Quantum Computing

Alessandra Di Pierro

Il Modello dei Circuiti per la computazione quantistica

1 Modelli di computazione classica e universalitá

L'esigenza di formalizzare l'idea intuitiva di funzione *calcolabile* tramite un algoritmo (o *funzione algoritmica*) diede origine nella prima metá del ventesimo secolo alla teoria della calcolabilitá che nasce con le varie proposte di *modelli di calcolo* da parte di Kleene (1936), Church (1936) e Turing (1936). Negli anni successivi furono proposte altre caratterizzazioni formali, tra cui ad esempio quelle di Post (1943) e Markov (1954). La definizione formale di algoritmo dipende ovviamente dal modello di calcolo adottato e puó avere quindi formulazioni molto diverse l'una dall'altra.

Un risultato fondamentale della teoria della calcolabilitá afferma che tutte queste diverse caratterizzazioni sono equivalenti: **ogni funzione calcolabile in un certo modello di calcolo formale, é calcolabile anche negli altri**, purché opportunamente codificata in accordo ai diversi formalismi. In particolare, la *Tesi di Church-Turing* afferma che ogni funzione calcolabile si puó caclolare nel formalismo di Turing, basandosi sull'evidenza che non esiste un concetto di calcolabilitá piú ampio di quello di Turing.

Esamineremo successivamente piú in dettaglio il formalismo di Turing e il concetto di universalitá associato, cioé la Macchina di Turing Universale, mentre ci concentreremo ora su un altro modello di calcolo che risulta in alcuni casi piú utile come modello di riferimento per lo studio della complessitá computazionale. Questo modello computazionale alternativo per la computazione classica é basato sui *circuiti booleani.* In questo caso l'universalitá del modello si esprime individuando un insieme di circuiti elementari mediante i quali é possibile esprimere qualsiasi altro circuito. Tale insieme si dice una *base completa.* Come é noto i circuiti elementari che calcolano le funzioni booleane NOT, OR e AND formano una base completa per la computazione classica [1].

L'obiettivo di questa lezione é di definire in analogia con il caso classico un modello astratto per la computazione quantistica e far vedere che, come per la computazione classica, esiste anche per la computazione quantistica una nozione di universalitá.

2 Circuiti quantistici: realizzazione esatta

Il modello di computazione basato sui circuiti quantistici é stato introdotto da Deutsch (1989) e sviluppato ulteriormente da Yao (1993) che ne dimostra anche l'equivalenza con un altro modello per la computazione quantistica: la Macchina di Turing quantistica. Quest'ultima é stata introdotta da Benioff (1980) e successivamente sviluppata da Deutsch (1985) e Yao (1993). L'articolo di Bernstein e Vazirani *Quantum Complexity Theory* (1997) contiene una descrizione completa e moderna della macchina di Turing quantistica. In questo articolo viene affrontato il tema della complessitá computazionale dimostrando per la computazione quantistica gli analoghi dei risultati della complessitá computazionale classica.

Questa lezione é dedicata al modello dei circuiti, mentre si consiglia fortemente la lettura dell'articolo di Bernstein e Vazirani[1] per una descrizione del modello basato sulla Macchina di Turing quantistica.

Dal momento che si possono definire infiniti operatori unitari (e quindi infinite operazioni quantistiche), la questione che si pone é come definire una base completa per la computazione quantistica. Ci sono due possibilitá:

- considerare un insieme infinito di gates,
- considerare un insieme finito e sostituire la condizione di realizzazione esatta dalla condizione di realizzazione *approssimata*.

Esamineremo nel seguito la prima possibilitá, mentre la seconda sará discussa in 3.

2.1 Operazioni su un qubit

Le operazioni quantistiche piú elementari sono quelle su un singolo qubit. Ricordiamo che un qubit é un vettore $|\psi\rangle = a|0\rangle + b|1\rangle$ nello spazio di Hilbert $\mathbb{C}^2$, tale che $|a|^2 + |b|^2 = 1$. Le operazioni su un qubit devono preservare questa norma e sono pertanto descritte da matrici unitarie 2×2. Tra queste

[1] `www.di.unipi.it/~dipierro/QC.htm#reference`

abbiamo visto le matrici di Pauli:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \text{ e } Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

spesso denotate anche con σ_x, σ_y e σ_z. Un ruolo molto importante nella definizione del modello dei circuiti é svolto anche dalle seguenti operazioni:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},\ S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \text{ e } T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}.$$

Abbiamo giá visto la porta di Hadamard H. Il gate S é detto gate di fase, mentre il gate T é anche chiamato gate $\pi/8$ perché risulta

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

Si verifica facilmente che $H = (X + Z)/\sqrt{2}$, $S = T^2$ e $S = \sqrt{Z}$.

Ricordiamo inoltre che lo stato di un qubit $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$ puó essere visualizzato come un punto (θ, φ) sulla sfera unitaria in $\mathbb{R}^3$ (la sfera di Bloch) dato che esiste una corrispondenza biunivoca tra qubits e questi punti. Analogamente, le operazioni su un qubit (cioé le matrici unitarie 2×2) corrispondono alle *rotazioni* del punto che rappresenta il qubit sulla sfera di Bloch. In generale il teorema di Bloch (cf. Teorema 2.4) assicura che qualsiasi matrice unitaria 2×2 si puó decomporre in un prodotto di rotazioni nel piano $(\vec{x}, \vec{y})$ e rotazioni intorno all'asse $\vec{z}$. Le prime sono descritte da matrici della forma

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix},$$

mentre, come vedremo, le seconde corrispondono a operazioni unitarie della forma

$$\begin{bmatrix} e^{-i\beta/2} & 0 \\ 0 & e^{i\beta/2} \end{bmatrix}.$$

Per poter enunciare questo teorema abbiamo bisogno di introdurre la nozione di *esponenziale di una matrice* e la definizione di *operatore di rotazione* intorno agli assi $\vec{x}$, $\vec{y}$ e $\vec{z}$.

2.1.1 Esponenziale di una matrice

In generale, l'esponenziale e^A di una matrice complessa $n \times n$, A, é definita mediante la serie di potenze

$$e^A = \sum_{m=0}^{\infty} \frac{A^m}{m!}.$$

Si dimostra che questa serie converge e che e^A é una funzione continua di A. Si dimostrano inoltre le seguenti proprietá:

1. $e^0 = I$
2. e^A é invertibile e $(e^A)^{-1} = e^{-A}$
3. $e^{(a+b)A} = e^{aA}e^{bA}$, per ogni $a, b \in \mathbb{C}$
4. se $AB = BA$, allora $e^{A+B} = e^A e^B = e^B e^A$
5. se C é invertibile, allora $e^{CAC^{-1}} = Ce^AC^{-1}$
6. $\|e^A\| \leq e^{\|A\|}$.

Se A é diagonalizzabile possiamo calcolare esplicitamente e^A nel seguente modo[2]. Poiché A é diagonalizzabile esiste una matrice invertibile C tale che $A = CDC^{-1}$, dove

$$D = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix}.$$

Osserviamo che e^A é la matrice diagonale con autovalori e^{λ_i}; quindi per la proprietá 5, otteniamo

$$e^A = C \begin{bmatrix} e^{\lambda_1} & & 0 \\ & \ddots & \\ 0 & & e^{\lambda_n} \end{bmatrix} C^{-1}.$$

Esempio 2.1 *Consideriamo la matrice*

$$A = \begin{bmatrix} 0 & -a \\ a & 0 \end{bmatrix}.$$

[2]Nota che le nostre operazioni su qubits sono matrici unitarie e quindi in particolare operatori normali e ammettono pertanto una scomposizione spettrale, cioé sono diagonalizzabili.

A ha autovettori $\begin{bmatrix} 1 \\ i \end{bmatrix}$ *e* $\begin{bmatrix} i \\ 1 \end{bmatrix}$ *con corrispondenti autovalori $-ia$ e ia. Quindi la matrice invertibile* $C = \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix}$ *trasforma i vettori della base* $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ *e* $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ *negli autovettori di A. Si verifica facilmente che la matrice $D = C^{-1}AC$ che rappresenta A nella nuova base é una matrice diagonale. Dalla relazione $A = CDC^{-1}$ e la proprietá (5) dell'esponenziale di una matrice si ottiene:*

$$e^A = \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix} \begin{bmatrix} e^{-ia} & 0 \\ 0 & e^{ia} \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{-i}{2} \\ \frac{-i}{2} & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \cos a & -\sin a \\ \sin a & \cos a \end{bmatrix}.$$

2.1.2 Operatori di rotazione

Gli esponenziali delle matrici di Pauli danno luogo a tre utili classi di matrici unitarie: gli *operatori di rotazione* intorno agli assi $\vec{x}$, $\vec{y}$ e $\vec{z}$ definiti come:

$$R_x(\theta) \equiv e^{-i\theta\sigma_x/2} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}\sigma_x = \begin{bmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{bmatrix} \quad (1)$$

$$R_y(\theta) \equiv e^{-i\theta\sigma_y/2} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}\sigma_y = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix} \quad (2)$$

$$R_z(\theta) \equiv e^{-i\theta\sigma_z/2} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}\sigma_z = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}. \quad (3)$$

Esercizio 2.2 *Verifica che $XYX = -Y$ e $XZX = -Z$. Quindi dimostra che $XR_y(\theta)X = R_y(-\theta)$ e $XR_z(\theta)X = R_z(-\theta)$.*

Possiamo usare il procedimento visto nell'Esempio 2.1 per calcolare esplicitamente l'esponenziale delle matrici di Pauli e verificare le espressioni (1), (2) e (3). Per esempio, per calcolare l'esponenziale $e^{-i\theta\sigma_z/2}$, ricordiamo che σ_z ha autovettori $|0\rangle$ e $|1\rangle$ con corrispondenti autovalori 1 e -1, quindi in questo caso $C = I$ e pertanto

$$e^{-i\theta\sigma_z/2} = I \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix} I.$$

Analogamente, si calcolano gli esponenziali per σ_x e σ_y ricordando che gli autovettori sono:

$$|x_1\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \text{ e } |x_2\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \text{ per } \sigma_x,$$

$$|y_1\rangle = \frac{|0\rangle + i\,|1\rangle}{\sqrt{2}} \text{ e } |y_2\rangle = \frac{|0\rangle - i\,|1\rangle}{\sqrt{2}} \text{ per } \sigma_y,$$

corrispondenti rispettivamente agli autovalori 1 e -1. Si noti che in questo caso la matrice del cambio di base C é $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ per σ_x, e $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix}$ per σ_y.

Un modo alternativo per verificare le equazioni (1), (2) e (3) é usare la seguente proprietá che é una diretta conseguenza della definizione di esponenziale come serie di potenze data precedentemente .

Proposizione 2.3 *Dato $x \in \mathbb{R}$ e A una matrice tale che $A^2 = I$, allora*

$$e^{iAx} = \cos xI + i \sin xA.$$

Dimostrazione Per definizione

$$e^{iAx} = \sum_{m=0}^{\infty} \frac{(ix)^m}{m!} A^m.$$

Dall'ipotesi $A^2 = I$, si ha quindi, separando le somme sugli indici pari e dispari:

$$\begin{aligned} e^{iAx} &= \sum_{m=0}^{\infty} \frac{(ix)^{2m}}{(2m)!} I + \sum_{m=0}^{\infty} \frac{(ix)^{2m+1}}{(2m+1)!} A \\ &= \sum_{m=0}^{\infty} \frac{(-1)^m x^{2m}}{(2m)!} I + i \sum_{m=0}^{\infty} \frac{(-1)^m x^{2m+1}}{(2m+1)!} A \end{aligned}$$

Osserviamo ora che le due somme sono lo sviluppo in serie di potenze per $\cos x$ e $\sin x$ rispettivamente. □

Le equazioni (1), (2) e (3) si ottengono direttamente dalla Proposizione 2.3 notando che per le matrici di Pauli vale la condizione della proposizione, cioé $\sigma_x^2 = \sigma_y^2 = \sigma_z^2 = I$.

Se $\hat{n} = (n_x, n_y, n_z)$ é un vettore unitario nello spazio $\mathbb{R}^3$, allora possiamo generalizzare le definizioni di R_x, R_y e R_z per definire una rotazione di θ intorno all'asse $\hat{n}$:

$$R_{\hat{n}}(\theta) \equiv e^{-i\theta\hat{n}\cdot\vec{\sigma}/2} = \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2}(n_x\sigma_x + n_y\sigma_y + n_z\sigma_z),$$

dove $\vec{\sigma} = \sigma_x + \sigma_y + \sigma_z$.

Possiamo ora enunciare il teorema di Bloch:

Teorema 2.4 *Per ogni matrice unitaria 2×2, U esistono numeri reali α, β, γ e δ tali che*

$$U = e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta).$$

Dimostrazione

Come visto nell'esercizio 1.1 dell'Esercitazione 1 (vedi Esercizi di Riepilogo), esistono α, β, δ e γ tali che

$$U = \begin{bmatrix} e^{i(\alpha-\beta/2-\delta/2)} \cos(\frac{\gamma}{2}) & -e^{i(\alpha-\beta/2+\delta/2)} \sin(\frac{\gamma}{2}) \\ e^{i(\alpha+\beta/2-\delta/2)} \sin(\frac{\gamma}{2}) & e^{i(\alpha+\beta/2+\delta/2)} \cos(\frac{\gamma}{2}) \end{bmatrix}.$$

Si verifica che questa matrice é esattamente il risultato della moltiplicazione $e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta)$. □

Il seguente corollario sará utile nel seguito per dimostrare il risultato di universalitá in 2.3.

Corollario 2.5 *Supponiamo che U sia un'operazione unitaria su un singolo qubit. Allora esistono operatori unitari su un singolo qubit A, B e C tali che $ABC = I$ e $U = e^{i\alpha} AXBXC$, per qualche $\alpha \in \mathbb{R}$.*

Dimostrazione Considera i valori α, β, δ e γ del Teorema 2.4 e definisci

$$A = R_z(\beta) R_y(\frac{\gamma}{2}),\ B = R_y(-\frac{\gamma}{2}) R_z(-\frac{\delta+\beta}{2}),$$
$$\text{e } C = R_z(\frac{\delta-\beta}{2}).$$

Si verifica facilmente che $ABC = I$. Dall'esercizio 2.2 e dal fatto che $X^2 = I$ si ottiene:

$$XBX = XR_y(-\frac{\gamma}{2}) X^2 R_z(-\frac{\delta+\beta}{2}) X = R_y(\frac{\gamma}{2}) R_z(\frac{\delta+\beta}{2}).$$

Pertanto si ha che

$$\begin{aligned} AXBXC &= R_z(\beta) R_y(\tfrac{\gamma}{2}) R_y(\tfrac{\gamma}{2}) R_z(\tfrac{\delta+\beta}{2}) R_z(\tfrac{\delta-\beta}{2}) \\ &= R_z(\beta) R_y(\gamma) R_z(\delta), \end{aligned}$$

e quindi dal Teorema 2.4

$$U = e^{i\alpha} AXBXC.$$

□

Le seguenti identitá sono utili e facili da verificare:

$$HXH = Z;\ HYH = -Y;\ HZH = X.$$

La terza identitá ci permette di dimostrare la seguente proposizione

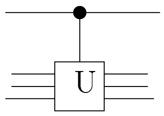

Figure 1: Operazione controllata su singolo qubit

Proposizione 2.6 *$HTH = R_x(\pi/4)$ a meno di un fattore di fase globale.*

Dimostrazione
Per definizione di operatore di rotazione e dalla identitá $HZH = X$, si ha che $R_x(\pi/4) = e^{-i\frac{\pi}{8}\sigma_x} = e^{-i\frac{\pi}{8}H\sigma_z H}$. Poiché H é invertibile, per la proprietá (5) dell'esponeziale di una matrice (cf. 2.1.1) otteniamo

$$e^{-i\frac{\pi}{8}H\sigma_z H} = H \begin{bmatrix} e^{-i\frac{\pi}{8}} & 0 \\ 0 & e^{i\frac{\pi}{8}} \end{bmatrix} H \approx H e^{i\frac{\pi}{8}} \begin{bmatrix} e^{-i\frac{\pi}{8}} & 0 \\ 0 & e^{i\frac{\pi}{8}} \end{bmatrix} H = HTH.$$

□

Esercizio 2.7 *Dimostrare che un arbitrario operatore unitario su singolo qubit puó essere scritto nella forma*

$$U = e^{i\gamma} R_{\hat{n}}(\alpha),$$

dove $\gamma, \alpha \in \mathbb{R}$ e $\hat{n}$ é un vettore unitario in $\mathbb{R}^3$.

2.2 Operazioni controllate

Il prototipo di queste operazioni é il `CNOT` (controlled-`NOT`). Come abbiamo giá visto, `CNOT` agisce su due qubits chiamati rispettivamente *controllo* e *target* e che il suo effetto nella base computazionale é dato da $|c\rangle |t\rangle \rightarrow |c\rangle |t \oplus c\rangle$.

In generale, si puó definire un'operazione controllata a partire da una qualsiasi operazione unitaria U mediante il circuito in Figura 1. Se U agisce su un singolo qubit, il circuito trasforma i qubits in input $|c\rangle |t\rangle$ in $|c\rangle U^c |t\rangle$, cioé U é applicata al qubit target soltanto se il qubit di controllo é 1.

Il circuito in Figura 2 rappresenta l'operazione controllata piú generale $C^n(U)$, dove n é il numero dei qubits di controllo e U é una operazione unitaria su k qubits per qualche intero k.

Le operazioni controllate possono essere implementate usando solo `CNOT` e operazioni unitarie su un singolo qubit che, come vedremo in 2.3, costituisce una *base completa* di cardinalitá *infinita* per la computazione quantistica. Un

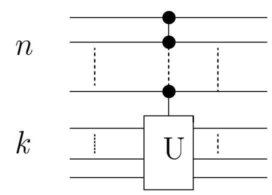

Figure 2: Controlled-U generale con n qubit di controllo e operazione U su k qubit

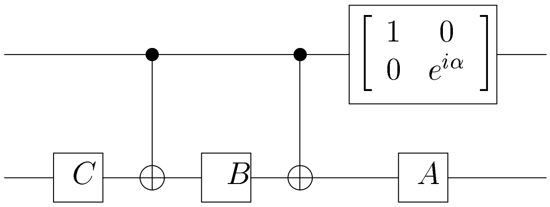

Figure 3: Circuito che implementa controlled-U per operazioni U su singolo qubit.

aspetto fondamentale della computazione quantistica é l'implementazione fault-tolerant (cioé resistente agli errori) delle varie operazioni. Esiste un insieme discreto di operazioni elementari che é universale e puó essere implementato in modo robusto usando dei codici di correzione degli errori. Come vedremo in 3, questo insieme universale discreto é formato da Hadmard, `CNOT`, S e T. Poiché l'insieme delle operazioni unitarie é continuo, l'universalitá di questo insieme discreto deve essere intesa nel senso che le quattro operazioni che lo compongono possono essere usate per *approssimare* una qualsiasi operazione unitaria, piuttosto che per implementarla esattamente. Il significato di tale approssimazione verrá chiarito in 3.

2.2.1 Implementazione di $C^n(U)$ mediante `CNOT` e operazioni su singolo qubit

Consideriamo prima il caso piú semplice dove $n = 1$ e U é un'operazione unitaria su un singolo qubit, cioé circuiti come quello in Figura 1 dove U ha un solo qubit di input. Per il Corollario 2.5, esistono matrici unitarie su un singolo qubit A, B, C tali che $U = e^{i\alpha}AXBXC$. Allora possiamo costruire un circuito equivalente usando solo `CNOT`, le operazioni A, B, C e l'operazione che realizza la moltiplicazione controllata del qubit target per $e^{i\alpha}$ come mostrato in Figura 3.

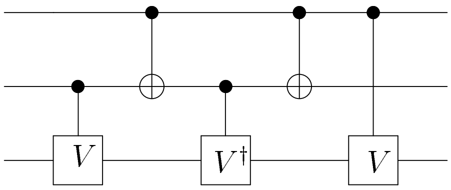

Figure 4: Circuito che realizza $C^2(U)$ per $U = V^2$

L'operazione

$$\alpha = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{bmatrix}$$

applicata al primo qubit effettivamente realizza la moltiplicazione per $e^{i\alpha}$ se il qubit di controllo é 1. Infatti, il suo effetto sulla base computazionale é:

$$|0\rangle \mapsto |0\rangle\,, \quad |1\rangle \mapsto e^{i\alpha}\,|1\rangle\,.$$

Quindi il circuito in Figura 3 opera nel seguente modo[3]:

$$\begin{gathered} |0\rangle \otimes |x\rangle \mapsto |0\rangle \otimes ABC\,|x\rangle = |0\rangle \otimes |x\rangle\,, \\ |1\rangle \otimes |x\rangle \mapsto e^{i\alpha}\,|1\rangle \otimes AXBXC\,|x\rangle = e^{i\alpha}\,|1\rangle \otimes e^{-i\alpha}U\,|x\rangle = |1\rangle \otimes U\,|x\rangle\,. \end{gathered}$$

Consideriamo ora il caso in cui $n = 2$ e U é ancora un operatore unitario su un singolo qubit. Allora il circuito in Figura 4 implementa $C^2(U)$ nell'ipotesi che esista un operatore unitario V tale che $V^2 = U$.

Per $U = X$ si ottiene l'operazione di Toffoli che, vista come operazione classica, é universale per le computazioni classiche reversibili. Ricordiamo che questo risultato di universalitá non si poteva ottenere con un numero di input inferiore a tre. Al contrario, l'operazione di Toffoli si puó implementare come operazione quantistica usando solo operazioni su uno o due qubit. Tale implementazione si ottiene mediante il circuito in Figura 4 con $V = (1 - i)(I + iX)/2$.

Possiamo ora considerare il caso di n qubits di controllo e ancora un solo qubit target, cioé l'operazione $C^n(U)$ definita dall'equazione

$$C^n(U)\,|c_1c_2\dots c_n\rangle\,|\psi\rangle = |c_1c_2\dots c_n\rangle\,U^{c_1c_2\dots c_n}\,|\psi\rangle\,.$$

L'operatore U é applicato al qubit target $|\psi\rangle$ se i primi n qubits sono tutti 1 (l'esponente di U nell'equazione rappresenta un `AND` dei bit c_i). Un circuito che implementa $C^n(U)$ si puó ottenere usando l'operazione di Toffoli per realizzare l'`AND` reversibile dei bit di controllo come mostrato in Figura 5.

[3]Per le trasformazioni lineari basta verificarne l'effetto sulla base computazionale

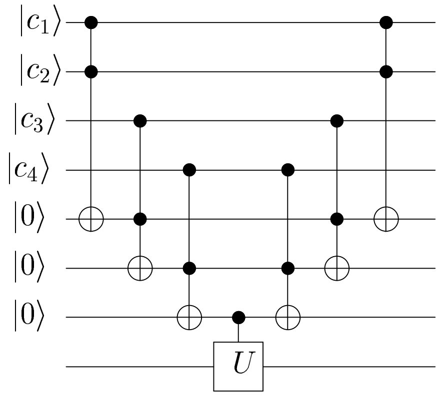

Figure 5: Circuito che realizza $C^n(U)$ per $n = 4$ e $k = 1$

Esercizio 2.8 *Dimostra che ogni operazione $C^n(U)$, dove U é una matrice unitaria su un singolo qubit, puó essere implementata usando $O(n)$ operazioni su singolo qubit e* `CNOT`.

2.3 Universalitá di `CNOT` e operazioni su singolo qubit

Qualsiasi operatore unitario si puó esprimere in maniera *esatta* mediante `CNOT` e operazioni su singolo qubit. Per dimostrarlo dobbiamo completare i risultati visti precedentemente per le operazioni controllate, considerando operazioni unitarie U su un numero arbitrario $k \geq 2$ di qubits. L'idea fondamentale é quella di scomporre una U arbitraria nel prodotto di matrici unitarie che operano in maniera non banale su al piú due componenti del vettore a cui sono applicate. Poiché tali matrici, dette *matrici unitarie a due livelli*, si possono implementare mediante circuiti costruiti usando `CNOT` e operazioni su singolo qubit, potremo quindi concludere il risultato di universalitá sopra citato.

Dato un operatore U su uno spazio d-dimensionale, é sempre possibile mediante una procedura puramente algebrica costruire matrici unitarie a due livelli $V_1, V_2 \dots V_{d'}$ tali che $U = V_1 V_2 \dots V_{d'}$ e $d' \leq d(d-1)/2$.

Quindi non ci rimane altro che far vedere come costruire un circuito che implementa una matrice unitaria a due livelli ed é formato solo da `CNOT` e operazioni su singolo qubit.

Supponiamo che U sia una matrice unitaria a due livelli che opera su n qubits e supponiamo che gli stati su cui agisce effettivamente siano tutti quelli generati dai due stati $|s\rangle$ e $|t\rangle$ della base computazionale, dove s e t sono le rappresentazioni binarie dei numeri $s, t \in [0, n-1]$. Allora la sottomatrice non banale $\tilde{U}$ di U é una matrice unitaria 2×2 e si puó vedere come un operatore unitario su un singolo qubit. Per costruire il nostro circuito consideriamo la sequenza $g_1, g_2, \ldots, g_m$ di numeri binari tali che $g_1 = s$, $g_m = t$ e per ogni $1 \leq i < m$, g_i e g_{i+1} differiscono per esattamente un bit. Tali sequenze sono dette *codici di Gray*. Ad esempio 1011, 1001, 0001, 0000 é un codice di Gray che connette 1011 e 0000.

Usando operazioni controllate si realizza il passaggio da g_i a g_{i+1} per $1 \leq i < m-1$. A questo punto si esegue un controlled-$\tilde{U}$ con qubit target corrispondente al singolo bit su cui g_{m-1} e g_m differiscono e con i qubits di controllo corrispondenti ai valori dei bits che sono identici in g_{m-1} e g_m. Le operazioni di passaggio da g_i a g_{i+1} devono essere realizzate in modo reversibile e quindi in generale saranno necessarie $2(n-1)$ operazioni controllate per implementarle.

Esempio 2.9 *Considera la matrice su tre qubits definita da*

$$U = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a & 0 & 0 & 0 & 0 & c \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & b & 0 & 0 & 0 & 0 & d \end{bmatrix},$$

dove $\tilde{U} = \begin{bmatrix} a & c \\ b & d \end{bmatrix}$ *é una arbitraria matrice unitaria* 2×2*. Notiamo che* U *agisce effettivamente solo sugli stati (generati da)* $|s\rangle = |010\rangle$ *e* $|t\rangle = |111\rangle$*. Il circuito che implementa* U *é dato in Figura 6. Gli input* A, B, C *sono inizialmente nella configurazione corrispondente a* s*, cioé* 010*. Lo stato successivo della sequenza* $\{g_i\}$ *é* 011 *e il passaggio é effettuato da un* `NOT` *su* C *condizionato dai valori dei bit* $A = 0$ *e* $B = 1$ *(ricordiamo che il pallino bianco indica che il target viene modificato se il controllo é* 0*, in maniera duale al pallino nero). Lo stato successivo é lo stato finale* $|t\rangle = |111\rangle$*. Quindi applichiamo* $\tilde{U}$ *ad* A *a condizione che* $B = 1$ *e* $C = 1$*. La parte destra del circuito riproduce simmetricamente la parte sinistra per ottenere la reversibilitá.*

Nota 2.10 *La procedura descritta corrisponde essenzialmente a trovare un cambio nella base computazionale tale che i due livelli non banali corrispondano a un singolo qubit. Ricordiamo che la matrice di rappresentazione nella*

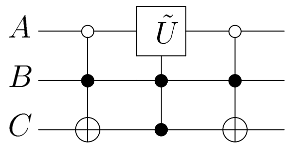

Figure 6: Circuito che implementa la matrice a due livelli U dell'esempio 2.9

nuova base é $V = BUB^{-1}$*, dove* B *é la matrice del cambio di base, da cui si ottiene* $U = B^{-1}VB$*. Individuare un codice di Gray e implementare le trasformazioni descritte dall'algoritmo corrisponde essenzialmente a costruire* B *e la sua inversa.*

Dall'esercizio 2.8 sappiamo che ciascuna delle operazioni controllate necessarie per effettuare il passaggio da g_1 a g_m in modo reversibile richiede $O(n)$ operazioni su singolo qubit e `CNOT`. Abbiamo visto inoltre che tale passaggio richiede almeno $2(n-1)$ operazioni controllate. Quindi per implementare una matrice unitaria a due livelli sono necessarie $O(n^2)$ operazioni su singolo qubit e `CNOT`.

In conclusione, possiamo implementare una generica operazione unitaria su n qubits come composizione di al piú $2^n(2^n-1)/2$, cioé $O(2^{2n})$, operazioni unitarie a due livelli, e quindi con $O(n^2 2^{2n})$ operazioni su singolo qubit e `CNOT`.

Naturalmente questa implementazione é estremamente inefficiente. In pratica si usa un'altra tecnica che permette di utilizzare un insieme discreto di gates per ottenere in maniera efficiente un'implementazione non piú esatta ma *approssimata* di una qualsiasi trasformazione unitaria su due o piú qubits.

3 Circuiti quantistici: realizzazione approssimata

3.1 Universalitá di Hadamard, `CNOT`, S e T

Le operazioni Hadamard, `CNOT`, S e T formano un insieme universale *finito* per la computazione quantistica nel senso che data un'operazione unitaria arbitraria, é possibile simularla con una "buona approssimazione" mediante un circuito che contiene un numero finito di queste operazioni.

Per definire la nozione di approssimazione abbiamo bisogno di introdurre la definizione di norma di un operatore.

Definizione 3.1 *Dato un vettore* $|\psi\rangle \in \mathbb{C}^n$*, consideriamo la norma euclidea* *definita da* $\sqrt{\langle\psi|\psi\rangle}$. *La norma di un operatore* $L : \mathbb{C}^n \to \mathbb{C}^n$ *é definita da:*

$$\|L\| = sup_{|\psi\rangle\neq 0}\frac{\|L\,|\psi\rangle\,\|}{\|\,|\psi\rangle\,\|}.$$

Per definizione di norma, la norma di un operatore gode delle seguenti proprietá:

$$\begin{aligned}
&\|L\| \geq 0\\
&\|L\| \geq 0 \text{ sse } L = 0\\
&\|L + V\| \leq \|L\| + \|V\|\\
&\|cL\| = c\|L\|.
\end{aligned}$$

Si verificano inoltre le seguenti importanti proprietá:

(i) $\|LV\| \leq \|L\| \cdot \|V\|$,

(ii) $\|L^\dagger\| = \|L\|$,

(iii) $\|L \otimes V\| = \|L\| \cdot \|V\|$,

(iv) $\|U\| = 1$ se U é unitaria.

Possiamo ora dare la definizione di realizzazione approssimata di un operatore U, che denoteremo con $\tilde{U}$.

Definizione 3.2 *L'operatore* $\tilde{U}$ approssima l'operatore U con precisione δ *se*

$$\|\tilde{U} - U\| \leq \delta.$$

Dalle proprietá (i) e (iv) segue che se $\tilde{U}$ approssima U con precisione δ, allora $\tilde{U}^{-1}$ approssima U^{-1} con la stessa precisione δ.

Un'altra importante conseguenza é che gli errori si accumulano linearmente: se $U = U_m \cdots U_2 U_1$ e ogni U_k ha un'approssimazione $\tilde{U_k}$ con precisione δ_k, allora il prodotto di queste approssimazioni $\tilde{U} = \tilde{U_m} \cdots \tilde{U_2}\tilde{U_1}$ approssima U con precisione $\sum_k \delta_k$.

Il risultato di universalitá sopra citato é conseguenza del fatto che le due operazioni T e HTH possono essere usate per approssimare una qualsiasi operazione unitaria su singolo qubit con un errore arbitrariamente piccolo.

Ricordiamo che, a meno di un fattore di fase globale, T corrisponde a una rotazione di $\pi/4$ intorno all'asse $\hat{z}$ e HTH a una rotazione di $\pi/4$ intorno all'asse $\hat{x}$ (cf. Proposizione 2.6). Componendo queste due operazioni si ottiene:

$$\begin{aligned} e^{-i\frac{\pi}{8}\sigma_x}e^{-i\frac{\pi}{8}\sigma_z} &= (\cos\frac{\pi}{8}I - i\sin\frac{\pi}{8}\sigma_x)(\cos\frac{\pi}{8}I - i\sin\frac{\pi}{8}\sigma_z) \\ &= \cos^2\frac{\pi}{8}I - i(\cos\frac{\pi}{8}(\sigma_x+\sigma_z) + i\sigma_x\sigma_z\sin\frac{\pi}{8})\sin\frac{\pi}{8} \\ &= \cos^2\frac{\pi}{8}I - i(\cos\frac{\pi}{8}(\sigma_x+\sigma_z) + \sin\frac{\pi}{8}\sigma_y)\sin\frac{\pi}{8} \end{aligned}$$

Ricordando che $R_{\hat{n}}(\theta) = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}(n_x\sigma_x + n_y\sigma_y + n_z\sigma_z)$, l'espressione ottenuta é quindi una rotazione intorno all'asse $\vec{n} = (\cos\frac{\pi}{8}, \sin\frac{\pi}{8}, \cos\frac{\pi}{8})$, con corrispondente vettore unitario $\hat{n}$, di un angolo θ definito da $\cos\theta/2 = \cos^2\frac{\pi}{8}$. Iterando l'applicazione di $R_{\hat{n}}(\theta)$ per un numero sufficiente di volte si puó approssimare una qualsiasi rotazione $R_{\hat{n}}(\alpha)$ in modo da ottenere un arbitrario livello di precisione δ.

Quindi, ricordando il risultato dimostrato nell'esercizio 2.7, dato un operatore unitario U su singolo qubit e un $\delta > 0$, é possibile approssimare U con precisione δ usando un circuito composto soltanto da operazioni T e H (il gate S non é strettamente necessario per questo risultato di universalitá, ma viene usato per effettuare approssimazioni fault-tolerant).

In conclusione, dato un circuito che contiene solo `CNOT` e operazioni unitarie su un singolo qubit (che formano una base completa, come visto in 2.3), possiamo simulare questo circuito con un'approssimazione fissata ma arbitraria, usando solo Hadamard, S, T e `CNOT`. Inoltre tale simulazione é efficiente nel senso che puó essere eseguita in $\log m/\delta$, dove m é il numero di gates nel circuito originario e δ é l'accuratezza desiderata (cf. Teorema di Solovay-Kitaev, Appendice 3 del libro di testo [3]).

Un altro insieme finito di gates per il quale si dimostra un analogo risultato di universalitá é formato da Hadamard, `CNOT`, S e l'operazione $C^2(X)$ (gate di Toffoli). Una dimostrazione si trova in [2].

References

[1] A. Bernasconi and B. Codenotti. *Introduzione alla Complessitá Computazionale.* Springer-Verlag, Milano, Italy, 1998.

[2] A.Yu. Kitaev, A.H. Shen, and M.N. Vyalyi. *Classical and Quantum Computation.* Cambridge University Press, Cambridge, UK, 2000.

[3] M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information.* AMS, Providence, Rhode Island, 2002.