sfruttata la periodicità di f: $f(x+kP) = f(x)$ con $k \in \mathbf{Z}$

- In (6.21) si è sfruttato il fatto che:

$$\sum_{x_0=0}^{r-1} \omega^{x_0 y} \omega^{Py\frac{Q_0}{P}} |f(x_0)\rangle =$$
$$= \sum_{x_0=0}^{r-1} \omega^{x_0 y} (\sum_{x_1=0}^{\frac{Q_0}{P}} \omega^{Pyx_1}) |f(x_0)\rangle - \sum_{x_0=0}^{r-1} \omega^{x_0 y} (\sum_{x_1=0}^{\frac{Q_0-1}{P}} \omega^{Pyx_1}) |f(x_0)\rangle .$$

Dal momento che f, come precedentemente indicato, ha dominio e codominio coincidenti e visto che r è strettamente minore del periodo P, allora la mappa $x_0 \mapsto f(x_0)$ con $x_0 \in 0, \dots r-1$ è biunivoca e quindi, essendo l'insieme di ket

$$|0\rangle , |1\rangle , \dots , |r-1\rangle$$

un insieme ortonormale, lo sarà anche l'insieme

$$|f(0)\rangle , |f(1)\rangle , \dots , |f(r-1)\rangle .$$

Conseguentemente, considerando che nell'espressione $(\sum_{x_1=0}^{\frac{Q_0}{P}} \omega^{Pyx_1}) \omega^{x_0 y} |f(x_0)\rangle$, $\omega^{x_0 y}$ indica la fase e $(\sum_{x_1=0}^{\frac{Q_0}{P}} \omega^{Pyx_1})$ indica il modulo del fattore moltiplicativo del ket $|f(x_0)\rangle$; $\omega^{x_0 y} = e^{\frac{2\pi i}{Q} x_0 y}$ e, moltiplicandolo per il suo complesso coniugato, dà come risultato 1; $\sum_{x_0=0}^{r-1} 1 = r$ e $\sum_{x_0=r}^{P-1} 1 = P - r$, si ha allora:

$$\langle \Gamma(y), \Gamma(y) \rangle = r | \sum_{x_1=0}^{\frac{Q_0}{P}} \omega^{Pyx_1} |^2 + (P-r) | \sum_{x_1=0}^{\frac{Q_0-1}{P}} \omega^{Pyx_1} |^2 . \tag{6.26}$$

Se $Py \equiv 0 (mod N)$ allora possiamo porre $Py = kQ$ con $k \in \mathbf{N}$ e quindi $\omega^{Pyx_1} = \omega^{\frac{2\pi i}{Q} Pyx_1} = \omega^{2\pi i k x_1} = 1$ e $\sum_{x_1=0}^{\frac{Q_0}{P}} 1 = \frac{Q_0}{P} + 1$ e

$$\langle \Gamma(y), \Gamma(y) \rangle = r(\frac{Q_0}{P} + 1)^2 + (P-r)(\frac{Q_0}{P})^2 . \tag{6.27}$$

D'altra parte, se $Py \not\equiv 0 (mod Q)$, allora possiamo usare la serie geometrica per ottenere:

$$\langle \Gamma(y), \Gamma(y) \rangle = r | \frac{\omega^{Py(\frac{Q_0}{P}+1)} - 1}{\omega^{Py} - 1} |^2 + (P-r) | \frac{\omega^{Py(\frac{Q_0}{P})} - 1}{\omega^{Py} - 1} |^2 \tag{6.28}$$
$$= r | \frac{\omega^{\frac{2\pi i}{Q} Py(\frac{Q_0}{P}+1)} - 1}{\omega^{Py} - 1} |^2 + (P-r) | \frac{\omega^{\frac{2\pi i}{Q} Py(\frac{Q_0}{P})} - 1}{\omega^{Py} - 1} |^2 . \tag{6.29}$$

Sfruttando infine il fatto che

$$|e^{i\theta} - 1|^2 = 4 sin^2 (\frac{\theta}{2}) \tag{6.30}$$

e che, dal Postulato 3 (§3.1.3)

$$Prob(y) = | \langle \Gamma(y), \Gamma(y) \rangle |^2 , \tag{6.31}$$

si ha:

$$Prob(y) = \begin{cases} \frac{r(Q_0+P)^2+(P-r)Q_0^2}{Q^2P^2} & \text{se } Py \equiv 0(modQ) \\ \frac{rsin^2(\frac{\pi Py}{Q}(\frac{Q_0}{P}+1))+(P-r)sin^2(\frac{\pi Py}{Q}\frac{Q_0}{P})}{Q^2sin^2(\frac{\pi Py}{Q})} & \text{se } Py \not\equiv 0(modQ) \end{cases}. \tag{6.32}$$

Se P è un divisore esatto di Q, allora, da (6.20), segue che $r = 0$ e quindi $Q_0 = Q$. Si ottiene quindi:

$$Prob(y) = \begin{cases} \frac{1}{P} & \text{se } Py \equiv 0(modN) \\ 0 & \text{se } Py \not\equiv 0(modN) \end{cases}. \tag{6.33}$$

Definiamo

$$\begin{cases} a \equiv \{a\}_Q(modQ) \\ -\frac{Q}{2} < \{a\}_Q \leq \frac{Q}{2} \end{cases}. \tag{6.34}$$

Da (6.34) segue che $a = kQ + \{a\}_Q$ con $k \in \mathbf{N}$. $-\frac{Q}{2} < \{a\}_Q \leq \frac{Q}{2}$ perché:

- se $\{a\}_Q < -\frac{Q}{2}$, potrei passare da k a $k-1$ ottenendo quindi una $\{a\}'_Q < \frac{Q}{2}$;

- se $\{a\}_Q > \frac{Q}{2}$, potrei passare da k a $k+1$ ottenendo quindi una $\{a\}'_Q > -\frac{Q}{2}$.

Imponendo tali condizioni sulla $\{a\}_Q$, sto quindi cercando il resto $\{a\}_Q$ minimo per cui valga $a = kQ + \{a\}_Q$ con $k \in \mathbf{N}$. Per soddisfarre tali condizioni si deve avere

$$\{a\}_Q = a - Q\lfloor\frac{a}{Q} + \frac{1}{2}\rfloor. \tag{6.35}$$

Il simbolo $\lfloor x \rfloor$ indica la parte intera di x, ovvero il più grande intero n tale che $n \leq x$. Notiamo che

$$Py \equiv 0(modN) \Leftrightarrow \{Py\}_Q = 0.$$

Ricordando (6.8), $Q_0 < Q$ e ponendoci nelle condizioni

$$0 < P \leq N \tag{6.36}$$

$$0 < |\{Py\}_Q| \leq \frac{P}{2}(1 - \frac{1}{N}) \tag{6.37}$$

(dove la condizione (6.36), ricordiamo, implica che la funzione f definita in (6.6) sia biettiva), stimiamo il valore $|\frac{\pi\{Py\}_Q}{Q}(\frac{Q_0}{P}+1)|$:

$$|\frac{\pi\{Py\}_Q}{Q}(\frac{Q_0}{P}+1)| \leq \frac{\pi}{Q}\frac{P}{2}(1-\frac{1}{N})(\frac{Q_0+P}{P}) \leq \frac{\pi}{2}(1-\frac{1}{N})(\frac{Q+P}{Q}) \tag{6.38}$$

$$\leq \frac{\pi}{2}(1-\frac{1}{N})(1+\frac{P}{Q}) \leq \frac{\pi}{2}(1-\frac{1}{N})(1+\frac{N}{N^2}) \tag{6.39}$$

$$< \frac{\pi}{2}. \tag{6.40}$$

Dal momento che $\frac{sin(\theta)}{\theta}$ è monotona decrescente in $(0, \frac{\pi}{2})$, $\lim_{\theta\to 0}\frac{sin(\theta)}{\theta} = 1$ e $\frac{sin(\frac{\pi}{2})}{\frac{\pi}{2}} = \frac{2}{\pi}$ vale la relazione

$$\frac{4}{\pi^2}\theta^2 \leq sin^2(\theta) \leq \theta^2 \quad \text{se} \quad |\theta| < \frac{\pi}{2}. \tag{6.41}$$

$|\frac{\pi\{Py\}_Q}{Q}\frac{Q_0}{P}| < |\frac{\pi\{Py\}_Q}{Q}(\frac{Q_0}{P}+1)| < \frac{\pi}{2}$ e dunque soddisfa (6.41). Dalla seconda condizione di (6.32) segue che:

$$\begin{aligned}
Prob(y) &= \frac{rsin^2(\frac{\pi Py}{Q}(\frac{Q_0}{P}+1)) + (P-r)sin^2(\frac{\pi Py}{Q}\frac{Q_0}{P})}{Q^2 sin^2(\frac{\pi Py}{Q})} \\
&\geq \frac{r\frac{4}{\pi^2}(\frac{\pi\{Py\}_Q}{Q}(\frac{Q_0}{P}+1))^2 + (P-r)\frac{4}{\pi^2}(\frac{\pi\{Py\}_Q}{Q}\frac{Q_0}{P})^2}{Q^2(\frac{\pi\{Py\}_Q}{Q})^2} \\
&= \frac{4}{\pi^2}\frac{r(\frac{Q_0^2}{P^2}+\frac{2Q_0}{P}+1) + P\frac{Q_0^2}{P^2} - r\frac{Q_0^2}{P^2}}{Q^2} \\
&\geq \frac{4}{\pi^2}\frac{P(\frac{Q_0}{P})^2}{Q^2} = \frac{4}{\pi^2}\frac{1}{P}(\frac{Q-r}{Q})^2 \\
&= \frac{4}{\pi^2}\frac{1}{P}(1-\frac{r}{Q})^2 \geq \frac{4}{\pi^2}\frac{1}{P}(1-\frac{1}{N})^2.
\end{aligned}$$

Segue quindi

$$Prob(y) \geq \begin{cases} \frac{1}{P}(1-\frac{1}{N})^2 & \text{se } \{Py\}_Q = 0 \\ \frac{4}{\pi^2}\frac{1}{P}(1-\frac{1}{N})^2 & \text{se } |\{Py\}_Q| \leq \frac{P}{2}(1-\frac{1}{N}) \end{cases}, \tag{6.42}$$

dove la prima disuguaglianza si ottiene banalmente da (6.32) ponendoci nelle condizioni (6.36).

6.3.2 Digressione sulle frazioni continue

Ogni numero razionale positivo Q può essere espresso nella forma

$$Q = a + \cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{\cdots+\frac{1}{b}}}}} \tag{6.43}$$

con $a = \lfloor Q \rfloor$. La costruzione cessa quando si ottiene $b \in \mathbf{N}$. Vediamo un esempio per capire come costruire una funzione continua.

Esempio 6.2
Sia $Q = \frac{47}{13}$

$$a = \lfloor \frac{47}{13} \rfloor = 3$$

In particolare, si ha $\frac{47}{13} = 3$ con resto 8.

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

Il numero $Q = \frac{47}{13}$ si può indicare nella forma compatta come [3,1,1,1,1,2]. I valori [3] = 3, [3,1] = 4, [3,1,1] = $\frac{7}{2}$, [3,1,1,1] = $\frac{11}{3}$, [3,1,1,1,1] = $\frac{31}{8}$ sono detti ***convergenti*** *n-esimi*.

Un importante teorema legato alla teoria delle funzioni continue (di cui omettiamo la dimostrazione in quanto non attinente agli obiettivi di questo lavoro) afferma che: sia Q un numero razionale e siano a e b due interi con $b > 0$. Se

$$|Q - \frac{a}{b}| \leq \frac{1}{2b^2}, \tag{6.44}$$

allora il numero $\frac{a}{b}$ è un convergente dell'espansione in funzioni continue di Q.

Definiamo inoltre la funzione **totiente di Eulero** $\phi(N)$ che indica il numero di interi positivi minori di N che sono coprimi a N. Si dimostra [9] che

$$\liminf \frac{\phi(N)}{N/lnlnN} = e^{-\gamma}, \tag{6.45}$$

con $\gamma = 0.57721566490\ldots$ detta **costante di Eulero** e lim inf definito come

$$\liminf_{n\to\infty} = \lim_{n\to\infty}(\inf_{m\geq n} x_m) = \sup_{n\geq 0}\inf_{m\geq n} x_m \tag{6.46}$$

(ovvero, è possibile pensare al limite inferiore come al più grande dei estremi inferiori).

6.3.3 Parte finale dello *Step 2*

Definiamo l'insieme

$$Y = \{y \in S_Q \mid \quad |\{Py\}_Q| \leq \frac{P}{2}\}. \tag{6.47}$$

Come risultato della misura effettuata in (6.17), abbiamo ottenuto un intero y. La probabilità che $y \in Y$ può essere ottenuta con buona approssimazione da (6.42). Infatti, la condizione $|\{Py\}_Q| \leq \frac{P}{2}(1 - \frac{1}{N}) \simeq \frac{P}{2}$ con N sufficientemente grande (come è lecito pensare, in quanto un N piccolo sarebbe banale da fattorizzare). Quindi

$$Prob(y \in Y) \geq \frac{4}{\pi^2}\frac{1}{P}(1 - \frac{1}{N})^2. \tag{6.48}$$

Definiamo

$$d = \lfloor \frac{P}{Q} y \rfloor, \tag{6.49}$$

segue allora da (6.35) che

$$\{Py\}_Q = Py - Qd \tag{6.50}$$

. Dalla condizione (6.47) segue che

$$\begin{aligned} |Py - Qd| &\leq \frac{P}{2} \\ \Rightarrow |\frac{y}{Q} - \frac{d}{P}| &\leq \frac{1}{2Q}. \end{aligned}$$

Dal momento che $Q \geq N^2$, segue che

$$|\frac{y}{Q} - \frac{d}{P}| \leq \frac{1}{2N^2}. \tag{6.51}$$

Visto che $P \leq N \Rightarrow \frac{1}{2N^2} \leq \frac{1}{2P^2}$ e quindi si può utilizzare (6.44) e stabilire che $\frac{d}{P}$ è un convergente dell'espansione a frazione continua di $\frac{y}{Q}$. Quindi vale la relazione

$$\frac{d}{P} = \frac{p_n}{q_n}, \tag{6.52}$$

con $\frac{p_n}{q_n}$ n-esimo convergente di $\frac{y}{Q}$. Tuttavia, per fare l'identificazione

$$\begin{cases} p_n = d \\ q_n = P \end{cases}, \tag{6.53}$$

è necessario che d e P siano primi fra loro (altrimenti conoscerei solamente il loro rapporto e quindi non potrei stabilire il valore di P). Da (6.48) e dalla definizione della funzione totiente segue che

$$Prob\{y \in Y \mid gcd(d(y), P) = 1\} \geq \frac{4}{\pi^2}\frac{\phi(P)}{P}(1 - \frac{1}{N})^2. \tag{6.54}$$

Da (6.45) segue che

$$Prob\{y \in Y \mid gcd(d, P) = 1\} \geq \frac{4}{\pi^2 ln2}\frac{e^{-\gamma} - \epsilon^P}{log_2 log_2 N}(1 - \frac{1}{N})^2, \tag{6.55}$$

con $\epsilon(P)$ funzione monotona decrescente convergente a 0 e quindi limitata. Quindi

$$Prob\{y \in Y \mid gcd(d, P) = 1\} = \Omega(\frac{1}{log_2 log_2 N}), \tag{6.56}$$

dove la notazione *omega grande* $\Omega(*)$ indica indica una convergenza asintotica almeno come $(*)$. Dunque, per terminare lo *Step 2*, si calcolino i convergenti dell'espansione in frazioni continue di $\frac{y}{Q}$. Partendo con $n = 1$ si verifichi se è vera (6.53) verificando se

$$m^{q_n} \equiv 1(modN). \tag{6.57}$$

Se tale relazione è verificata, allora si ponga $q_n = P$ e si proceda allo *Step 3*; in caso contrario tentare con gli altri n.

Capitolo 7

Implementazione dell'Algoritmo di Shor

7.1 Preparazione dell'ambiente di sviluppo

In questo capitolo il nostro scopo è quello di creare un algoritmo funzionante per applicare l'algoritmo di Shor. Per riuscire in questo obiettivo utilizzeremo l'SDK **Qiskit**[10], un serivizio open-source che contiene strumenti per creare e manipolare circuiti quantistici interfacciando linguaggi di programmazione classici con i computer quantistici messi a disposizione dall'**IBM Q Experience**. In particolare si è utilizzato il linguaggio *python* e l'ambiente di sviluppo integrato *Spyder*. Per configurare il tutto è necessario avere la distribuzione *Anaconda* installata sul PC. Dal terminale Anaconda digitare i comandi:

```
conda create -n my-quantum-env python=3.8
conda activate my-quantum-env
pip install qiskit
pip install qiskit[visualization]
pip install spyder
pip install numpy==1.19.3
```

Fare attenzione a non utilizzare la versione *Python 3.9* in quanto non funziona correttamente con *Qiskit* (stessa cosa con la versione *Numpy 1.19.4*. A questo punto, per lanciare l'ambiente di sviluppo con i pacchetti appena installati è sufficiente digitare, nel terminale di Anaconda, il comando

spyder

7.2 Introduzione al problema

In questa sezione useremo un approccio in parte diverso[10] rispetto a quello usato nel capitolo precedente che ci aveva permesso una stima precisa delle probabilità e tempo impiegato dall'algoritmo per essere portato a termine. Avendo già ottenuto queste stime, possiamo ora usare un approccio la cui implementazione circuitale sia più semplice (e di conseguenza anche la scrittura del codice per eseguirlo) senza però snaturare le idee di fonda già descritte.
Iniziamo con il definire un operatore unitario U, tale che

$$U\left|y\right\rangle \equiv \left|my \bmod N\right\rangle . \tag{7.1}$$

Ogni applicazione successiva dell'operatore U moltiplicherà il nostro stato per $m\,(mod\,N)$ (dove le parentesi tonde intorno al modulo stanno ad indicare che bisogna prima moltiplicare il valore del nostro stato per m e poi effettuare l'operazione di modulo). Supponiamo che dopo P applicazioni dell'operatore P, ci ritroviamo nello stato di partenza e poniamo, per esempio, $\left|y\right\rangle = \left|1\right\rangle$, $N = 35$ e $m = 3$:

$$\begin{aligned} U\left|1\right\rangle &= \left|3\right\rangle \\ U^2\left|1\right\rangle &= \left|9\right\rangle \\ &\vdots \\ U^{P-1}\left|1\right\rangle &= \left|12\right\rangle \\ U^{P}\left|1\right\rangle &= \left|1\right\rangle \quad . \end{aligned}$$

Dal momento che $y = 1$, $U\left|y\right\rangle = \left|m \bmod N\right\rangle$, e quindi, trovare P tale che $U^P\left|y\right\rangle = \left|y\right\rangle$ equivale a trovare P tale che $\left|m^P \bmod N\right\rangle = \left|m \bmod N\right\rangle$ (ovvero il periodo P della funzione $f : a \mapsto a \bmod N$. Per avere una rappresentazione numerica e visiva del periodo dell'esempio proposto con $N = 35$ e $m = 3$ scriviamo un programma in python riportato di seguito.

```
1  # -*- coding: utf-8 -*-
2  """
3  Shor's Algorithm
4
5  @author: mattia
6  """
7
8  import matplotlib.pyplot as plt
9  import numpy as np
10
11 #Let's graphically see an example of the function f finding its period  P
```

```
print("Let's see an example on the periodicity of the function f(a)=m^a
    modN:")
N = 35
m = 3
xvals = np.linspace(1, 35, 35)
yvals = [m**x % N for x in xvals]
plt.plot(xvals, yvals, c = 'b')
plt.scatter(xvals, yvals, label = 'punti', c='r', marker='x')
plt.grid()
plt.legend()
plt.xlabel('x')
plt.ylabel('3^x mod 35')
plt.xlim(0, 35)
plt.show()
try:
    P = yvals.index(1) + 1
    print('The period P is equal to: %5.0f' %P)
except:
    print("It wasn't possible to find the period', verify that m < N and
    that m and N are relatively prime")
```

Listato 7.1. Rappresentazione visiva del Periodo

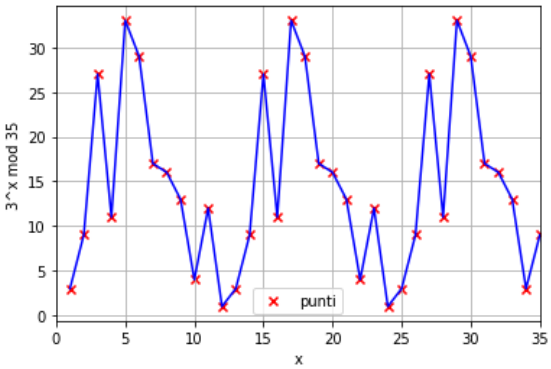

Figura 7.1. Rappresentazione visiva del periodo

Il periodo è 12

Una sovrapposizione di stati

$$|u_0\rangle = \frac{1}{\sqrt{P}} \sum_{k=0}^{P-1} |m^k \; mod \; N\rangle \tag{7.2}$$

sarà un autostato di U. Torniamo al nostro esempio per avere un'idea più chiara:

$$\begin{aligned} |u_0\rangle &= \frac{1}{\sqrt{12}}(|1\rangle + |3\rangle + |9\rangle + \cdots + |4\rangle + |12\rangle) \\ U\,|u_0\rangle &= \frac{1}{\sqrt{12}}(U\,|1\rangle + U\,|3\rangle + U\,|9\rangle + \cdots + U\,|4\rangle + U\,|12\rangle) \\ &= \frac{1}{\sqrt{12}}(|3\rangle + |9\rangle + |27\rangle + \cdots + |12\rangle + |1\rangle) \\ &= |u_0\rangle \quad . \end{aligned}$$

In questo esempio l'autovalore è 1, che non è molto interessante. Definendo invece

$$|u_s\rangle = \frac{1}{\sqrt{P}} \sum_{k=0}^{P-1} e^{-\frac{2\pi i s k}{P}}\, |m^k \; mod \; N\rangle \tag{7.3}$$

$$U\,|u_s\rangle = e^{\frac{2\pi i s}{P}}\,|u_s\rangle \tag{7.4}$$

(ovvero l'autovalore è $e^{\frac{2\pi i s}{P}}$). Per visulaizzare meglio il concetto usiamo ancora una volta il nostro esempio:

$$\begin{aligned} |u_s\rangle &= \frac{1}{\sqrt{12}}(|1\rangle + e^{-\frac{2\pi i s}{12}}\,|3\rangle + e^{-\frac{4\pi i s}{12}}\,|9\rangle + \cdots + e^{-\frac{20\pi i s}{12}}\,|4\rangle + e^{-\frac{22\pi i s}{12}}\,|12\rangle) \\ U\,|u_s\rangle &= \frac{1}{\sqrt{12}}(|3\rangle + e^{-\frac{2\pi i s}{12}}\,|9\rangle + e^{-\frac{4\pi i s}{12}}\,|27\rangle + \cdots + e^{-\frac{20\pi i s}{12}}\,|12\rangle + e^{-\frac{22\pi i s}{12}}\,|1\rangle) \\ &= e^{\frac{2\pi i s}{12}}\,\frac{1}{\sqrt{12}}(e^{-\frac{2\pi i s}{12}}\,|3\rangle + e^{-\frac{4\pi i s}{12}}\,|9\rangle + e^{-\frac{6\pi i s}{12}}\,|27\rangle + \cdots + e^{-\frac{22\pi i s}{12}}\,|12\rangle + e^{-\frac{24\pi i s}{12}}\,|1\rangle) \\ &= e^{\frac{2\pi i s}{12}}\,|u_s\rangle\,. \end{aligned}$$

Possiamo porre

$$0 \leq s \leq P-1 \tag{7.5}$$

in modo tale da avere P vettori linearmente indipendenti che formano quindi una base. Vale la relazione

$$\frac{1}{\sqrt{P}} \sum_{k=0}^{P-1} |u_s\rangle = |1\rangle \tag{7.6}$$

(in quanto le fasi di tutti i ket si cancellano, tranne la fase di $|1\rangle$ che vale $2\pi i n$, con $n \in \mathbf{N}$, $\forall\ u_s$). Verifichiamolo questa volta utilizzando valori di m e N diversi dagli esempi precedenti in modo tale da ottenere una P più piccola. In particolare, prendiamo $m = 7$

e $N = 15$. Si può verificare (per esempio utilizzando il Listato 7.1) che $P = 4$.

$$\begin{aligned}
&\frac{1}{\sqrt{4}}[\,|u_0\rangle + |u_1\rangle + |u_2\rangle + |u_3\rangle] \\
&= \frac{1}{2}\,[\,\frac{1}{2}(|7\rangle + |4\rangle + |13\rangle + |1\rangle) + \\
&\quad + \frac{1}{2}(e^{-\frac{2\pi i}{4}}\,|7\rangle + e^{-\frac{4\pi i}{4}}\,|4\rangle + e^{-\frac{6\pi i}{4}}\,|13\rangle + e^{-\frac{8\pi i}{4}}\,|1\rangle) + \\
&\quad + \frac{1}{2}(e^{-\frac{4\pi i}{4}}\,|7\rangle + e^{-\frac{8\pi i}{4}}\,|4\rangle + e^{-\frac{12\pi i}{4}}\,|13\rangle + e^{-\frac{16\pi i}{4}}\,|1\rangle) + \\
&\quad + \frac{1}{2}(e^{-\frac{6\pi i}{4}}\,|7\rangle + e^{-\frac{12\pi i}{4}}\,|4\rangle + e^{-\frac{18\pi i}{4}}\,|13\rangle + e^{-\frac{24\pi i}{4}}\,|1\rangle)] \\
&\quad = |1\rangle\,.
\end{aligned}$$

7.3 Quantum Phase Estimation

Per procedere oltre è necessario introdurre un algoritmo quantistico di stima della fase. Dato un operatore unitario U e un autovettore $|\psi\rangle$ tali che $U\,|\psi\rangle = e^{2\pi i\theta}\,|\psi\rangle$, tale algoritmo stima la fase θ. Il circuito necessario per la realizzazione di tale algoritmo è il seguente: Lo stato di partenza è dato da

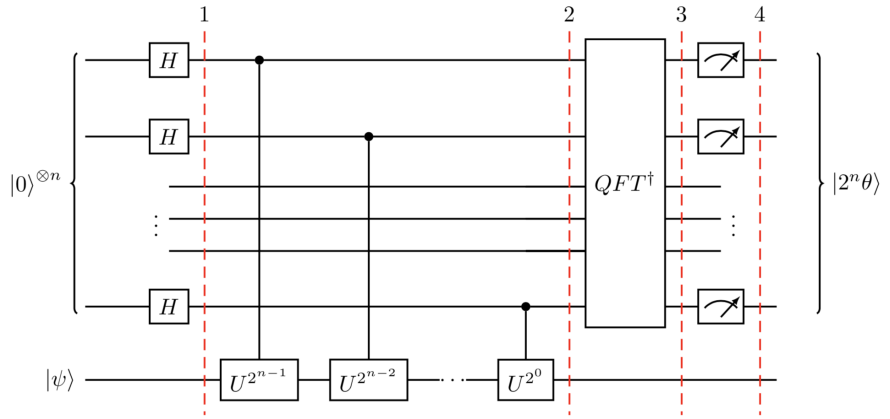

Figura 7.2. Rappresentazione circuitale dell'algoritmo quantistico di stima della fase

$$\psi_0 = |0\rangle^{\otimes n}\,|\psi\rangle\,. \tag{7.7}$$

Applicando le porte di Hadamard si ottiene

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle)^{\otimes n}\,|\psi\rangle\,. \tag{7.8}$$

Scriviamo

$$U^{2^j}|\psi\rangle = U^{2^j-1}U|\psi\rangle = U^{2^j-1}e^{2\pi i\theta}|\psi\rangle = U^{2^j-2}e^{2\cdot 2\pi i\theta}|\psi\rangle = U^{2^j-3}e^{2\cdot 3\pi i\theta}|\psi\rangle \tag{7.9}$$

$$= \cdots = e^{2\pi i 2^j\theta}|\psi\rangle\,. \tag{7.10}$$

Considerando che le porte U sono operazioni controllate, allora l'operatore U agirà solamente sui ket 1 delle sovrapposizioni ottenute tramite porte di Hadamard. Si ottiene quindi:

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}}(|0\rangle + e^{2\pi i\theta 2^{n-1}}|1\rangle)\otimes\cdots\otimes(|0\rangle + e^{2\pi i\theta 2^1}|1\rangle)\otimes(|0\rangle + e^{2\pi i\theta 2^0}|1\rangle)\otimes|\psi\rangle\,. \tag{7.11}$$

Consideriamo che $(|0\rangle + |1\rangle)^{\otimes n}$ dà come risultato la somma di tutti i possibili ket tra $|00\ldots0\rangle$ (un ket con n zeri) e $|11\ldots1\rangle$ (un ket con n uno), che possono essere interpretati rispettivamente come la scrittura in binario del numero 0 e del numero 2^n-1. Inoltre, dalle regole delle somme geometriche, segue che $\sum_{k=0}^{n-1}2^k = 2^n-1$. Dunque, indicando con k la rappresentazione decimale del corrispettivo numero binario, la (7.11) può essere riscritta nel modo seguente:

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1}e^{2\pi i\theta k}|k\rangle\otimes|\psi\rangle\,. \tag{7.12}$$

Tale espressione, sebbene presenti qualche differenza, è molto simile alla formula della trasformata di Fourier quantistica (5.8). Proviamo quindi ad applicare un'antitrasformata di Fourier quantistica sullo stato $\frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1}e^{2\pi i\theta k}|k\rangle$. Notando che l'antitrasformata della somma di ket è la somma delle antitrasformate, si ottiene:

$$QFT^\dagger(\frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1}e^{2\pi i\theta k}|k\rangle) = \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1}e^{2\pi i\theta k}\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}e^{\frac{-2\pi ikx}{2^n}}|x\rangle \tag{7.13}$$

$$= \frac{1}{2^n}\sum_{x=0}^{2^n-1}\sum_{k=0}^{2^n-1}e^{2\pi i\theta k}e^{\frac{-2\pi ikx}{2^n}}|x\rangle \tag{7.14}$$

$$= \frac{1}{2^n}\sum_{x=0}^{2^n-1}\sum_{k=0}^{2^n-1}e^{\frac{-2\pi ik}{2^n}(x-2^n\theta)}|x\rangle\,. \tag{7.15}$$

Quindi si ha che

$$|\psi_3\rangle = \frac{1}{2^n}\sum_{x=0}^{2^n-1}\sum_{k=0}^{2^n-1}e^{\frac{-2\pi ik}{2^n}(x-2^n\theta)}|x\rangle\otimes|\psi\rangle\,. \tag{7.16}$$

Poniamo $2^n\theta = a + 2^n\delta$, con a l'intero più vicino a $2^n\theta$ e con $0 \le |2^n\delta| \le \frac{1}{2}$. Effettuando una misura sul primo registro dello stato $|\psi_3\rangle$ si ottiene il valore a con probabilità:

$$Prob(a) = |\langle a|\frac{1}{2^n}e^{\frac{-2\pi ik}{2^n}(x-a)}e^{2\pi i\delta k}|x\rangle|^2. \tag{7.17}$$

Tale valore è nullo $\forall\, x \neq a$, quindi

$$Prob(a) = \frac{1}{2^{2n}} |\sum_{k=0}^{2^n-1} e^{2\pi i \delta k}|^2 = \begin{cases} 1 & \delta = 0 \\ \frac{1}{2^{2n}} |\frac{1-e^{2\pi i 2^n \delta}}{1-e^{2\pi i \delta}}|^2 & \delta \neq 0 \end{cases}. \tag{7.18}$$

Quindi, per $\delta = 0$, si ha $a = 2^n\theta$ e $Prob(a) = 1$. La stima della fase è esatta e lo stato finale collassa in

$$|\psi_4\rangle = |2^n\theta\rangle\,|\psi\rangle\,. \tag{7.19}$$

Se $\delta \neq 0$, siccome $|\delta| \leq \frac{1}{2} << \frac{1}{2^{n+1}}$, allora la probabilità di ottenere a è comunque buona ($\geq \frac{4}{\pi^2}$). Infatti si ha che

$$\begin{aligned} Prob(a) &= \frac{1}{2^{2n}} |\frac{1-e^{2\pi i 2^n \delta}}{1-e^{2\pi i \delta}}|^2 && (7.20)\\ &= \frac{1}{2^{2n}} |\frac{2sin(\pi 2^n \delta)}{2sin(\pi\delta)}|^2 && (7.21)\\ &= \frac{1}{2^{2n}} \frac{|sin(\pi 2^n \delta)|^2}{|sin(\pi\delta)|^2} && (7.22)\\ &\geq \frac{1}{2^{2n}} \frac{|sin(\pi 2^n \delta)|^2}{|\pi\delta|^2} && (7.23)\\ &\geq \frac{1}{2^{2n}} \frac{|2 \cdot 2^n \delta|^2}{|\pi\delta|^2} && (7.24)\\ &\geq \frac{4}{\pi^2}, && (7.25) \end{aligned}$$

dove la prima uguaglianza è giustificata dal fatto che $|1 - e^{2ix}|^2 = |2sin(x)|^2$, la prima disuguaglianza da $|sin(\pi\delta)| \leq |\pi\delta|$ per $|\delta| << \frac{1}{2^{n+1}}$ e la seconda disuguaglianza da $|2 \cdot 2^n\delta| \leq |sin(\pi 2^n \delta)|$ per $|\delta| << \frac{1}{2^{n+1}}$.

7.4 Implementazione circuitale

Torniamo all'algoritmo di Shor: dal momento che lo stato $|1\rangle$ è sovrapposizione di tutti gli autostati dell'operatore U, applicando l'algoritmo *Quantum Phase Estimation* su U utilizzando lo stato $|1\rangle$, si misurerà un fase

$$\phi = \frac{s}{P}, \tag{7.26}$$

con s un numero random tra 0 e $r-1$. A questo punto è possibile sfruttare le frazioni continue per ottenere il valore P (come descritto in §6.3.3).
Per implementare l'algoritmo poniamo $N = 15$ e $m = 7$. Notiamo che, essendo $N = 15 = (1111)_2$, avremo bisogno di 4 qubits per rappresentare il ket $|1\rangle$ a cui vengono applicate le porte U. Per il resto, il circuito è esattamente analogo a quello visto per la stima

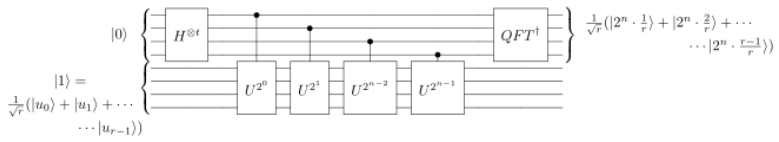

Figura 7.3. Rappresentazione circuitale dell'algoritmo di Shor

della fase. Un'importante caratteristica che ci permetterà di implementare facilmente le operazioni $U|y\rangle = |7y \, mod \, 15\rangle$ è la seguente:

$$U|1\rangle = |7\rangle = |0111\rangle \xrightarrow[\text{SWAP 2,3}]{} |0111\rangle \xrightarrow[\text{SWAP 1,2}]{} |0111\rangle \xrightarrow[\text{SWAP 0,1}]{} |1011\rangle \xrightarrow[X]{} |0100\rangle \tag{7.27}$$

$$U|7\rangle = |4\rangle = |0100\rangle \xrightarrow[\text{SWAP 2,3}]{} |0100\rangle \xrightarrow[\text{SWAP 1,2}]{} |0010\rangle \xrightarrow[\text{SWAP 0,1}]{} |0010\rangle \xrightarrow[X]{} |1101\rangle \tag{7.28}$$

$$U|4\rangle = |13\rangle = |1101\rangle \xrightarrow[\text{SWAP 2,3}]{} |1110\rangle \xrightarrow[\text{SWAP 1,2}]{} |1110\rangle \xrightarrow[\text{SWAP 0,1}]{} |1110\rangle \xrightarrow[X]{} |0001\rangle \tag{7.29}$$

$$U|13\rangle = |1\rangle = |0001\rangle \xrightarrow[\text{SWAP 2,3}]{} |0010\rangle \xrightarrow[\text{SWAP 1,2}]{} |0100\rangle \xrightarrow[\text{SWAP 0,1}]{} |1000\rangle \xrightarrow[X]{} |0111\rangle \tag{7.30}$$

dove l'operazione `SWAP x,y` inverte il qubit x con quello y e la porta X è quella definita in (4.5) e mappa $|0\rangle \mapsto |1\rangle$, $|1\rangle \mapsto |0\rangle$. Inoltre, il circuito che useremo per implementare l'antitrasformata di Fourier quantistica avrà la forma seguente:

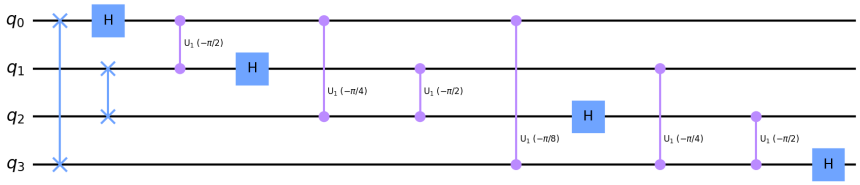

Figura 7.4. Rappresentazione circuitale dell'antitrasformata di Fourier quantistica che agisce su 4 qubits

Come è possibile notare dalla Fig. 7.4, la $\text{QFT}^\dagger$ è stata implementata in modo "opposto" (nel senso che va letta dal basso verso l'alto) rispetto a quella vista nei paragrafi precedenti. Si è fatta questa scelta in quanto in questo modo l'implementazione attraverso

iterazioni sarà molto più semplice (come sarà possibile vedere nel codice che segue). Il circuito che si ottiene per implementare l'algoritmo di Shor è invece il seguente:

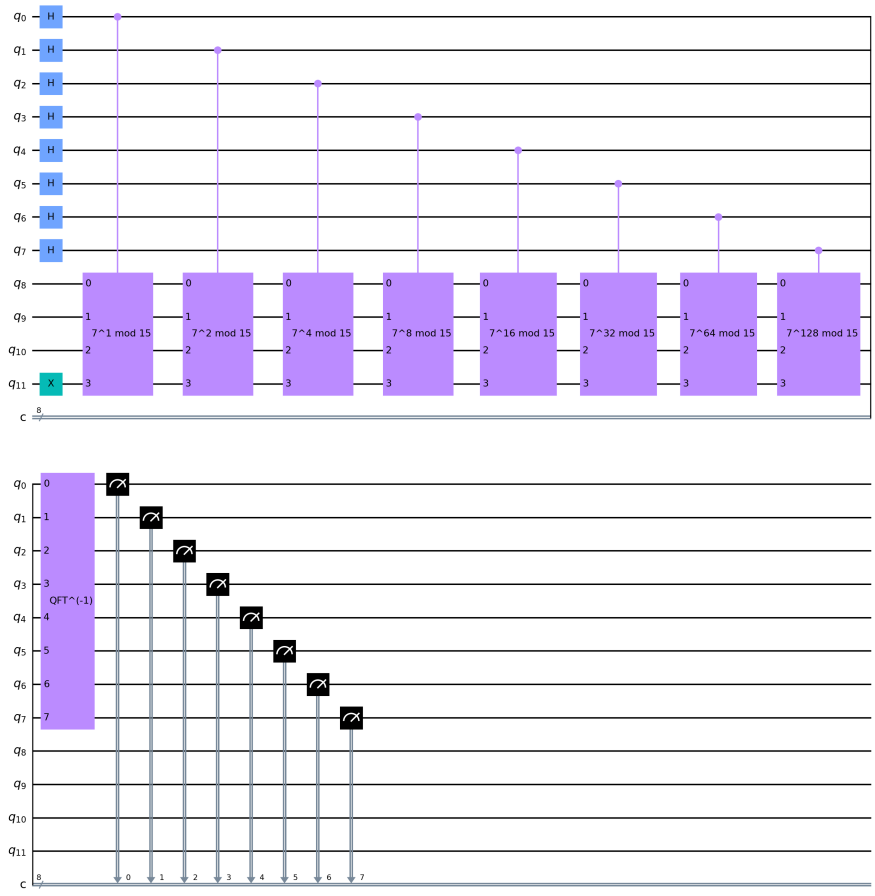

Figura 7.5. Circuito utilizzato per l'implementazione dell'algoritmo di Shor

Sia la Fig. 7.4, sia la Fig.7.5 sono state ottenute come output del codice che segue scritto in Python.

```
# -*- coding: utf-8 -*-
"""
Shor's Algorithm

@author: mattia
"""


import numpy as np
from qiskit import QuantumCircuit, Aer, execute
from math import gcd
from fractions import Fraction


#Define the function c_7mod15 which returns controlled-U gate for m repeated x times
#c_7mod15 will be a 4 qubit unitary matrix controlled by a 5th qubit which will be added
def c_7mod15(power):
    "Controlled multiplication by 7 mod 15"
    U = QuantumCircuit(4) #crea un circuito a 4 qubits
    for iteration in range(power):
        U.swap(2,3)
        U.swap(1,2)
        U.swap(0,1)
        for q in range(4):
            U.x(q) #Apply X gate to the q-th qubit
    U = U.to_gate() #Define a custom gate
    U.name = "%i^%i mod 15" %(7, power) #Give a name to the custom gate
    c_U = U.control() #Make the custom gate controlled
    return c_U


#Define the circuit for the inverse QFT
def qft_dagger(n):
    "inverse quantum Fourier transform to the first n qubits"
    qc = QuantumCircuit(n)
    for qubit in range(n//2): #The swaps are done at the beginning because this is the inverse QFT; if n is odd then the central term doesn't need to be swapped
        qc.swap(qubit, n-qubit-1) #n-qubit indicates the (n-qubit)-th qubit, note that we start counting the qubits from 0 to n-1
    for j in range(n):
        for m in range(j):
```

```
40              qc.cu1(-2*np.pi/float(2**(j-m+1)), m, j) #the function qc.cu1(
    x, q[0], q[1]) applies the rotation (which is diagonal with all 1
    except for the last term which is e^(ix))  to the target q[1], only if
     the control q[0] is 1
41          qc.h(j)
42      qc.name = "QFT^(-1)"
43      return qc
44 qft_dagger(4).draw(output='mpl', filename='Inverse QFT')
45
46
47 #Let's use 8 counting qubits (more counting qubits = bigger phase
    predictable) plus 4 qubits for U to act on
48 n_count = 8
49 qc = QuantumCircuit(n_count + 4, n_count) #The second argument indicates
    the number of bits (which will be used to store information of the
    measurements)
50 for q in range(n_count): #Applying the Hadamard gate to the counting
    qubits (at the begininning they are initialized to 0)
51     qc.h(q)
52 qc.x(3+n_count) #the last ancilla qubit becomes 1, in this way the ancilla
     register is initialized to 0001
53 for q in range(n_count):
54     qc.append(c_7mod15(2**(q)), [q] + [i+n_count for i in range(4)]) #
    Applies the gate c_7mod15(2**q) to the ancilla qubits [1+n_count], [2+
    n_count], [3+n_count], [4+n_count] utilizing the qubit [q] as control
55 qc.append(qft_dagger(n_count), range(n_count)) #Applies the gate
    qft_dagger(n_count) to the n_count qubits
56 for j in range (n_count):
57     qc.measure(range(n_count), range(n_count)) #Measure the n_count qubits
     storing information to the n_count bits
58 qc.draw(output='mpl', filename='Shor circuit implementation') #Displays
    the image of the circuit saving it as "Shor's circuit implementation"
59
60
61 #Define a function which simulate the circuit using a quantum computer
62 def shor_7mod15():
63     backend = Aer.get_backend('qasm_simulator') #Connecting to the Aer
    simulator of IBM
64     result = execute(qc, backend, shots = 1, memory=True).result() #
    Execute the circuit qc using the simulator backend 1 time; memory=True
     indicates that each individual shot is reported in a different list
65     readings = result.get_memory() #Gets the sequence of memory states (
    results) for each shot
66     print("Register Reading: " + readings[0]) #readings will be an array
    with only one element since I made only one shot
```

```
67      phase = int(readings[0], 2)/(2**n_count) #int(a, 2) returns the string
        "readings[0]" (which expresses a number in binary notation) converted
        to an integer in base-ten
68      print("Corresponding Phase: %f" % phase)
69      return phase
70
71
72  #Simulation
73  factor_found = False
74  attempt = 0
75  while not factor_found:
76      attempt += 1
77      print("Attempt %i" % attempt)
78      phase = shor_7mod15()
79      frac = Fraction(phase).limit_denominator(15) #Converts phase to the
        nearest fraction between two integers (we want the denominator to be
        less than 15 since the period P<15)
80      P = frac.denominator
81      print("Result: P = %i" % P)
82      if P % 2 != 0:
83          print("Unfortunately, the period isn't even")
84      else:
85          guesses = [gcd(7**(P//2)-1, 15), gcd(7**(P//2)+1, 15)]
86          print("Guessed Factors: %i and %i" %(guesses[0], guesses[1]))
87          for guess in guesses:
88              if guess != 1 and 15%guess == 0: #Checking if the guesses are
        non trivial factors of 15
89                  print("Non trivial factor found: %i" %guess)
90                  factor_found = True
91              else:
92                  print("Unfortunately, one of the factor that has been
        found is trivial")
```

Listato 7.2. Implementazione dell'algoritmo di Shor

Proviamo ora ad applicare tale algoritmo più volte (2048 volte per l'esattezza) e grafichiamo i risultati ottenuti aiutandoci con un istogramma e con due tabelle.

```
1  # -*- coding: utf-8 -*-
2  """
3  Shor's Algorithm
4
5  @author: mattia
```

```
"""

from qiskit.visualization import plot_histogram
import pandas as pd
import time

#Give a starting time to measure execution time
start = time.time()

#Let's see what results we measure
backend = Aer.get_backend('qasm_simulator') #Connecting to the Aer simulator of IBM
results = execute(qc, backend, shots = 2048).result() #Execute the circuit qc using the simulator backend 2048 time
counts = results.get_counts() #Creates an array: "results: number of times I got that result"
fig = plot_histogram(counts)
fig.savefig('istogramma.png', bbox_inches = 'tight') #The bbox command is necessary to make the figure of the right dimension
rows, measured_phases = [], []
for output in counts:
    decimal = int(output, 2) #convert base 2 string to decimal
    phase = decimal/(2**n_count)
    measured_phases.append(phase)
    rows.append(["%s(bin) = %i(dec)" %(output, decimal),
                 "%i/%i = %.2f" %(decimal, 2**n_count, phase)])

#Print the rows in a table
headers = ["Register Output", "Phase"]
df = pd.DataFrame(rows, columns = headers)
print(df)

#Creating a table with the guesses for P
rows = []
for phase in measured_phases:
    frac =  Fraction(phase).limit_denominator(15)
    rows.append([phase, "%i/%i" %(frac.numerator, frac.denominator), frac.denominator])
headers = ["Phase", "Fraction", "Guesses for P"]
df = pd.DataFrame(rows, columns = headers)
print(df)

end = time.time()
print("Execution time = %5.3f" %(end - start))
```

Listato 7.3. Iterazioni multiple dell'algoritmo

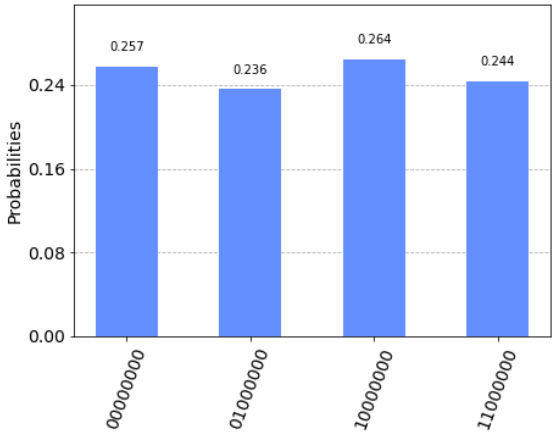

Figura 7.6. Istogramma rappresentativo delle misure effettuate dal simulatore quantistico

REGISTER OUTPUT	PHASE
00000000(bin) = 0(dec)	0/256 = 0.00
01000000(bin) = 64(dec)	64/256 = 0.25
10000000(bin) = 128(dec)	128/256 = 0.50
11000000(bin) = 192(dec)	192/256 = 0.75

PHASE	FRACTION	GUESSES FOR P
0.00	0/1	1
0.25	1/4	4
0.50	1/2	2
0.75	3/4	4

Figura 7.7. Tabelle riassuntive dei risultati ottenuti applicando l'algoritmo ripetutamente

Ripetendo 10 volte il Listato 7.3 si è ottenuto un tempo di esecuzione medio pari a 5.281 secondi. Sebbene il numero da fattorizzare fosse piccolo ($N = 15$ e il tempo di esecuzione dell'algoritmo è lineare al crescere di N come mostrato in §6), il Listato 7.3 ripete la fattorizzazione ben 2048 volte. Ciò mette in luce le potenzialità dell'algoritmo di Shor.

Bibliografia

[1] Alessandra Di Pierro. *Quantum Computing*. 2002. URL: http://profs.sci.univr.it/~dipierro/InfQuant/articles/Lezioni-IQ.pdf.

[2] Samuel J. Lomonaco Jr. *A Lecture on Shor's Quantum Factoring Algorithm*. 2000. URL: https://arxiv.org/abs/quant-ph/0010034.

[3] Enrico Onofri. *Lezioni sulla Teoria degli Operatori Lineari*. 1984.

[4] Majit Kumar. *Quantum. Da Einstein a Bohr, la teoria dei quanti, una nuova idea della realtà*. Mondadori, 2011.

[5] Niels Bohr. *Essays on Atomic Physics and Human knowledge*. 1957.

[6] Albert Einstein. *Handwritten letter to Paul Epstein*. 1945.

[7] M.S. Manesse Lenstra A.K. H.W. Lenstra Jr. e J.M. Pollard. *The number field sieve*. A cura di Springer-Velag. 1993, pp. 564–572.

[8] Charles E. Leiserson Cormen Thomas H. e L. Rivest. *Introduction to Algorithms*. A cura di McGraw-Hill. 1990.

[9] E.M. Wright Hardy G.H. *An Introduction to the Theory of Numbers*. A cura di Oxford Press. 1965.

[10] URL: https://qiskit.org.