UNIVERSITAT DE BARCELONA

Treball final de grau

GRAU DE MATEMÀTIQUES

Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

INTRODUCCIÓ A LA COMPUTACIÓ QUÀNTICA

Autor: Naila Ortiz Clos

Directora: **Dra. Maite López-Sánchez**
Realitzat a: **Departament de Matemàtiques i Informàtica**

Barcelona, **13 de setembre de 2020**

Abstract

Starting with no knowledge about what quantum computing is and how it works, this paper will study and review the concepts needed to understand how it works, know of what consists , know how can be compared to classical computing and come to understand how a quantum computer is programmed.

From the most basic concepts of mathematics, through the theoretical concepts of classical computing, to the postulates that define quantum mechanics, to finally study classical computing and applying its concepts programmed in a quantum computer.

Resum

Començant amb un coneixement nul sobre el què és i com funciona la computació quàntica, en aquest treball s'estudiaran i repassaran els conceptes necessaris per arribar a compendre com funciona, saber en què consisteix, saber en què es pot comparar amb la computació clàssica i arribar a entendre com es programa un ordinador quàntic.

Des dels conceptes més bàsics de matemàtiques, passant pels conceptes teòrics de la computació clàssica, fins els postulats que defineixen la mecànica quàntica, per acabar estudiant finalment la computació clàssica i aplicant els seus conceptes programats en un ordinador quàntic.

Agraïments

Vull agrair a l'Enric Delgado per assessorar-me en un principi, a la meva tutora la Dra Maite López-Sánchez per la seva ajuda i assessorament durant la realització del projecte i finalment a tots els meus familiars, amics i companys que m'han recolzat i ajudat durant aquest temps.

Índex

Introducció **iii**

1 Fonaments matemàtics **1**

1.1 Nombres complexos 1

1.1.1 L'àlgebra dels nombres complexes 2

1.1.2 La geometria dels nombres complexes 4

1.2 Àlgebra lineal 5

1.2.1 Bases dels espais vectorials 7

1.2.2 Matrius 8

1.2.3 Aplicacions lineals 10

1.2.4 Producte escalar 10

1.2.5 Vector propi i valor propi 11

1.2.6 Hermític i matrius 12

1.2.7 Producte tensorial 13

2 Teoria de la computació **15**

2.1 Teoria de la computabilitat 15

2.1.1 La màquina de Turing 15

2.2 Teoria de la complexitat computacional 19

2.2.1 Quantificació de recursos computacionals. Notació asimptòtica. . . 19

2.2.2 Classes de complexitat computacional. 20

3 Preàmbul a la mecànica quàntica **23**

4 Mecànica quàntica **27**

4.1 Experiments importants en la mecànica quàntica 27

4.1.1 Experiment de Stern–Gerlach i l'espín 27

4.1.2 Experiment de la doble escletxa i la dualitat ona-partícula 29
4.2 Postulats de la mecànica quàntica 31
4.2.1 Postulat I 32
4.2.2 Postulat II 32
4.2.3 Postulat III 32
4.2.4 Postulat IV 33

5 Computació quàntica **35**
5.1 Bit quàntic 35
5.2 Portes quàntiques 38

6 Aplicació de la computació quàntica amb Qiskit **43**
6.1 Circuits quàntics 44
6.1.1 Teleportació quàntica 48
6.2 Algoritmes quàntics 52
6.2.1 Algoritme de Deutsch–Jozsa 53

7 Conclusions **57**

A Insal·lació Qiskit **59**

Bibliografia **61**

Introducció

Context i motivació

Fins fa 80 anys no existien els ordinadors, actualment hem arribat a un punt que gairebé no en poden existir de més potents. Els transistors, l'element principal dels ordinadors, els hem millorat i empetitit tant que hem arribat al seu límit. Què bé ara?

A principis del segle XX es va fundar la mecànica quàntica, la ciència que estudia les partícules subatòmiques, aquestes partícules tenen comportaments molt característics i els científics han sabut veure una forma d'aprofitar aquestes característiques per aplicar-les en computació, i han creat el següent pas en els ordinadors, els ordinadors quàntics.

Igual que amb la computació i els ordinadors clàssics, la computació i els ordinadors quàntics avancen molt ràpid. És una tecnologia molt nova però les grans empreses de tecnologia hi tenen moltes esperances i hi estan invertint molt. Hi ha molts matemàtics, físics i programadors treballant amb l'avanç d'aquesta nova tecnologia i cada dia surten noves funcionalitats i millores.

Objectius

L'objectiu principal d'aquest treball és conèixer i entendre en què consisteix la computació quàntica i quines diferències té amb la computació clàssica. La majoria d'experts que estudien i milloren aquest nou paradigma de computació són del camp de la física i/o de les matemàtiques, volem saber si realment són necessaris extensos coneixements en el camp de la física i les matemàtiques per entendre la computació quàntica i programar un ordinador quàntic.

Estructura de la Memòria

- **Fonaments matemàtics:** En el capítol de fonaments matemàtics s'expliquen i es repassen els conceptes matemàtics més bàsics que es necessiten per entendre la computació quàntica.

- **Teoria de la computació:** En aquest capítol es repassa el model computacional de la màquina de Turing i el concepte i les classes de complexitat computacional.
- **Preàmbul a la mecànica quàntica:** En aquest capítol es proporciona una forma de passar de la teoria clàssica a la quàntica mitjançant la reoria de grafs i matrius.
- **Mecànica quàntica:** En el capítol de mecànica quàntica es defineix aquesta nova ciència i s'expliquen els fenòmens més característics que necessitem per la computació quàntica
- **Computació quàntica:** En aquest capítol s'explica i es defineix en què consisteix la computacio quàntica
- **Aplicació de la computació quàntica amb Qiskit:** Finalment a l'últim capítol s'utilitza el Framework de Qiskit per progrmar ordinadors quàntics

Capítol 1

Fonaments matemàtics

Per començar el nostre camí en el coneixement de la computació quàntica, cal que tinguem frescos certs conceptes matemàtics que ens faran falta per entendre millor com funciona.

Per aquest capítol he resumit el capítol 2.1 del llibre "*Quantum Computation and Quantum Information*" [2] i els capítols 1 i 2 del llibre "*Quantum computing for computer scientists*"[3]

1.1 Nombres complexos

La computació quàntica s'aprofita dels fenòmens fonamentals de la mecànica quàntica i aquesta treballa majoritàriament amb nombres complexos, per tant ens serà necessari repassar-los.

Però, abans d'explicar un nou sistema numèric recordem els conjunts de nombres amb els que normalment treballem:

1. Nombres positius: $\mathbb{P} = \{1, 2, 3, \dots\}$
2. Nombres naturals: $\mathbb{N} = \{0, 1, 2, 3, \dots\}$
3. Nombres enters: $\mathbb{Z} = \{\dots, -3, -2, -2, 0, 1, 2, 3, \dots\}$
4. Nombres racionals: $\mathbb{Q} = \left\{ \frac{m}{n} \middle| \ m \in \mathbb{Z}, n \in \mathbb{P} \right\}$
5. Nombres reals: $\mathbb{R} = \mathbb{Q} \cup \left\{ \dots, \sqrt{2}, \dots, e, \dots, \pi, \dots \right\}$

Cap dels anteriors sistemes poden trobar cap solució vàlida per l'equació:

$$x^2 + 1 = 0. \tag{1.1.1}$$

Per això utilitzarem la unitat imaginaria i:

$$i^2 = -1. \tag{1.1.2}$$

$$\imath = \sqrt{-1}. \tag{1.1.3}$$

Els múltiples de $\imath$ per un nombre real s'anomenen **nombres imaginaris**, per exemple, $2 \times \imath$. Si, a més, sumem un nombre real al nombre imaginari obtindrem un **nombre complex,** que és una expressió tal que:

$$c = a + b \times \imath = a + b\imath, \tag{1.1.4}$$

on a, b són nombres reals. La part a s'anomena part real de c i la part b s'anomena part imaginaria de c. Aquesta forma de representar els nombres complexos s'anomena forma binòmica. Per operar amb els nombres complexos, ajuntem cada una de les seves parts i les operem per separat, és a dir, la part real amb la part real i la part imaginaria amb la part imaginaria.

Exemple d'operacions amb nombres complexos:

$$c_1 = 3 - \imath c_2 = 1 + 4\imath \tag{1.1.5}$$

$$c_1 + c_2 = (3 - \imath) + (1 + 4\imath) = (3 + 1) + (-1 + 4)\,\imath = 4 + 3\imath. \tag{1.1.6}$$

1.1.1 L'àlgebra dels nombres complexes

En primer lloc definim un nombre complex com un parell ordenat de reals:

$$c = (a, b) \tag{1.1.7}$$

Per tant qualsevol nombre real es por identificar com el parell:

$$a = (a, 0) \tag{1.1.8}$$

Per tant, els nombres imaginaris seran parells (0,b), més concretament:

$$i = (0, 1) \tag{1.1.9}$$

Les operacions de suma i multiplicació en parells ordenats funcionen de la següent forma:

$$(a_1, b_1) + (a_2, b_2) = (a_1 + a_2, b_1 + b_2) \tag{1.1.10}$$

$$(a_1, b_1) \times (a_2, b_2) = (a_1, b_1)\,(a_2, b_2) = (a_1 a_2 - b_1 b_2, a_1 b_2 + a_2 b_1)$$

Si utilitzem la suma i la multiplicació podem escriure qualsevol nombre complex en la seva forma binòmica:

$$c = (a, b) = (a, 0) + (0, b) = (a, 0) + (b, 0) \times (0, 1) = a + b\imath \tag{1.1.11}$$

Podem veure, per tant, que les dues operacions, la multiplicació i la suma són **commutatives**,

(1.1.11)

$$c_1 + c_2 = c_2 + c_1$$

$$c_1 \times c_2 = c_2 \times c_1$$

i **associatives**:

(1.1.12)

$$(c_1 + c_2) + c_3 = c_1 + (c_2 + c_3)$$

$$(c_1 \times c_2) \times c_3 = c_1 \times (c_2 \times c_3)$$

A més la multiplicació és **distributiva** sobre la suma,

$$c_1 \times (c_2 + c_3) = (c_1 \times c_2) + (c_1 \times c_3) \tag{1.1.12}$$

Els nombres reals tenen una operació unitària, el valor absolut, donat per:

$$|a| = +\sqrt{a^2} \tag{1.1.13}$$

Podem definir una generalització de la operació 1.1.13 pel domini complex:

$$|c| = |a + bı| = +\sqrt{a^2 + b^2}, \tag{1.1.14}$$

aquesta qualitat s'anomena **mòdul** d'un nombre complex. Tal com podem deduir de l'operació 1.1.14:

$$|c|^2 = a^2 + b^2 \tag{1.1.15}$$

El parell (0,0) és la identitat de la suma, és a dir, si el sumem a qualsevol nombre complex ens retornarà com a resultat el nombre complex i el parell (1,0) és la identitat de la multiplicació, que multiplicat per qualsevol nombre complex tindrà com a resultat el nombre complex.

$\mathbb{C}$ és un cos, és un sistema algebraic en el què és possible efectuar la suma, resta, multiplicació i divisió (llevat de la divisió per 0), i en el qual se satisfan certes lleis. Un cos que té solucions per qualsevol de les seves equacions polinòmiques s'anomena algebraicament complert, així doncs, $\mathbb{C}$ és un cos algebraicament complert.

L'operació de canvi de signe te un rol crucial en el domini dels nombres complexos, tenim dos nombres reals units a un nombre complex, per tant tenim tres formes de canviar de signe: canviar la part real, la part imaginària o les dues.

Per canviar el signe a ambdues parts hem de multiplicar el nombre complex per el nombre -1 = (-1,0). Per altre banda, canviar el signe tan sols a la part imaginària és conegut com a **conjugació**, ja que si tenim c = a + bi el seu conjugat és $\bar{c} = c^* = a - bi$.

Propietats de la conjugació:

$$\overline{c_1} + \overline{c_2} = \overline{c_1 + c_2} \tag{1.1.16}$$

$$\overline{c_1} \times \overline{c_2} = \overline{c_1 \times c_2} \tag{1.1.17}$$

$$c \times \overline{c} = |c|^2 \tag{1.1.18}$$

Per referir-nos als conjugats d'un nombre complex utilitzarem les notacions $\overline{c}$ i c^* indistintament.

1.1.2 La geometria dels nombres complexes

Anteriorment hem après que un nombre complex és un parell de nombres reals, aquesta parella de reals es pot representar com a punts del pla o, de forma equivalent, com a vectors que comencen des de l'origen i que apunten cap a aquest punt. El pla real que coneixem està format per l'eix x i l'eix y, per representar els nombres complexes utilitzarem el pla complex o d'Argand, on la part real la representarem a l'eix x, eix real, i la part imaginaria a l'eix y, eix imaginari. Per tant el mòdul definit anteriorment no deixa de ser la longitud del vector.

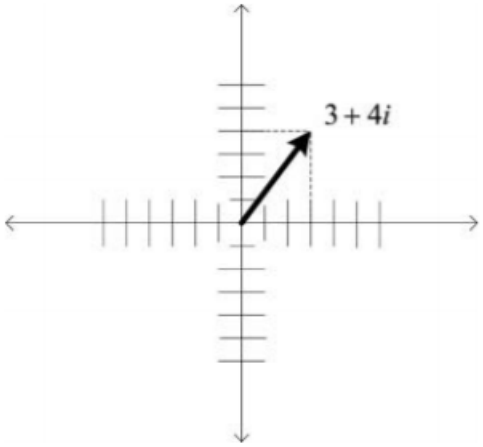


Figura 1.1: Representació del vector 3+4$\imath$ en el pla complex

Els vectors de nombres complexes es poden sumar utilitzant la regla del paral·lelogram, que és una regla pràctica per a sumar gràficament vectors, segons la qual la suma a + b de dos vectors a i b, no paral·lels i amb l'origen comú, és el vector que té el mateix origen comú i que és determinat per la diagonal corresponent del paral·lelogram que els vectors a i b determinen. La resta és el mateix que la suma negant el vector que es resta, i el vector negat d'un vector, és aquell amb la mateixa longitud i direcció oposada.

Si tenim el mòdul d'un nombre complex i volem dibuixar el triangle que forma necessitem l'angle amb l'origen, per tant per determinar un nombre complex de forma única necessitem el mòdul i l'angle. Tenint en compte l'anterior afirmació:

$$(a,b) \rightarrow (\rho,\theta) \tag{1.1.19}$$

$$\rho = \sqrt{(a^2+b^2)} \tag{1.1.20}$$

$$\theta = \tan^{-1}\left(\frac{a}{b}\right) \tag{1.1.21}$$

La representació del nombre complex com (ρ,θ), s'anomena notació **polar.** Podem passar de notació cartesiana a polar de la següent forma:

$$a = \rho\cos(\theta) \tag{1.1.22}$$

$$b = \rho\sin(\theta) \tag{1.1.23}$$

Existeix una quarta forma de representar els nombres complexes, es la **notació exponencial,** es basa en la formula d'Euler,

$$\mathrm{e}^{\mathrm{i}\theta} = \cos(\grave{}) + i\sin(\theta). \tag{1.1.24}$$

i és de la següent forma:

$$\mathrm{c} = \mathrm{e}^{\mathrm{i}\grave{}} \tag{1.1.25}$$

I per últim la multiplicació, per poder multiplicar dos nombres complexes en notació polar cal multiplicar les seves magnituds i sumar les seves fases:

$$(\rho_1,\theta_1)\times(\rho_2,\theta_2) = (\rho_1\rho_2,\theta_1+\theta_2) \tag{1.1.26}$$

Implícitament hem après un fet important: la multiplicació en el domini complex té alguna cosa a veure amb les rotacions del pla complex.

1.2 Àlgebra lineal

L'àlgebra lineal és l'estudi d'espais vectorials i d'operacions lineals en aquests espais. Una bona comprensió de la mecànica quàntica es basa en una comprensió sòlida de l'àlgebra lineal elemental.

Els elements que formen un espai vectorial son els vectors, l'espai vectorial que més ens interessa és $\mathbb{C}^n$, que és l'espai de totes les n-pla (seqüències o llistes ordenades de n objectes) de nombres complexos, $(z_1,\ldots,z_n)$.

Per tal de de veure amb claredat quin tipus d'estructura té aquest conjunt, examinem detingudament un exemple concret: el conjunt de vectors de longitud 4. Denotarem aquest conjunt com a $\mathbb{C}^4$=$\mathbb{C}\times\mathbb{C}\times\mathbb{C}\times\mathbb{C}$, on cada vector és una n-pla de quatre nombres complexes. Dos elements de $\mathbb{C}^4$ podrien ser:

$$V = \begin{bmatrix} 6-4\imath \\ 7+3\imath \\ 4.2-8.1\imath \\ -3\imath \end{bmatrix} \quad W = \begin{bmatrix} 16-2.3\imath \\ -7\imath \\ 6 \\ -4\imath \end{bmatrix} \tag{1.2.1}$$

Cada un dels nombres complexes del vector és un **element** o una **entrada** del vector, que la podem denominar com, prenent com exemple el vector V, V[j], per exemple, V[0] = $6-4\imath$. Podem sumar els dos vectors de l'equació 1.2 de forma que V+W ϵ $\mathbb{C}^4$, sumant els seus elements respectivament:

$$\begin{bmatrix} 6-4\imath \\ 7+3\imath \\ 4.2-8.1\imath \\ -3\imath \end{bmatrix} + \begin{bmatrix} 16-2.3\imath \\ -7\imath \\ 6 \\ -4\imath \end{bmatrix} = \begin{bmatrix} (6-4\imath)+(16-2.3\imath) \\ (7+3\imath)+(-7\imath) \\ (4.2-8.1\imath)+(6) \\ (-3\imath)+(-4\imath) \end{bmatrix} = \begin{bmatrix} 22-1.7\imath \\ 7-4\imath \\ 10.2-8.1\imath \\ -7\imath \end{bmatrix} \tag{1.2.2}$$

Aquesta operació equival a (V + W) [j] = V[j] + W[j]. La suma és **commutativa**, V + W = W + V, i **associativa**, si afegim un vector X, (V + W) + X = V + (W + X). També contem amb un vector distingit, el **vector zero**,

$$0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{1.2.3}$$

que satisfà la següent propietat: per tots els vectors V ϵ $\mathbb{C}^4$ es compleix que, V + 0 = V = 0 + V. Tot vector té el seu **invers o negatiu,** agafem el vector V indicat a l'equació 1.2 :

$$-V = \begin{bmatrix} -6+4\imath \\ -7-3\imath \\ -4.2+8.1\imath \\ 3\imath \end{bmatrix} \tag{1.2.4}$$

A més -V ϵ $\mathbb{C}^4$. Per tot vector pertanyent a $\mathbb{C}^4$ aquest té un invers que complex V+(−V)=(−V)+V = 0.

El conjunt $\mathbb{C}^4$ amb la suma, les operacions inverses, i el zero, formen una grup Abelià. Un grup és una estructura formada per un conjunt, X, sobre el qual s'ha definit una operació, *, pels seus elements que té la propietat associativa, té un element neutre i un invers. Si l'operació a més té la propietat commutativa serà un grup abelià.

En el conjunt $\mathbb{C}^4$ també tenim l'operació de multiplicació per un escalar, que tal com diu el nom és la multiplicació d'un nombre complex per un vector complex, per realitzar-la multiplicarem l'escalar per cada entrada del vector:

$$(3+2\imath)\cdot\begin{bmatrix}6+3\imath\\0+0\imath\\5+1i\\4\end{bmatrix}=\begin{bmatrix}12+21\imath\\0+0i\\13+13\imath\\12+8\imath\end{bmatrix} \tag{1.2.5}$$

La multiplicació escalar satisfà les següents propietats per tots els escalars $c_1, c_2, c_3 \in \mathbb{C}$ i per tots els vectors $V, W \in \mathbb{C}^4$,

1. $1 \cdot V = V$
2. $c_1 \cdot (c_2 \cdot V) = (c_1 \times c_2) \cdot V$
3. $c \cdot (V + W) = c \cdot V + c \cdot W$
4. $(c_1 + c_2) \cdot V = c_1 \cdot V + c_2 \cdot V$

Un grup abelià amb multiplicació escalar que satisfaci les anteriors propietats s'anomena **espai vectorial complex**.

Tot el que hem vist per l'espai $\mathbb{C}^4$ es compleix pels espais $\mathbb{C}^n$.

En mecànica quàntica, els vectors en un espai vectorial s'expressen de la següent forma:

$$|\,\varphi\rangle \tag{1.2.6}$$

on φ és el nom del vector i $|\,\cdot\rangle$, que s'anomena notació de Dirac o bra-ket, ens indica que l'objecte és un vector. A l'equació1.2.6 veiem el que s'anomena *ket*, però també tenim el *bra* $\langle\cdot\,|$, veurem el seu significat a l'apartat 1.2.4 producte escalar. A partir d'ara utilitzarem aquesta notació per referir-nos als vectors.

1.2.1 Bases dels espais vectorials

Un vector $|\,v\rangle$ es diu que és **combinació lineal** d'un conjunt de vectors A = $|\,v_1\rangle$, ... , $|\,v_n\rangle$, si existeix una manera d'expressar-lo com a suma de part o tots els vectors de A multiplicats cadascun per un coeficient escalar $a_1, a_2, a_3, \ldots, a_n$ de manera que:

$$|\,v\rangle \;= \sum_{i=1}^{n} a_i |\,v_i\rangle \tag{1.2.7}$$

Un **conjunt generador** per a un espai vectorial és un conjunt de vectors $|\,v_1\rangle$, ... , $|\,v_n\rangle$, de manera que qualsevol vector $|\,v\rangle$ de l'espai vectorial es pot escriure com una combinació lineal de vectors d'aquest conjunt. Per exemple, un conjunt generador per l'espai vectorial $\mathbb{C}^2$ seria el conjunt:

$$|\,v_1\rangle \equiv \begin{bmatrix}1\\0\end{bmatrix}; \qquad |\,v_2\rangle\,, \equiv \begin{bmatrix}0\\1\end{bmatrix} \tag{1.2.8}$$

ja que qualsevol vector $|\, v\rangle$ de $\mathbb{C}^2$ es por expressar com $|\, v\rangle = a_1|\, v_1\rangle + a_2|\, v_2\rangle$. Direm doncs que $|\, v_1\rangle$ i $|\, v_2\rangle$ **generen** l'espai vectorial $\mathbb{C}^2$. Un espai vectorial, normalment, té diferents conjunts generadors.

Direm que un conjunt de vectors $A = |\, v_1\rangle, \ldots, |\, v_n\rangle$ és **linearment dependent** si existeix un conjunt de nombres complexes $a_1, a_2, a_3, \ldots, a_n$, on com a mínim un dels valors del conjunt sigui diferent a 0, que faci que el resultat de la seva combinació lineal sigui zero:

$$a_1\,|v_1\rangle + a_2|\, v_2\rangle + \cdots + a_n|\, v_n\rangle = 0 \tag{1.2.9}$$

Si no existeix cap conjunt de nombres que compleixi aquesta condició, voldrà dir que A és **linearment independent**, és a dir, que cap dels seus vectors es pot escriure com a combinació lineal dels altres.

Tenint en compte tot el que hem après anteriorment, un conjunt de vectors A serà **base** d'un espai vectorial, si A és linealment independent i genera de forma única l'espai vectorial, és a dir, que tots els vectors d'aquest espai vectorial es poden escriure com a combinació lineal del conjunt A.

La **dimensió** d'un espai vectorial serà el nombre de vectors de la base de l'espai vectorial.

1.2.2 Matrius

Repassem les operacions amb matrius ja que son una part que utilitzarem al llarg del document. La nostre matriu:

$$M = \begin{bmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,n-1} \\ c_{1,0} & c_{1,1} & \cdots & c_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m-1,0} & c_{m-1,1} & \cdots & c_{m-1,n-1} \end{bmatrix} \tag{1.2.10}$$

1. Matriu multiplicada **per un escalar**, per obtenir-la hem de multiplicar cada element de la matriu per el nombre:

$$M{\cdot}c = \begin{bmatrix} c \times c_{0,0} & c \times c_{0,1} & \cdots & c \times c_{0,n-1} \\ c \times c_{1,0} & c \times c_{1,1} & \cdots & c \times c_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ c \times c_{m-1,0} & c \times c_{m-1,1} & \cdots & c \times c_{m-1,n-1} \end{bmatrix} \tag{1.2.11}$$

2. Matriu **inversa**, per obtenir-la hem d'invertir cada element de la matriu:

$$-M = \begin{bmatrix} -c_{0,0} & -c_{0,1} & \cdots & -c_{0,n-1} \\ -c_{1,0} & -c_{1,1} & \cdots & -c_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ -c_{m-1,0} & -c_{m-1,1} & \cdots & -c_{m-1,n-1} \end{bmatrix} \tag{1.2.12}$$

3. Matriu **transposada**, per obtenir-la hem de permutar les files i les columnes de la matriu:

$$M^T = \begin{bmatrix} c_{0,0} & c_{1,0} & \dots & c_{m-1,0} \\ c_{0,1} & c_{1,1} & \dots & c_{m-1,1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{0,n-1} & c_{1,n-1} & \dots & c_{m-1,n-1} \end{bmatrix} \tag{1.2.13}$$

4. Matriu **conjugada**, per obtenir-la hem d'obtenir el conjugat de cada element de la matriu:

$$\overline{M} = \begin{bmatrix} \overline{c_{0,0}} & \overline{c_{0,1}} & \dots & \overline{c_{0,n-1}} \\ \overline{c_{1,0}} & \overline{c_{1,1}} & \dots & \overline{c_{1,n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{c_{m-1,0}} & \overline{c_{m-1,1}} & \dots & \overline{c_{m-1,n-1}} \end{bmatrix} \tag{1.2.14}$$

5. Matriu **conjugada transposada**, per obtenir-la combinarem la transposada i la conjugada:

$$\left(\overline{M}\right)^T = \overline{M^T} = M^\dagger = \begin{bmatrix} \overline{c_{0,0}} & \overline{c_{1,0}} & \dots & \overline{c_{m-1,0}} \\ \overline{c_{0,1}} & \overline{c_{1,1}} & \dots & \overline{c_{m-1,1}} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{c_{0,n-1}} & \overline{c_{1,n-1}} & \dots & \overline{c_{m-1,n-1}} \end{bmatrix} \tag{1.2.15}$$

6. Multiplicació d'una **matriu per un vector**, multipliquem cada element de cada fila pel vector, respectivament:

$$(1.2.16\ ) \quad \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{bmatrix} w \\ y \end{bmatrix} = \begin{bmatrix} aw + by \\ cw + dy \end{bmatrix}$$

7. Multiplicació de **matrius per matrius**, multipliquem cada element d'una fila de la primera matriu per cada element d'una columna de la segona matriu respectivament:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} w & x \\ y & z \end{pmatrix} = \begin{pmatrix} aw + by & ax + bz \\ cw + dy & cx + dz \end{pmatrix} \tag{1.2.16}$$

8. **Determinant** d'una matriu quadrada, és el número associat a una matriu i el càlcul depèn del ordre de la matriu:

Matriu d'ordre 2, 2x2:

$$B = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \tag{1.2.17}$$

$$\det(B) = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc \tag{1.2.18}$$

Matriu d'ordre 3, 3x3, Regla de Sarrus:

$$C = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

(1.2.19)

$$det\,(C) = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}a_{22}a_{33} + a_{21}a_{32}a_{13} + a_{12}a_{23}a_{31} - (a_{31}a_{22}a_{13} + a_{32}a_{23}a_{11} + a_{21}a_{12}a_{33})$$

(1.2.20)

Les matrius quadrades $\mathbb{C}^{nxn}$ són elements importants per la computació quàntica. Els elements $\mathbb{C}^n$ són la forma de descriure els estats quàntics i alguns elements $\mathbb{C}^{nxn}$ corresponen als canvis d'estat d'un sistema quàntic. Tenint un estat X pertanyent a $\mathbb{C}^n$ i una matriu A pertanyent a $\mathbb{C}^{nxn}$ podem formar un altre estat del sistema A·X $\in \mathbb{C}^n$. Formalment ho escriurem com $\mathbb{C}^{nxn} \times \mathbb{C}^n \to \mathbb{C}^n$.

1.2.3 Aplicacions lineals

Una aplicació entre dos conjunts A i B és una regla que permet assignar a cada element d'A, un de B. Una aplicació lineal entre els espais vectorials V i W es defineix per ser qualsevol funció A: V $\to$ W que és lineal en les seves entrades:

$$A\left(\sum_i a_i |\, v_i\rangle\right) = \sum_i a_i A(|\, v_i\rangle) \tag{1.2.21}$$

Les aplicacions lineals més importants de qualsevol espai vectorial són la **identitat** I_v, $I_v|\, v\rangle = |\, v\rangle$ per qualsevol vector $|\, v\rangle$ i la **zero**, que anotem com a 0 i transforma tots els vectors al vector zero $0|\, v\rangle = 0$.

Una forma més senzilla d'entendre una aplicació lineal és per la seva **representació matricial**. Una matriu A $\in \mathbb{C}^{mxn}$ és en definitiva una aplicació lineal que transforma els vectors de l'espai vectorial $\mathbb{C}^n$ a $\mathbb{C}^m$ a partir de la multiplicació d'A per un vector de $\mathbb{C}^n$.

1.2.4 Producte escalar

El producte escalar és una funció que pren dos vectors $|\, v\rangle$ i $|\, w\rangle$ d'un espai vectorial i produeix un resultat en forma de número complex. Aquesta funció compleix les següents condicions:

1. És linear pel segon argument:

$$\left(\mid v\rangle \, , \sum_i \lambda_i \mid w_i\rangle \right) = \sum_i \lambda_i \left(\mid v\rangle \, , \mid w_i\rangle \right)$$

2. $(\mid v\rangle \, , \mid w\rangle) = (\overline{\mid w\rangle \, , \mid v\rangle})$

3. $(\mid v\rangle \, , \mid v\rangle) \geq 0$, serà 0 si $\mid v\rangle = 0$

Un espai vectorial amb l'operació de producte escalar s'anomena espai **prehilbertià**, que en els *espais vectorials finits*, que son amb els que treballem, serà el mateix que **l'espai de Hilbert**. En mecànica quàntica anotarem el producte escalar amb la notació bra-ket $\langle \cdot \mid \cdot \rangle$, si tenim els vectors $\mid v\rangle$ i $\mid w\rangle$ el seu producte escalar serà $\langle v \mid w\rangle$ on $\langle v\mid$, el bra, s'utilitza per anotar el **vector dual** de $\mid v\rangle$; el dual és una aplicació linear des de l'espai prehilbertià als números complexos, definida pel producte escalar, més endavant en aquest capítol el definirem de forma més concreta.

Dos vectors seran **ortogonals,** perpendiculars, si el seu producte escalar és zero. Definirem la **norma** d'un vector $\mid v\rangle$ com:

$$\| \mid v\rangle \; \| \equiv \sqrt{\langle v \mid v\rangle} \tag{1.2.22}$$

Si $\| \mid v\rangle \; \| = 1$ voldrà dir que el vector $\mid v\rangle$ és un vector **unitari** o que està **normalitzat**, per normalitzar un vector hem de dividir el vector per la seva norma. Un conjunt de vectors V=$\{v_0,v_1,...,v_{n-1}\}$ serà **ortonormal** si els vectors diferents del conjunt són ortogonals, és a dir, $\langle v_i \mid v_j \rangle = 0$ per i $\neq$ j i tots els vectors del conjunt són unitaris.

1.2.5 Vector propi i valor propi

Si per una matriu $\mathbb{C}^{nxn}$ A existeix un nombre complex c i un vector complex V $\neq 0$ que compleixi:

$$AV = c \cdot V \tag{1.2.23}$$

llavors c serà el **valor propi** d'A i V serà el **vector propi** d'A associat amb c.

Per exemple:

$$\begin{pmatrix} 4 & -1 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \end{pmatrix} = 3 \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{1.2.24}$$

Per trobar els valors propis d'una matriu s'utilitza **l'equació característica**,

$$det|A - \lambda I| = 0 \tag{1.2.25}$$

on λ és una variable desconeguda, I és la matriu identitat i det és el determinant de la matriu $A - \lambda I$.

Per tant, si A és una aplicació lineal, els vectors propis d'aquesta aplicació són aquells que, o no els afecta la transformació o el resultat és el mateix vector multiplicat per un escalar, si resulta ser la segona opció l'escalar que multipliqui aquest vector serà el valor propi de l'aplicació lineal. Si c_0 és un valor propi d'A i V_0 un vector propi corresponent a c_0, llavors tenim que per qualsevol $c \in \mathbb{C}$:

$$A\left(cV_0\right) = cAV_0 = cc_0V_0 = c_0\left(cV_0\right), \tag{1.2.26}$$

podem veure doncs, que cV_0 també és un vector propi d'A pel valor propi c_0. Tenint en compte això, un **espai propi** serà l'espai format per tots els vectors propis del mateix valor propi, i serà un subespai vectorial del espai on actua A.

1.2.6 Hermític i matrius

Una aplicació lineal A en un espai Hilbert V, té una aplicació lineal única $A^\dagger$ a V que per tots els vectors $|\, v\rangle, |\, w\rangle \in V$ compleix:

$$\left(|\, v\rangle, A|\, w\rangle\right) = \left(A^\dagger |\, v\rangle, |\, w\rangle\right) \tag{1.2.27}$$

Aquesta aplicació lineal es coneix com **l'adjunt** d'A. Si prenem la forma matricial de l'aplicació lineal A, el que fem és transformar A en la seva matriu conjugada transposada $A^\dagger$, si resulta que $A = A^\dagger$ llavors la matriu és **hermítica** i l'aplicació que representa s'anomena **operador autoadjunt**.

Si una matriu és hermítica els seus valors propis seran reals $\mathbb{R}$ i els vectors propis dels valors propis seran ortogonals.

Una matriu U serà **unitària** si:

$$U^\dagger U = UU^\dagger = I \tag{1.2.28}$$

La matriu unitària U preserva el producte escalar i la norma, per qualsevol $V, V' \in \mathbb{C}$:

$$\langle UV \mid UV' \rangle = \langle V \mid V' \rangle$$

$$\|| UV\rangle \| = \sqrt{\langle UV \mid UV\rangle} = \sqrt{\langle V \mid V\rangle} = \|| V\rangle \| \tag{1.2.29}$$

Si tenim un conjunt de vectors de longitud u que forma una bola al voltant de l'origen tindrem una esfera unitària, si V forma part d'aquest conjunt UV també, d'alguna manera el que farà U serà rotar l'esfera unitària.

Si U és una matriu unitària i $UV = V'$, podem formar fàcilment $U^\dagger$, si multipliquem $U^\dagger$ per les dos bandes de l'equació obtenim $U^\dagger UV = U^\dagger V'$ que és el mateix que $IV = V = U^\dagger V'$, la importància d'això és que l'acció que realitza $U^\dagger$ és anular l'acció que ha realitzat U sobre V i retornà a V, això ho fa reversible i aquesta acció serà important més endavant.

1.2.7 Producte tensorial

El producte tensorial és un mètode per combinar espais vectorials. Si tenim dos espais vectorials $\mathbb{V}$ i $\mathbb{W}$ denotarem el producte tensorial entre ells com $\mathbb{V} \otimes \mathbb{W}$ i el generarem amb un conjunt de combinacions lineals de productes tensorials de tots els seus vectors:

$$c_0 \left(\mid v_1\rangle \otimes \mid w_1\rangle \right) + \; c_2 \left(\mid v_2\rangle \otimes \mid w_2\rangle \right) + \cdots + c_{p-1} \left(\mid v_{p-1}\rangle \otimes \mid w_{p-1}\rangle \right) \tag{1.2.30}$$

on $\mid v\rangle$ són elements de $\mathbb{V}$ i $\mid w\rangle$ elements de $\mathbb{W}$. En particular, si $\mid i\rangle$ i $\mid j\rangle$ són bases ortonormals per els espais $\mathbb{V}$ i $\mathbb{W}$ llavors $\mid i\rangle \otimes \mid j\rangle$ serà una base per $\mathbb{V} \otimes \mathbb{W}$. També podem anotar el producte tensorial entre dos vectors $\mid v\rangle \otimes \mid w\rangle$ com: $|v\rangle| w\rangle$, $\mid v, w\rangle$ o $\mid vw\rangle$. El producte tensorial satisfà les següents propietats:

1. Per un escalar qualsevol z i elements $\mid v\rangle$ de $\mathbb{V}$ i $\mid w\rangle$ de $\mathbb{W}$:

$$z(\mid v\rangle \otimes |w\rangle) = (z| \, v\rangle) \otimes \mid w\rangle = \; \mid v\rangle \otimes (z| \, w\rangle) \tag{1.2.31}$$

1. Per qualsevol $\mid v_1\rangle$ i $\mid v_2\rangle$ de $\mathbb{V}$ i $\mid w\rangle$ de $\mathbb{W}$:

$$(\mid v_1\rangle + \; |v_2\rangle) \otimes \; \mid w\rangle = \; \mid v_1\rangle \otimes \; \mid w\rangle + \mid v_2\rangle \otimes \; \mid w\rangle \tag{1.2.32}$$

1. Si tenim $\mid v\rangle$ com a vector de$\mathbb{V}$, $\mid w\rangle$ com a vector de $\mathbb{W}$ i A i B com a aplicacions lineals de $\mathbb{V}$ i $\mathbb{W}$ respectivament, podem definir $A \otimes$ B a $\mathbb{V} \otimes \mathbb{W}$ com:

$$(A \otimes \mathrm{B}) \; (\mid v\rangle \otimes |w\rangle) \; \equiv A| \, v\rangle \otimes \; B| \, w\rangle \tag{1.2.33}$$

El càlcul del producte tensorial de dos vector serà doncs:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \otimes \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} a_0 \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} \\ a_1 \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} \\ a_2 \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} \\ a_3 \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} a_0 b_0 \\ a_0 b_1 \\ a_0 b_2 \\ a_1 b_0 \\ a_1 b_1 \\ a_1 b_2 \\ a_2 b_0 \\ a_2 b_1 \\ a_2 b_2 \\ a_3 b_0 \\ a_3 b_1 \\ a_3 b_2 \end{bmatrix} \tag{1.2.34}$$

Per tant, la dimensió de $\mathbb{V} \otimes \mathbb{W}$ serà la dimensió de $\mathbb{V}$ tantes vegades com la dimensió de $\mathbb{W}$.

El producte tensorial de dos matrius A i B serà:

(1.2.38)

$$A \otimes \mathrm{B} \equiv \begin{bmatrix} A_{11}B & A_{12}B & \dots & A_{1n}B \\ A_{21}B & A_{22}B & \dots & A_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1}B & A_{m2}B & \dots & A_{mn}B \end{bmatrix}$$

Una notació molt útil en productes tensorials és $\mid v\rangle^{\otimes k}$ que simbolitza el producte tensorial de $\mid v\rangle$ amb ell mateix k vegades.

Capítol 2

Teoria de la computació

La teoria de la computació és una branca de les ciències de la computació que busca determinar què pot ser i què no pot ser computat, amb quina rapidesa, amb quanta memòria i amb quin tipus de model de computació.

En aquest capítol tractarem dos de les tres branques de la teoria de la computació, la teoria de la computabilitat i la teoria de la complexitat computacional, ja que ens serà necessari tenir-les fresques per seguir endavant amb l'estudi de la computació quàntica. Tot i així com que son temes molt extensos tan sols tocarem el que necessitem per tirar endavant.

Algoritme: *procediment de càlcul que consisteix a acomplir un seguit ordenat i limitat d'instruccions que condueix, un cop especificades les dades, a la solució que el problema genèric en qüestió té per a les dades considerades* .

Per aquest capítol he resumit el capítol 3 del llibre "*Quantum Computation and Quantum Information*" [2] i els capítols 1 i 8 del llibre "*Algorísmia comentada*"[1]

2.1 Teoria de la computabilitat

La teoria de la computabilitat estudia quins problemes són computables i quins no. Els problemes computables, de forma bàsica, són aquells pels que existeix com a mínim un algoritme que pot resoldre'ls. El model de computació més utilitzat per determinar si un problema és computable o no és la màquina de Turing.

2.1.1 La màquina de Turing

És un model computacional introduït per Alan Turing, en el treball "On computable numbers, with an application to the Entscheidungsproblem", que estudiava el repte que va plantejar el matemàtic David Hilbert, en el que formulava la pregunta de si les matemàtiques eren decidibles, és a dir, si existia algun mètode que es pogués aplicar a qualsevol sentència matemàtica i indiques si és certa o falsa. En aquest treball Turing

va demostrar que existien problemes que la màquina no podia resoldre, això significava que les matemàtiques no eren decidibles, però a més, al llarg d'aquest estudi va establir les bases per a la teoria moderna dels algorismes i, per tant, per a la teoria moderna de la informàtica.

La màquina de Turing la formen quatre elements:

1. Una **cinta** que es pot estendre de forma infinita en una direcció, que actua com una memòria d'ordinador, dividida per cel·les contigües numerades. Cada cel·la contindrà un símbol d'un alfabet finit, Γ.

2. Un **capçal**, que apunta a una cel·la de la cinta i és capaç de llegir i escriure el símbol de la cel·la i moure's cel·la a cel·la per la cinta.

3. Un **registre d'estat** que guarda l'estat intern de la maquina, que serà un estat del conjunt d'estats finits possibles de la màquina $q_1, \ldots, q_m$. A part dels estats del conjunt d'estats, tenim dos estats destacats, l'estat q_s que és l'estat inicial i l'estat q_h que és l'estat d'aturada, l'estat final.

4. Un **programa** o taula finita d'instruccions, que és un conjunt de funcions de transició expressades com a tuples, que segons l'estat intern actual i el símbol de la cel·la de la cinta indicaran que ha de fer la màquina.

Formalment una màquina de Turing la podem expressar de forma simplificada com:

$$(\Sigma, \Gamma, Q, \delta) \tag{2.1.1}$$

on:

$\Sigma =$ input $Q =$ conjunt d'estats $\{q_1, \ldots, q_m\}$ $\Gamma =$ alfabet $\{s_1, \ldots, s_m\}$

$\delta =$ funció de transició = $Q \times \ \Gamma \rightarrow Q \ \times \Gamma \ \times \{L, R\}$ on L és un moviment a

l'esquerra i R és el moviment a la dreta. També la podem expressar com a tupla de la següent forma:

$$\left(q_i, s_i, q_j, s_j, d\right)$$

És a dir, si tenim com a estat intern l'estat q_i i la cel·la a la que ens trobem conté el símbol s_i, canviarem l'estat intern a q_j, el símbol pel s_j i ens mourem d que serà dreta o esquerre.

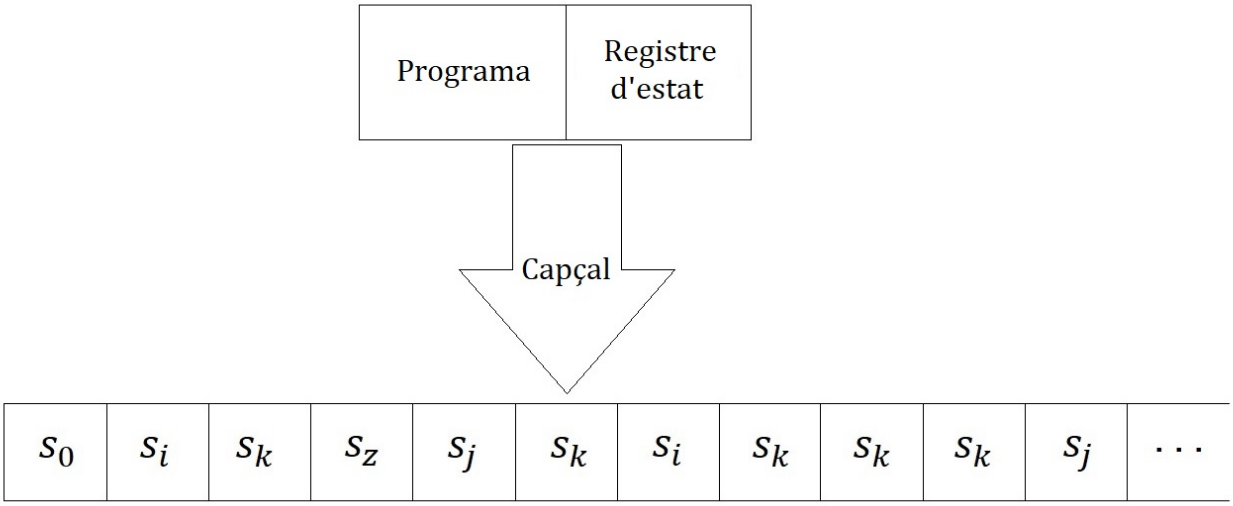


Figura 2.1: Representació gràfica de la màquina de Turing

Per exemple, si tenim una màquina de Turing M amb: $\Gamma = 0, 1, b\,(blanc)$, $\triangleright(per\ marcar\ l'^{inici\ de\ la\ cinta})$, $Q = \{\ q_1,\ q_2,\ q_3, q_s, q_h\}$, -1 = esquerre, 1 = dreta, 0 no moure's, $\Sigma = 0101b$ i tenim el següent programa:

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

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

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

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

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

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

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

Al iniciar al programa ens trobem a l'estat q_s que és l'inicial i el capçal es troba a la cel·la 0, a la taula 3.1.1 es mostra l'execució d'aquest programa per la entrada Σ,

0	1	2	3	4	5	Estat intern	Línia del programa
▷	0	1	0	1	b	q_s	1
▷	0	1	0	1	b	q_1	2
▷	b	1	0	1	b	q_1	3
▷	b	b	0	1	b	q_1	2
▷	b	b	b	1	b	q_1	3
▷	b	b	b	b	b	q_1	4
▷	b	b	b	b	b	q_2	5
▷	b	b	b	b	b	q_2	5
▷	b	b	b	b	b	q_2	5
▷	b	b	b	b	b	q_2	5
▷	b	b	b	b	b	q_2	6
▷	b	b	b	b	b	q_3	7
▷	1	b	b	b	b	q_h	-

Figura 2.2: *Exemple d'execució de la maquina de Turing M amb l'estrada Σ. Les cel·les blaves ens indiquen on es troba el capçal i l'ultima columna ens mostra la línia del programa que s'executarà per aquell estat de la maquina.*

Com podem veure a figura 2.2 el que fa aquesta màquina M és f(x) = 1. Aquesta operació és senzilla, però amb la màquina de Turing podem simular totes les operacions que realitza un ordinador modern, de fet els ordinadors quàntics poden computar la mateixa classe de funcions que son computables per la màquina de Turing, l'única diferència és l'eficiència en la computació.

"La classe de funcions computables per una màquina de Turing correspon exactament a la classe de funcions que naturalment consideraríem computables per un algorisme." – La tesis de Church-Turing –

Hi ha moltes variants de la màquina de Turing bàsica, podem tenir maquines de Turing amb diferents cintes, com per exemple, amb dos cintes o amb cintes de més d'una dimensió, però totes les diferents versions de les màquina de Turing podran computar les mateixa classe de funcions. Podem simular una maquina de Turing de dos cintes en una màquina de Turing d'una sola cinta, de fet existeix una màquina de Turing que pot simular totes les màquines de Turing, que és la màquina de Turing **universal**, aquesta màquina rep com a entrada la descripció de la màquina a simular i el contingut a computar, però no canvia la classe de funcions computables, ni tampoc canviaria si introduïm aleatorietat.

Si tornem a l'inici, amb el repte que va portar a la creació d'aquest treball, Turing havia demostrat que hi havia problemes de decisió, problemes que tenen com a resposta si-no, que la maquina de Turnig no podia resoldre, això volia dir que hi havia problemes indecidibles, problemes de decisió que eren molt difícils de resoldre, tant que no es podien resoldre amb un algoritme.

2.2 Teoria de la complexitat computacional

La teoria de la complexitat computacional investiga el temps, la memòria i altres recursos requerits per solucionar problemes computacionals.

2.2.1 Quantificació de recursos computacionals. Notació asimptòtica.

Per comprendre un problema computacional, volem una forma de quantificar els recursos a utilitzar independent dels canvis relativament trivials del model computacional. Una de les eines que s'ha desenvolupat per fer-ho és la notació asimptòtica, que s'utilitza per resumir el comportament essencial d'una funció.

Però abans d'entrar en la notació asimptòtica, hem de saber quins son els recursos computacionals. Com hem explicat anteriorment un problema computacional té, com a mínim, un algorisme associat, per tant el que farem serà estudiar el seu algorisme, i més concretament els recursos que utilitza. Normalment el recursos més estudiats són l'espai i el temps.

Per poder mesurar el temps que triga un algorisme a donar una solució a partir d'unes dades, ens cal de[U+FB01]nir primer: la mida de les dades, i el temps d'un algorisme per una entrada de mida n (dades = dades d'entrada = entrada). La mida d'una entrada serà un nombre enter positiu, és a dir, una entrada concreta x , tindrà una mida n. Per tant, agruparem les possibles entrades d'un algoritme per la seva mida. De[U+FB01]nim el temps d'un algorisme per una entrada com si només depengués de la mida d'aquesta entrada, per tant, el temps que triga un algorisme per una entrada x el podem descriure com el màxim de totes les entrades amb aquella mateixa mida (el temps de la resposta que més triga). Descrivim el *cas pitjor* d'un algorisme com el cas en el que l'entrada es triga més a respondre d'entre totes les entrades d'una mateixa mida. Sempre que parlem del cas pitjor, parlarem dels valors de l'entrada. El cas pitjor es produeix quan els valors de l'entrada compleixen certes condicions, no podem recórrer a la mida de les dades per identi[U+FB01]car el cas pitjor, cal identi[U+FB01]car els casos per un mida qualsevol, per totes les mides. Els valors de l'entrada pel cas pitjor compleixen alguna propietat i és aquesta, la que identi[U+FB01]ca el cas. Per altra banda, l'anàlisi de l'e[U+FB01]ciència espacial pot ser realitzat de manera paral·lela al temporal.

La notació asimptòtica utilitza les següents 3 eines:

1. **La gran O.** $O(g(n))$ "o de g de n". Donades $f = f(n)$ i $g = g(n)$, es diu que la funció f pertany al conjunt $O(g)$ si quan n es fa molt gran, f no creix més ràpidament que g.
2. **O zeta.** $\blacksquare(g(n))$ "zeta de g de n". Donades $f = f(n)$ i $g = g(n)$, es diu que la funció f pertany al conjunt $\Theta(g)$ si quan n es fa molt gran, f creix com g.
3. **Omega.** $\Omega(g(n))$ "omega de g de n". Donades $f = f(n)$ i $g = g(n)$, es diu que la funció f pertany al conjunt $\Omega(g)$ si quan n es fa molt gran, f no creix més lentament que g.

O, Θ, o Ω com a tals, sense parèntesis, són col·leccions de conjunts. Per l'estudi de l'eficiència dels algoritmes, utilitzarem els conjunts Θ sempre que puguem per l'e[U+FB01]ciència temporal i espacial, per l'e[U+FB01]ciència temporal si no podem a[U+FB01]tar amb prou precisió el seu temps, podem utilitzar els conjunts O i per l'espacial tant la O com el Ω. Per qualsevol algorisme tindrem que el temps que triga a obtenir una sortida a partir d'una entrada de mida n, pertanyerà a alguna $\Theta(f(n))$. Per exemple, si tenim un algorisme que no te en compte l'entrada, tindrà un creixement de $\Theta(1)$. En bucles, per exemple:

1. for(i=0;i<n;i++){} i for(i=0;i<n;i = i + 4). Tindran una complexitat d'O(n).
2. for(i=1;i<n;i = i * 2). Tindrà una complexitat de O(log n)

La potència de l'exponent de les funcions polinòmiques ens revela el nivell d'aniuament dels bucles, és a dir, si un algorisme té complexitat $\Theta(n^2)$ voldrà dir que té un bucle dins d'un bucle, i si te complexitat $\Theta(n^3)$ voldrà dir que té un bucle dins d'un bucle dins d'un altre bucle. (**??**)

2.2.2 Classes de complexitat computacional.

Una de les preguntes més importants sobre un problema computacional és quin temps i quins recursos requereix per la seva computació, i això és al que es respon quan s'estudia la complexitat computacional d'un algoritme. Té la finalitat de trobar l'algoritme més òptim per resoldre un problema, tot i així hi ha problemes complexes, problemes pels quals no existeix cap algoritme per resoldre'ls en temps satisfactoris. Així doncs, s'han classificat els problemes computacionals, més concretament els problemes decisionals, aquells problemes computacionals que tenen com a resultat cert o fals, si o no, 0 o 1, per classes segons la seva complexitat temporal.

En primer lloc tenim la classe de problemes **P,** que són el conjunt de problemes decisionals decidibles en temps polinòmic, és a dir, un problema decisional pertany a la classe **P** si es pot implementar un algorisme polinòmic que, davant una instància qualsevol del problema, ens doni com a sortida la solució del problema. La següent classe de problemes decisionals és l'**NP** que són el conjunt de problemes decisionals decidibles en temps polinòmic indeterminista, és a dir, si un problema decisional pertany a la classe **NP** llavors es pot implementar un algorisme polinòmic que, per qualsevol instància positiva del problema, la sortida de l'algorisme sigui 1, i si la solució del problema és

0, llavors la sortida de l'algorisme no es coneixerà. Els problemes de la classe **NP,** ens donaran respostes polinòmiques a instancies positives, per tant un problema serà **NP** si quan tenim una solució podem comprovar, en un temps polinòmic, que efectivament és una solució. La següent classe és la **NP-dur** (**NP-hard**), però abans necessitem la següent definició:

Reducció Polinòmica. *Donats els problemes decisionals* $L \subseteq E$ *i* $L' \subseteq E'$*, diem que un algorisme polinòmic* $R : E \rightarrow E'$ *és una reducció polinòmica de* $L \subseteq E$ *a* $L' \subseteq E'$*, si quan* $x \in L \Rightarrow R(x) \in L'$*, i a la inversa, que quan* $R(x) \in L' \Rightarrow x \in L$*.*

On L és el conjunt de les instàncies possibles d'entrada al problema que ens retorna cert, o sigui, el conjunt de les veritats i E és el conjunt total d'instàncies del problema, $L \subseteq E$ és un problema decisional i x és una instancia.

Per tant, un problema és **NP-dur** si qualsevol problema NP es pot reduir a ell. És a dir, és tant o més difícil que qualsevol problema **NP.** Els problemes **NP-durs** són alguns dels problemes d'**NP** que no estan a **P,** però poden no pertànyer a **NP**. I per últim la classe **NP-complet,** els problemes d'aquesta classe són aquells que són **NP-durs** i pertanyen a **NP**.

Capítol 3

Preàmbul a la mecànica quàntica

Per aquest capítol he resumit el capítol 3 del llibre "*Quantum computing for computer scientists*"[3]en ell proporcionen una forma de passar de la de la teoria clàssica a la quàntica mitjançant la teoria del grafs i les matrius que ho fa bastant entenedor.

Si tenim un sistema determinista clàssic, en el qual totes les entrades tenen una sortida determinada no lligada a l'atzar, el podem representar amb un graf dirigit sense pesos. Si tenim aquest estat pel nostre sistema $X = [11, 35, 1, 8, 9]^T$ i sabem la dinàmica del sistema, és a dir, com canvia, el podem representar amb un graf dirigit de la següent forma:

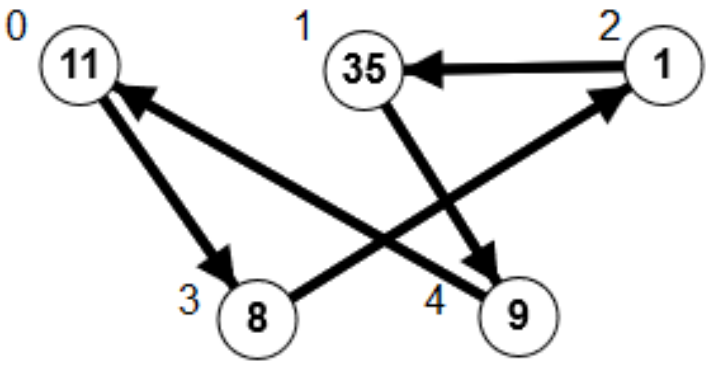


Figura 3.1: Graf dirigit pel sistema X

i també el podem representar amb la seva matriu d'adjacències:

$$\mathrm{M} = \begin{array}{c} \\ \mathbf{0} \\ \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \\ \mathbf{4} \end{array} \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad (3.0.1)$$

Si realitzem la següent operació, obtenim el següent estat del sistema:

$$\mathrm{MX} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{bmatrix} 11 \\ 35 \\ 1 \\ 8 \\ 9 \end{bmatrix} = \begin{bmatrix} 9 \\ 1 \\ 8 \\ 11 \\ 35 \end{bmatrix} = \mathrm{Y} \quad (3.0.2)$$

La mecànica quàntica treballa de la mateixa manera, els estats del sistema es representen per vectors columna i la dinàmica del sistema es representa amb matrius. A més si tenim dues o més matrius que afecten al nostre estat del sistema, aplicant el resultat del producte entre elles serà com si haguéssim aplicat una darrere de l'altre.

Tot i així en la mecànica quàntica existeix un inherent indeterminació en el coneixement del estat físic i, a més, els estats canvien segons les lleis de la probabilitat, és a dir, que l'evolució del sistema ve donada per la probabilitat de que un estat es transformi en un altre.

Així doncs passem a parlar de sistemes probabilístics. Imaginem que tenim una bala de vidre i l'estat del sistema ens indica quina probabilitat té de trobar-se a cada vèrtex del graf, $X = [\frac{1}{5}, \frac{3}{10}, \frac{1}{2}]$, un sistema d'aquest tipus es representaria amb un graf amb pesos,

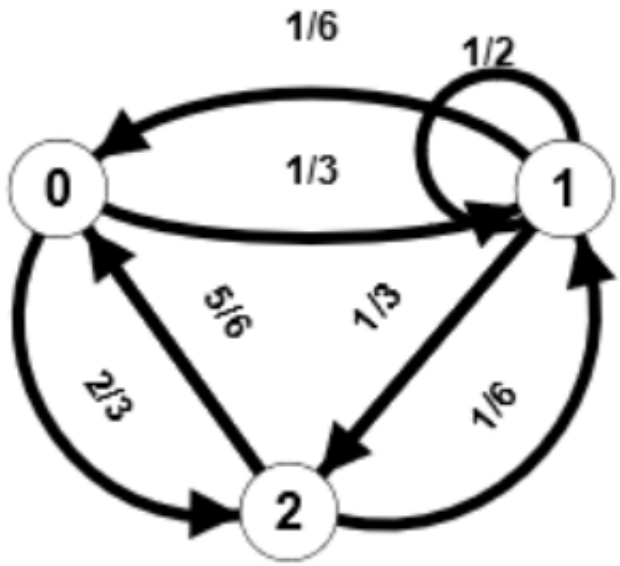


Figura 3.2: Graf dirigit pel sistema X

en aquest cas la nostre matriu d'adjacències seria:

$$M = \begin{bmatrix} 0 & \frac{1}{6} & \frac{5}{6} \\ \frac{1}{3} & \frac{1}{2} & \frac{1}{6} \\ \frac{2}{3} & \frac{1}{3} & 0 \end{bmatrix}$$

(3.0.3)

Igual que l'anterior exemple, per conèixer el següent estat hauríem de realitzar el producte MX.

Ara bé, el sistemes quàntics difereixen a aquests sistemes en que les probabilitats no s'expressen en nombres racionals, sinó que s'expressen en nombres complexes, aquesta diferencia comporta un canvi molt important, i és que els nombres complexes al sumar-se poden anular-se entre ells i disminuir les probabilitats, per exemple, si tenim $c_1 = 5+3i$ i $c_2 = -3-2i$, la probabilitat ens la donaria el seu mòdul al quadrat, així que, $|c_1|^2 = 34$ i $|c_2|^2 = 13$ i si sumem $c_1 + \ c_1 = 2+i$ que el seu mòdul és $|2+i|^2 = 5,5$ és menor que els mòduls de c_1 i c_2. El fet que els nombres complexos es puguin anul·lar els uns als altres quan se sumen té un significat físic ben definit en la mecànica quàntica. Es coneix com a interferència i és un dels conceptes més importants de la teoria quàntica que veurem més endavant.

Capítol 4

Mecànica quàntica

La mecànica quàntica és la ciència que estudia el comportament i les característiques dels sistemes físics microscòpics i les partícules microscòpiques. Va ser fundada a principis del segle XX per tal d'explicar diversos resultats experimentals de fenòmens d'origen microscòpic que no es podien entendre amb la física clàssica.

Per aquest capítol he resumit el capítol 2.2 del llibre *"Quantum Computation and Quantum Information"* [2] i el capítol 4 del llibre *"Quantum computing for computer scientists"*[3]

4.1 Experiments importants en la mecànica quàntica

Per explicar alguns dels fenòmens de la mecànica quàntica utilitzarem alguns dels experiments més importants que van treure a la llum comportaments de les partícules inexplicables amb els mètodes clàssics.

4.1.1 Experiment de Stern–Gerlach i l'espín

Per definir el gir d'un cos sobre un altre o sobre si mateix utilitzem la magnitud de moment angular, que depèn de la massa del cos, de la seva distribució respecte l'eix i de la velocitat de gir.

$$\overrightarrow{L} = \overrightarrow{r} \times m \cdot \overrightarrow{v} \tag{4.1.1}$$

On, $\overrightarrow{L}$ és el moment angular,

$\overrightarrow{r}$ és el vector posició del cos respecte el un punt O i

$m \cdot \overrightarrow{v}$ és la massa del cos pel vector de velocitat.

Durant la segona dècada del segle XX el científics es van qüestionar si l'electró, a més del moment angular orbital al voltant de l'àtom, podria tenir un moment angular intrínsec, és a dir, sobre el seu propi eix, igual que la Terra que orbita al voltant del sol

i sobre si mateixa a la vegada. Aquesta qüestió va sorgir l'any 1925 per l'observació de certes anomalies a l'espectre de l'electró per part dels físics George Uhlenbeck i Samuel Goudsmit, va ser llavors quan es va proposar per primer cop l'existència de l'espín de l'electró, és a dir, el moment angular intrínsec de l'electró.

Curiosament l'existència de l'espín es pot demostrar amb un experiment dut a terme l'any 1922 pels físics Otto Stern i Walther Gerlach, en el que es buscava provar la "quantització de l'espai" associada al moment angular orbital dels electrons atòmics, ja que la predicció feta per l'"antiga"teoria quàntica, era que els components espacials del moment angular només podien prendre valors discrets, de manera que la direcció del vector de moment angular es limitava només a un nombre concret de possibilitats. Això es podria provar fent ús del fet que un electró en òrbita dona lloc a un moment magnètic proporcional al moment angular orbital de l'electró, per tant, mesurant el moment magnètic d'un àtom s'hauria de poder determinar si existia o no la quantització de l'espai. L'experiment, que podem observar a la figura 4.1.1., consistia en passar un feix d'àtoms neutres de plata a través d'un camp magnètic heterogeni i observar com el feix es desviava per la força exercida pel camp en el moment magnètic dels àtoms. El detector era una placa de vidre sobre la qual es dipositaven els àtoms de plata del feix desviat, s'esperava un resultat aleatori dins d'un rang, però en canvi, van observar que hi havia dos grups diferenciats, és a dir, només hi havia dos valors de moment magnètic.

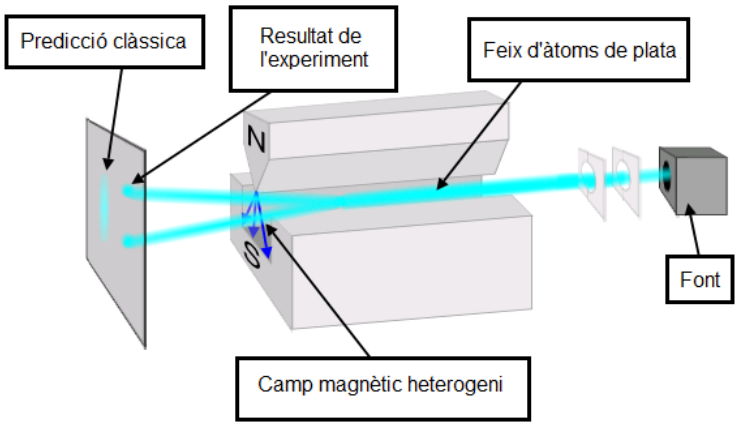


Figura 4.1: Imatge representativa de l'experiment de Stern i Gerlach.

En un principi els resultats de l'experiment van confirmar la quantització de l'espai, però anys més tard van reafirmar l'existència de l'espín, ja que els dos valors que es van observar corresponen amb els valors que pot prendre el moment angular de l'espín d'un electró $\pm\frac{\hbar}{2}$, on $\hbar$ és la constant de Planck ($h = 6,626070150\,(69)\cdot 10^{-34}\ Jxs$) dividida entre 2π

En definitiva, l'espín és una propietat intrínseca de les partícules microscòpiques, sense analogia en el món clàssic, ja que, tot i que hem parlat d'un moment angular intrínsec, les micropartícules no giren, de fet es creu que són partícules puntuals sense estructura interna. L'espín està quantitzat, no pot prendre valor aleatoris i sempre serà el mateix valor. És un magnitud vectorial, té una direcció. Les partícules elementals,

com l'electró, el protó o el neutró tenen espín ½ i per a cada direcció donada a l'espai, només tenen dos estats bàsics pel seu espín. Per a l'eix vertical, aquests estats tenen un nom: espín amunt $|\uparrow\rangle$ i espín avall $|\downarrow\rangle$ que matemàticament els podem representar com:

$$|\uparrow\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$|\downarrow\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

De la mateixa manera que hi ha lleis de conservació de moment, moment angular, l'energia i altres propietats físiques, també hi ha una llei de conservació de l'espín total d'un sistema quàntic. Això vol dir que en un sistema aïllat la quantitat total de spin ha de romandre igual.

4.1.2 Experiment de la doble escletxa i la dualitat ona-partícula

La dualitat ona-partícula és un dels conceptes fonamentals en la mecànica quàntica, en què el mateix sistema físic pot presentar en diferents circumstàncies, propietats semblants a les partícules o a les ones. Aquesta propietat dels sistemes físics, coneguda també com a "superposició d'estats", es pot demostrar amb l'experiment de la doble escletxa.

L'experiment de doble escletxa es va considerar de tres formes diferents: realitzat amb partícules macroscòpiques, amb ones i amb electrons. Els dos primers experiments només mostren el que esperem veure segons la nostra experiència quotidiana. És el tercer el que mostra el comportament contraintuïtiu dels sistemes microscòpics: una peculiar combinació de comportament de partícules i ones que no es pot entendre en termes dels conceptes de física clàssica.

Experiment amb bales.

Per aquesta versió del experiment hem d'imaginar que tenim un metralladora que dispara per tota una pantalla amb dues obertures que donen a una paret que hi ha darrera, el primer que pensarem és que la paret quedarà foradada en les parts on es troben les obertures de la pantalla, i així serà.

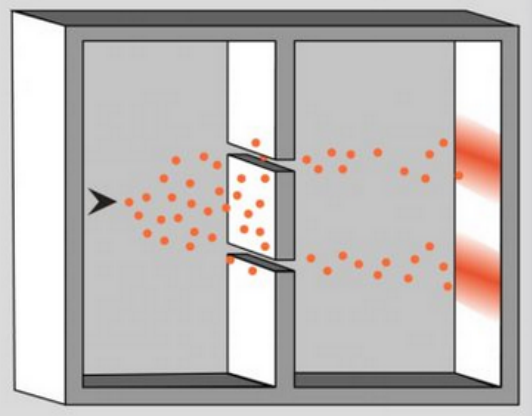


Figura 4.2: Representació de l'experiment de doble escletxa amb bales.

Com podem observar a la figura 4.2, la majoria de les bales s'agrupen en les dues línies del patró dibuixat a la paret i corresponent amb les obertures de la pantalla.

Experiment amb ones.

Si en lloc de bales utilitzem ones, l'experiment resulta com podem observar a la figura .

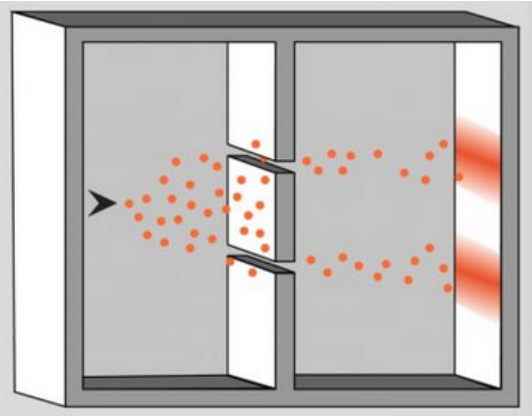


Figura 4.3: Representació de l'experiment de doble escletxa amb ones.

En aquest cas, l'ona que enviem a la primera pantalla es divideix en dos per passar per les obertures i aquestes dues noves ones, **interfereixen** l'una amb l'altre creant un patró diferent a la paret final. Aquest patró és el resultat de la **superposició** entre elles i s'anomena **patró d'interferència**.

Experiment amb electrons.

Ara imaginem que en lloc de bales o ones treballem amb electrons, per fer-ho canviarem una mica l'experiment, en lloc de dues obertures tindrem dues caixes i en una d'elles s'hi dipositarà un electró, però serà de forma aleatòria i no sabrem en quina es

troba. Els passos de l'experiment seran: obrirem una de les caixes, esperarem per si en surt l'electró i la tancarem, seguidament farem el mateix amb l'altre caixa i així varies vegades. Aquest experiment ens dona un resultat semblant al de les bales, de fet és el que s'esperava. Però ara fem un petit canvi, en lloc d'obrir una caixa darrera de l'altre, les obrirem a la vegada, hauria de ser el mateix que l'anterior, però no, ens dona el resultat que podem veure a la figura 4.1.4.

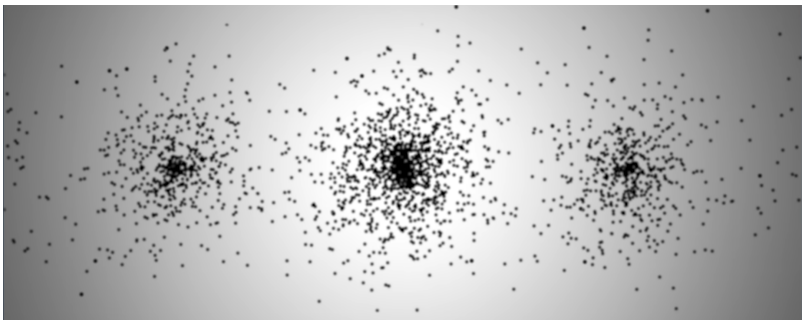


Figura 4.4: Patró en el que els electrons col·lisionen contra la pantalla.

Podem observar a la 4.4 que el patró en lloc de dibuixar dues franges com en l'experiment amb bales en dibuixa tres, cosa que s'assimila més al patró que ens dona l'experiment amb les ones. Sembla com si d'alguna manera l'electró no només estigués en una de les caixes, sinó que es trobés a les dues a la vegada i es comportés com una ona creant interferències. Per comprovar si es així, repetirem l'experiment amb un mediador a cada caixa, per saber on es troba l'electró a cada moment. Però resulta que, efectivament, l'electró es troba cada vegada en tan sols una caixa i el resultat es el patró del primer experiment.

Així doncs, sembla que obtenir informació sobre a quina caixa es troba l'electró significa que estem identificant l'electró com una partícula i només les partícules passen per una caixa o per l'altre, però si no determinem a quina caixa es troben, els electrons es comporten com ones lliures de sondejar la presència de cada caixa i donant així un patró d'interferència.

4.2 Postulats de la mecànica quàntica

Per si mateixa la mecànica quàntica no ens diu quines lleis ha de complir un sistema físic, però proporciona un marc matemàtic i conceptual per al desenvolupament d'aquestes lleis. Aquest marc matemàtic té com a base un seguit de postulats que proporcionen una connexió entre el món físic i el formalisme matemàtic de la mecànica quàntica, i seguidament els estudiarem. Hem de tenir en compte, però, que un postulat és una afirmació indemostrable que sempre està subjecte a contínua revisió i comprovació.

4.2.1 Postulat I

Qualsevol sistema físic aïllat s'associa a un espai vectorial complex amb producte escalar, és a dir, un espai de Hilbert, conegut com *espai d'estats* del sistema. El sistema es descriu de forma completa amb el seu *vector d'estat,* que és un vector unitari en el sistema d'espai d'estats.

La mecànica quàntica no ens diu, per a un sistema físic determinat, quin és l'espai d'estat d'aquest sistema ni ens diu quin és el vector d'estat del sistema. El sistema mecànic quàntic més senzill, és el qbit o bit quàntic.

4.2.2 Postulat II

L'evolució d'un sistema quàntic tancat es descriu mitjançant una transformació unitària. És a dir, l'estat $|\psi\rangle$ del sistema en el moment $t1$ està relacionat amb l'estat $|\psi'\rangle$ del sistema en el moment $t2$ per un operador unitari U que depèn només dels temps $t1$ i $t2$,

$$|\psi'\rangle = U|\psi\rangle \tag{4.2.1}$$

Donat un sistema quàntic en la vida real, la seva evolució respecte el temps es pot descriure amb l'equació d'Schrödinger:

$$\imath\hbar\frac{d|\psi\rangle}{dt} = H|\psi\rangle \tag{4.2.2}$$

on i és la unitat imaginaria i $\hbar$ és la constant reduïda de Planck. La mecànica clàssica va ensenyar als físics que l'energia global d'un sistema aïllat es conserva al llarg de la seva evolució. L'energia és observable i, per tant, per a un sistema quàntic concret és possible escriure una matriu hermítica que la representi (aquesta expressió, per descomptat, variarà d'un sistema a un altre). Aquest observable s'anomena hamiltonià del sistema, indicat per H i en la pràctica absorbeix $\hbar$, igualant el seu valor a 1.

Igual que en l'anterior postulat, la mecànica quàntica no ens diu quin és l'espai d'estats d'aquest sistema ni el vector d'estat, ni tampoc quins operadors unitaris U descriuen la dinàmica quàntica del món real, només ens assegura que l'evolució de qualsevol sistema quàntic tancat es pot descriure d'aquesta manera.

4.2.3 Postulat III

En física clàssica assumim implícitament que el fet de mesurar un sistema el deixa en el mateix estat en el que ja estava abans de mesurar-lo, i que el resultat d'una mesura en un estat ben definit és previsible, és a dir, si sabem l'estat amb absoluta certesa, podem anticipar el valor de la mesura en aquell estat. Però en física quàntica aquests supòsits són erronis, tal com han demostrat les investigacions a escala subatòmica els sistemes

es pertorben i es modifiquen al ser mesurats, com podem observar, per exemple, en l'experiment de doble escletxa. A més, només es pot calcular la probabilitat d'observar valors específics: la mesura és inherentment un procés no determinista.

El tercer postulat descriu els efectes de les mesures en un sistema quàntic, tenint en compte que un observable és tota propietat de l'estat d'un sistema que pot ser determinada ("observada") per alguna seqüència d'operacions físiques:

Quan un sistema es troba en l'estat $|\psi\rangle$, la mesura d'un observable A donarà com a resultat un valor propi a del operador corresponent a A. L'estat del sistema després de mesurar-lo serà un vector propi associat al valor propi a.

4.2.4 Postulat IV

Aquest postulat descriu com es construeix l'espai d'estats d'un sistema compost a partir dels espais d'estats dels sistemes que el componen.

L'espai d'estats d'un sistema físic compost és el producte tensorial dels espais d'estats dels sistemes físics que el componen. Si tenim n sistemes, numerats de 1 a n, i cada sistema i del conjunt es troba a l'estat $|\psi_i\rangle$, llavors l'estat conjunt del sistema total és $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$.

Un dels fenòmens més importants en la mecànica quàntica està lligat amb aquest postulat i és **l'entrellaçament** quàntic, que és un fenomen físic que es produeix quan es generen, interactuen o comparteixen proximitat espacial un parell o un grup de partícules de manera que l'estat quàntic de cada partícula del parell o del grup no es pot descriure independentment de l'estat de les altres, fins i tot quan les partícules es troben separades per una gran distància. Matemàticament els estats que es no es puguin dividir en el producte tensorial dels estats dels subsistemes constituents voldrà dir que estaran entrellaçats.

Capítol 5

Computació quàntica

Paradigma de computació que s'ocupa d'aplicar fenòmens quàntics a la resolució dels problemes habituals en la informàtica clàssica i, en general, a la realització d'operacions sobre conjunts de dades.

La computació quàntica te com a maquinari ordinadors quàntics, que enlloc de transistors utilitzen electrons i això fa que puguem aprofitar-nos dels fenòmens i propietats que hem explicat al capítol , com l'espín, la superposició, l'entrellaçament... Però també hi han limitacions, ja que s'han de complir tots els postulats, i un dels grans problemes és la mesura, ja que com hem explicat en el Postulat III amb ella perdem informació, i **col·lapsem** el sistema, és a dir, un cop mesurat no tornarem a tenir el sistema com el teníem i no podem recuperar la informació que hem perdut. A més hi han altres limitacions, com la impossibilitat de copiar un estat, ja que no el coneixem, per conèixer-lo l'hem de mesurar i llavors el col·lapsaríem i el perdríem, per tant no és possible copiar-lo mentre operem amb ell.

En els següents apartats veurem com es poden aprofitar aquests fenòmens, com es pot lidiar amb aquestes limitacions, i les analogies entre la computació clàssica i la quàntica.

Per aquest capítol he resumit el capítols 1.2, 1.3 i 4 del llibre "*Quantum Computation and Quantum Information*" [2] i els capítols 4 i 5 del llibre "*Quantum computing for computer scientists*"[3]

5.1 Bit quàntic

En computació clàssica el bit és la unitat mínima d'informació, té dos possibles estats 0 i 1 i només es pot trobar en un d'aquest estats a la vegada, és a dir, si 0 és apagat i 1 encès no pot estar apagat i encès en el mateix moment. En computació quàntica el seu anàleg és el qbit o bit quàntic, que també té dos estats elementals $|0\rangle$ i $|1\rangle$, que corresponen al 0 i 1 del bit clàssic respectivament, però en lloc de representar-se per un valor es representen amb vectors:

$$|\,0\rangle \;=\; \begin{bmatrix} 1 \\ 0 \end{bmatrix} |\,1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{5.1.1}$$

S'aprofita del fenomen quàntic de la superposició, per tant, un qbit pot trobar-se en l'estat $|\,0\rangle$ i $|\,1\rangle$ a la vegada. Tot i així, com que els sistemes quàntics al ser mesurats col·lapsen, el qbit quan s'observi col·lapsarà a un dels dos estats elementals i no en podrem observar la superposició. Un qbit es pot trobar en el següent estat:

$$|\,\psi\rangle = \;\alpha|\,0\rangle + \beta|\,1\rangle \tag{5.1.2}$$

que es tracta d'una **combinació lineal** dels dos estats elementals, on α i β són nombres complexes, tot i que per certes finalitats els representarem amb nombres reals, i s'anomenen **amplituds de probabilitat**, compleixen la següent igualtat $|\alpha|^2 + |\beta|^2 = 1$ on, $|\alpha|^2$ és la probabilitat de que el qbit col·lapsi a l'estat $|\,0\rangle$ i $|\beta|^2$ és la probabilitat de que col·lapsi a $|\,1\rangle$. Com podem observar, els estats $|\,0\rangle$ i $|\,1\rangle$ són una base canònica de $\mathbb{C}^2$. Posem un exemple numèric, si tenim el següent estat,

$$|\,\psi\rangle = \frac{1}{\sqrt{2}}|\,0\rangle + \;\frac{1}{\sqrt{2}}|\,1\rangle \tag{5.1.3}$$

sabem que la probabilitat de que al ser mesurat col·lapsi a l'estat $|\,0\rangle$ i $|\,1\rangle$ és la mateixa, $\left|\frac{1}{\sqrt{2}}\right|^2$ = 50%.

Un forma més visual amb la que pensar en els qbits és amb la seva representació geomètrica a l'esfera de Bloch, qualsevol punt de l'esfera de Bloch és un estat quàntic o *qbit* i es pot expressar com:

$$\psi\rangle \;=\; \cos\frac{\theta}{2}\;|\,0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|\,1\rangle \tag{5.1.4}$$

Els números θ a ϕ defineixen un punt de l'esfera tridimensional de la unitat, tal com es mostra a la figura 5.1. Tot i així aquesta representació només serveix per un qbit, no per múltiples.

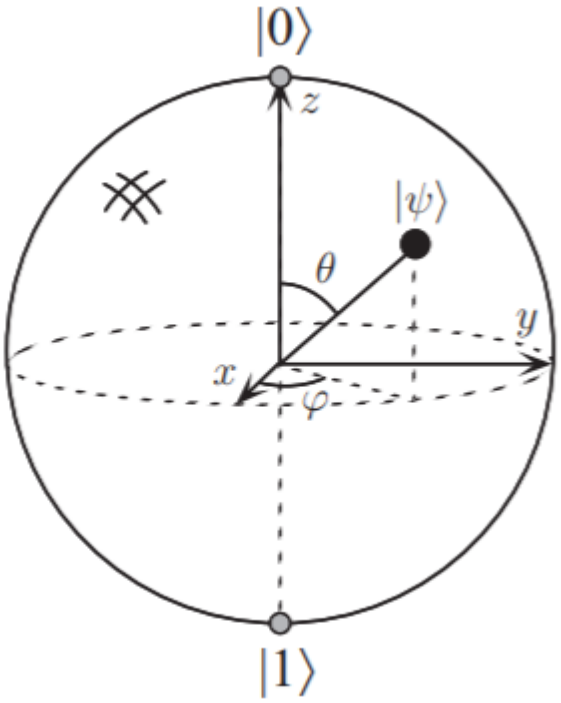


Figura 5.1: Representació d'un qbit a l'esfera de Bloch

Evidentment, podem tenir un sistema amb més d'un qbit, per exemple, l'estat base d'un sistema de dos qbits serà:

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \tag{5.1.5}$$

on els estats base són el resultat dels següents productes tensorials:

$$|0\rangle \otimes |0\rangle = |00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.1.6}$$

$$|0\rangle \otimes |1\rangle = |01\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{5.1.7}$$

$$|1\rangle \otimes |0\rangle = |10\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \tag{5.1.8}$$

$$|\,1\rangle \otimes |\,1\rangle = |\,11\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{5.1.9}$$

ja que com hem après en el postulat IV "l'espai d'estats d'un sistema físic compost és el producte tensorial dels espais d'estats dels sistemes físics que el componen". I igual que en els sistemes d'un sol bit, els resultats possibles a observar de l'estat són $|\,00\rangle, |\,01\rangle, |\,10\rangle$ o $|\,11\rangle$ amb una probabilitat $|_x|^2$. Això es pot extrapolar a la quantitat de qbits que es desitgi.

Un exemple, 1 byte, 8 bits, donen $2^8 = 256$ possibles estats però cada byte només pot estar en un d'aquests estats a cada moment, en canvi, amb 8 qbytes tenim aquests 256 estats a la vegada, el gran inconvenient és que d'aquests estats només en podem observar un, per tant cal buscar una manera per operar amb ells sense observar-los, i això és el que fem amb les portes quàntiques.

5.2 Portes quàntiques

Les portes lògiques realitzen manipulacions de la informació dels bits, convertint-la d'una forma a una altra, les portes quàntiques fan el mateix però amb els estats dels qbits, tot i així, aquestes manipulacions són lleugerament diferents. Si prenem com exemple la porta lògica NOT, que inverteix l'estat actual del bit per l'estat contrari, és a dir, si és 1 passa a ser 0 i si és 0 passa a ser 1, i l'apliquem als estats elementals quàntics $|\,0\rangle$ i $|\,1\rangle$, no hi hauria cap problema ja que els coneixem perfectament i els podem invertir, però què passa si tenim un estat en superposició, $|\,\psi\rangle = \alpha|\,0\rangle + \beta|\,1\rangle$ com el podríem convertir en el contrari, $|\,\psi\rangle = \alpha|\,1\rangle + \beta|\,0\rangle$ sense mesurar-lo? La resposta és amb una matriu, en el món quàntic totes les operacions que no són mesures es representen mitjançant matrius unitàries i són reversibles, que això vol dir que coneixent l'operació i el resultat podem trobar el valor exacte d'entrada. Físicament el que fan les portes quàntiques és modelar l'efecte del dispositiu que manipula l'espín del qbit sense mesurar-lo.

Qualsevol matriu unitària pot ser una porta quàntica, però no totes ens interessen. Per tant, tornant a la porta NOT, la seva versió quàntica seria la següent:

$$X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{5.2.1}$$

Per il·lustrar-ho matemàticament, si tenim el següent estat d'un qbit en superposició:

$$|\,\psi\rangle = \frac{1}{2}|\,0\rangle + \frac{\sqrt{3}}{2}|\,1\rangle$$

el representem com un vector i li apliquem la porta NOT, que això vol dir multiplicar el vector de l'estat per la matriu X :

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \end{bmatrix}$$

i podem veure com l'estat inicial s'ha invertit. Així és com funcionen totes les portes quàntiques, són matrius unitàries que aplicarem al nostre qbit o qbits i els transformaran. Encara que qualsevol matriu unitària pugui ser un porta quàntica, com hem esmentat abans, no totes ens interessen i n'hi han que s'utilitzen més sovint, així que veurem quines són les més bàsiques.

Ja hem vist la porta X que és com la NOT lògica que s'aplica a una sol qbit, ja que és una matriu 2 per 2, seguint amb portes d'un sol qbit també tenim la Z,

$$Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

que el que fa és deixar els estats igual, però canviar el signe de l'estat $|\,1\rangle$. Una altre porta molt important d'un sol qbit, és la porta Hadamard:

$$H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

el que fa aquesta porta és passar el qbit d'un estat elemental a superposició, és a dir,

$$H|\,0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}}\,|\,0\rangle + \frac{1}{\sqrt{2}}|\,1\rangle\ = |\,+\rangle$$

$$H|\,1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}}\,|\,0\rangle - \frac{1}{\sqrt{2}}|\,1\rangle\ = |\,-\rangle$$

com podem observar la superposició de l'estat $|\,0\rangle$ la podem expressar com $|\,+\rangle$ i la de l'estat $|\,1\rangle$ com $|\,-\rangle$.

Passem a les portes amb més d'una entrada, en computació clàssica per dos entrades tenim, per exemple, les portes AND i OR, però a diferència de la NOT aquestes no es poden traduir a portes quàntiques tan fàcilment, ja que perden informació, observem les seves taules de la veritat:

OR		
Entrada A	Entrada B	Sortida
0	0	0
0	1	1
1	0	1
1	1	1

AND		
Entrada A	Entrada B	Sortida
0	0	0
0	1	0
1	0	0
1	1	1

com podem observar a les taules les transformacions que realitzen no son reversibles, ja que si fem, per exemple, 1 AND 0 és el mateix resultat que 0 AND 1 per tant no podem conèixer l'entrada exacte i perdem informació. Per poder simular una AND o una OR necessitarem més de dos qbits. La porta binària que si que podem simular amb una porta quàntica és la XOR, i ho podem fer amb la porta quàntica CNOT, *controlled-NOT,* aquesta porta rep dos qbits i retorna dos qbits però només afecta a un ja que un d'ells farà de control. Si tenim el qbit $|\, c\rangle$ de control i el $|\, t\rangle$ d'objectiu l'acció que realitzarà és $|\, c\rangle|\, t\rangle \rightarrow \ |\, c\rangle \ \ |t \otimes c\rangle$ que es pot representar amb la següent matriu:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.2.2}$$

En definitiva, el que farà la CNOT és, si $|\, c\rangle$ = 1 negarà $|\, t\rangle$, sinó el deixarà igual. Per exemple apliquem la CNOT a l'estat $|\, 10\rangle$:

$$CNOT|\, 10\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = |\, 11\rangle$$

Si ens fixem en la matriu de la CNOT, hi ha una part que ens pot resultar familiar, la matriu 2x2 que creen els dos últims números de les dues ultimes columnes i les dos ultimes files:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

és la porta NOT, i és que aquesta matriu 2x2 és la que s'aplicarà al qbit objectiu i pot ser la porta quàntica que vulguem, per tant més globalment la CNOT és la CU(controlled-U) on U és una matriu unitària 2x2 que afectarà al nostre qbit objectiu. Podríem aplicar una Hadamard, per exemple, la CH (controlled-hadamard) es representaria amb la següent matriu:

$$\text{CH} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

I per últim, un altre porta molt útil de dos qbits és la SWAP, que tal com diu el seu nom permuta els dos qbits i es representa amb la següent matriu:

$$\text{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Per acabar les portes parlarem d'una molt importat per tres qbits, la Toffoli, amb aquesta podem simular la porta AND. Aquesta porta es com la CNOT però amb dos bits de control i es representa amb la següent matriu:

$$toffoli = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

realitza la següent operació $|c_1\rangle\,|\,c_2\rangle|\,t\rangle \rightarrow |\,c_1\rangle\,|\,c_2\rangle\,|t \otimes c_1c_2\rangle$ i te la següent taula de la veritat:

Toffoli					
c_1	c_2	t	$c_1^{'}$	$c_2^{'}$	$t^{'}$
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Si ens fixem en la taula de la veritat, si assignem sempre l'estat $|\,0\rangle$ a $|\,t\rangle$ obtenim el resultat de la porta AND i si en canvi li assignem $|\,1\rangle$ obtenim la porta NAND.

Igual que en la computació clàssica en la quàntica també podem combinar aquestes portes i crear circuits per manipular els nostres qbits i executar algoritmes, tot això ho veurem detalladament en els següent capítol, on utilitzarem programació per il·lustrar cada punt.

Capítol 6

Aplicació de la computació quàntica amb Qiskit

Qiskit és un framework open-source de l'empresa IBM per treballar amb ordinadors quàntics a nivell de circuits i algoritmes. El seu objectiu és construir una base per fer més fàcil a qualsevol utilitzar ordinadors quàntics i facilitar la investigació sobre els problemes actuals en la computació quàntica. Proporciona eines per crear i manipular programes quàntics i executar-los en simuladors en un ordinador local o en ordinadors quàntics. Va veure la llum per primer cop el 7 de març de 2017 i està compost per quatre elements:

1. Qiskit Terra: Per crear programes quàntics a nivell de circuits amb codi a prop de nivell de codi màquina.
2. Qiskit Aer: Desenvolupament accelerat mitjançant simuladors i models de soroll.
3. Qiskit Ignis: Resolució de sorolls i errors,
4. Qiskit Aqua: Creació d'algoritmes i aplicacions.

La versió principal de Qiskit utilitza el llenguatge de Python, tot i que també hi ha versions Javascript i Swift, i et recomanen l'entorn de Jupyter Notebook per programar. Podem veure l'instalaccio a l'annex A.

IBM Quantum ofereix diversos ordinadors quàntics reals i simuladors d'alt rendiment per poder executar el nostre codi, així que configurarem l'entorn Qiskit per enviar feines als sistemes IBM Quantum. Creem un compte a IBM Quantum Experience per aconseguir la nostre clau única i l'importem i l'activem de la següent forma,

```
from qiskit import *
IBMQ.save_account('a5fafd72fcf4ac83d0cb3a807c70c5ec8d43412bbb1963800f9ba7ab7777800e7262b124657d5b7616521e170d2c44e56
IBMQ.load_account()
```

Ja estem llestos per utilitzar-lo, al llarg del capítol veurem les utilitats i la programació amb aquest framework a mesura que estudiem els circuits i els algoritmes quàntics.

6.1 Circuits quàntics

Els circuits informàtics clàssics consisteixen en cables i portes lògiques. Els cables transporten la informació al llarg del circuit i connecten les portes lògiques entre elles, les quals realitzen manipulacions de la informació que reben, convertint-la d'una forma a una altra. Els circuits quàntics són similars, cada línia del circuit representa un fil del circuit quàntic, que no necessàriament correspon a un fil físic, i s'hi apliquen portes quàntiques.

Tot i així, hi ha algunes característiques permeses en els circuits clàssics que no solen estar presents en els circuits quàntics. En primer lloc, no es poden crear 'bucles', és a dir, retroalimentació d'una part del circuit quàntic a una altra; diem que el circuit és acíclic. En segon lloc, els circuits clàssics permeten que els cables s'uneixin, en un cable únic resultant, això no es pot fer en els circuits quàntics ja que aquesta operació no és reversible i, per tant, no és unitària. I en tercer lloc, tampoc es permet l'operació de copiar.

Els circuits quàntics són un model de computació quàntica, i és el que segueix Qiskit. L'element per crear un circuit en Qiskit és el *QuantumCircuit(),* que pot rebre: un enter, que especificarà el nombre de registres quàntics, qbits, que tindrà el nostre circuit; dos enters, on el primer especificarà el nombre de registres quàntics i el segon el nombre de registres clàssics, bits; un registre quàntic ja creat o dos registres un quàntic i un clàssic. L'element *QuantumCircuit* compta amb mètodes que apliquen les portes lògiques als qbits que s'especifiquin i inicialitzarà tots els registres a 0. Veiem un exemple de circuit:

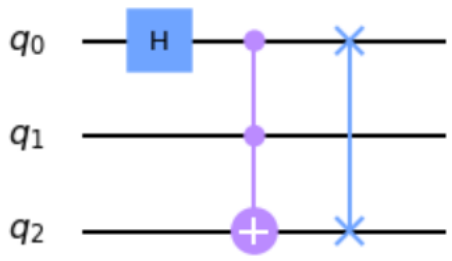


Figura 6.1: Circuit quàntic en Qiskit

El circuit de la figura 6.1 compta amb tres qbits, en primer lloc aplica la porta Hadamard en el primer qbit, seguidament una Toffoli, amb el primer i segon qbit com a qbits de control i el tercer com a objectiu, i finament aplica la porta SWAP que intercanvia els estats del primer i el tercer qbit. El codi que el crea és el següent:

```
circ3 = QuantumCircuit(3)#inicialitzem el circuit amb 3 qbits.
circ3.h(0)#apliquem la porta Hadamard en el qbit 0.
circ3.toffoli(0,1,2)#apliquem la porta toffoli amb els qbits 0 i 1 de control i el 2 d'objectiu.
circ3.swap(2,0)#apliquem la porta swap en els qbits 0 i 2.
circ3.draw(output = 'mpl')#mostrem el circuit
```

Figura 6.2: Codi circuit quàntic en Qiskit

Com podem observar el circuit de les figures 6.1 i 6.1 no compta amb ninguna mesura, per afegir-n'hi podem fer-ho de dues formes diferents. Si volem mesurar tots el qbits podem utilitzar el mètode de *QuantumCircuit mesure_all(),* aquest mètode el que fa és mesurar els qbits i enviar el resultat a registres clàssics per tal d'emmagatzemar-los. Si en canvi, per algun motiu no volem o no necessitem mesurar tots els qbits, o necessitem ordenar d'alguna forma especial el resultat de cada qbit, podem utilitzar el mètode *mesure*(), que rep parelles d'enters que indiquen a quin bit es guardarà el resultat de cada qbit a mesurar, cal tenir en compte que per utilitzar aquest mètode necessitarem afegir registres clàssics en el nostre circuit. Així doncs, afegim mesura en el circuit de la figura 6.1,

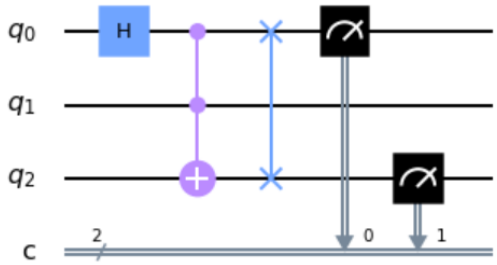


Figura 6.3: Circuit quàntic en Qiskit

```
circ3 = QuantumCircuit(3,2) #inicialitzem el circuit amb 3 qbits i 2 bits.
circ3.h(0) #apliquem la porta Hadamard en el qbit 0.
circ3.toffoli(0,1,2) #apliquem la porta toffoli amb els qbits 0 i 1 de control i el 2 d'objectiu.
circ3.swap(2,0)#apliquem la porta swap en els qbits 0 i 2
circ3.measure(0,0) #masurem el qbit 0 i el guardem en el bit 0
circ3.measure(2,1)#masurem el qbit 2 i el guardem en el bit 1
circ3.draw(output = 'mpl') #mostrem el circuit
```

Figura 6.4: Codi circuit quàntic en Qiskit

Un cop tenim el circuit totalment dissenyat ja podem executar-lo. Tenim dues opcions, utilitzar el simulador local que ens proporcions Qiskit o enviar el circuit a un ordinador quàntic real. Mostrarem les dues opcions aplicades en el circuit de les figures 6.3 i 6.4. Començarem per executar-lo amb el simulador local, crearem un nou simulador amb el mètode *get_backend()* de Aer, en el que l'indiquem per paràmetre quin tipus de simulador volem, i l'executarem amb el mètode *execute(),* passant-li per paràmetre el circuit que volem calcular i el simulador en el que el volem simular. Un

cop executat agafarem el resultat i el mostrarem en un gràfic, en el que apareixeran les probabilitats dels possibles resultats després d'haver calculat el circuit diverses vegades, per defecte 1024 si no assignem cap número diferent a la variable *shots* en els paràmetres del mètode *execute*(). Podem observar el codi d'aquest procediment a la figura 6.5 i els resultats a la figura 6.1

```
from qiskit.tools.visualization import plot_histogram #importem la funcio per mostrar la grafica
simulator = Aer.get_backend('qasm_simulator') #Creem un nou simulador
result = execute(circ5, backend = simulator).result()#executem i agafem el resultat del circuit.
plot_histogram(result.get_counts(circ5))#mostrem el resultat en una grafica
```

Figura 6.5: Codi circuit quàntic en Qiskit

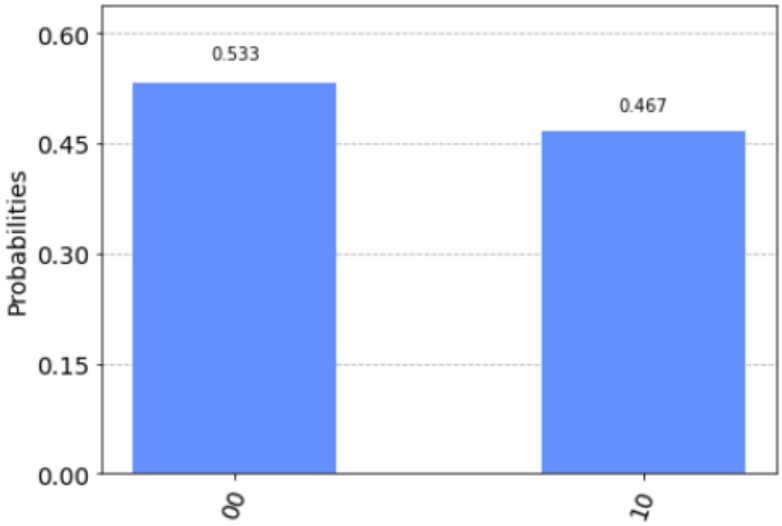


Figura 6.6: Gràfica de resultats circuit quàntic en Qiskit

Pel nostre simulador local hem utilitzat el qasm_simulator, que simula un ordinador quàntic amb cert soroll. Com es pot observar a la figura , el 53,3% de les vegades el resultat del primer i el tercer qbit ha sigut 0 i el 46,7% de les vegades el resultat del primer qbit ha sigut 0 i del tercer 1. Passem a executar-lo en un ordinador quàntic real, per enviar i executar el circuit a un ordinador quàntic, en primer lloc hem de tenir el nostre compte IBMQ carregat amb les instruccions que hem realitzat al principi del capítol, si ja el tenim seguim el següents pasos:

1. Inicialitzar el nostre proveïdor. L'accés als diversos serveis que ofereix IBM Quantum Experience està controlat per proveïdors i cada compte en te un d'assignat. Per saber quin tenim podem anar a l'apartat de proveïdors del nostre compte, tot i que per defecte als comptes públics se'ls assigna *ibm-q*.

2. Seleccionar el dispositiu quàntic que volem utilitzar. Cada proveïdor te certs dispositius, en el nostre compte podem veure de quins disposem i quantes tasques tenen en cua.
3. Executar amb el mètode execute(). En aquest cas per paràmetre li passarem el nostre circuit i el dispositiu quàntic que hem seleccionat.
4. Monitoritzar l'execució. Així podrem saber en quin punt es troba, per observar els resultats un cop executada.
5. Agafar els resultats i mostrar-los.

El codi dels passos 1,2,3 i 4 el podem observar a la figura 6.8.

```
from qiskit.tools.monitor import job_monitor #importem la funcio per monitoritzar l'execucio
provider = IBMQ.get_provider('ibm-q') #Inicialitzem el nostre proveidor
qcomp = provider.get_backend('ibmq_valencia') #seleccionem el dispositiu
job = execute(circ5, backend = qcomp) #exeutem
job_monitor(job) #monitoritzem l'execucio
```

Figura 6.7: Codi circuit quàntic en Qiskit

Un cop executat, la monitorització passarà pels següents estats: en validació, en cua, executant i executat. Un cop estigui executat ja podem mostrar els resultats.

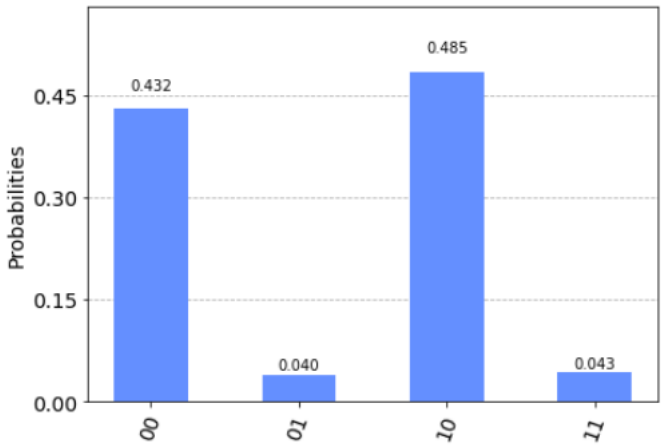


Figura 6.8: Gràfica de resultats circuit quàntic en Qiskit

A la figura 6.8 podem observar els resultat de l'execució del circuit de la figura 6.8 en un ordinador quàntic, en aquest cas a diferencia del resultat del simulador local s'han observat els resultats 01 i 11, això és degut a que els ordinadors quàntics reals tenen més soroll del que simula el simulador.

Amb aquesta visó general dels circuits, passem a parlar d'un protocol que és realment sorprenent en la computació quàntica i que podem simular amb circuits, la teleportació quàntica.

6.1.1 Teleportació quàntica

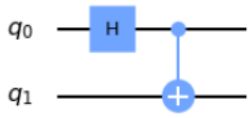


Figura 6.9: Circuit quàntic en Qiskit

Abans d'explicar la teleportació quàntica observem el circuit de la figura 6.9, aquest circuit aplica la porta Hadamard en el qbit 0, que com ja sabem el posa en superposició, després utilitza aquest qbit en superposició com a control en una porta CNOT que té com a qbit objectiu el qbit 1. Aquesta seqüencia de portes dona lloc als següents possibles estats resultants per cada estat inicial,

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

$$|\,01\rangle = \frac{|\,01\rangle + |\,10\rangle}{\sqrt{2}}$$

$$|\,10\rangle = \frac{|\,00\rangle - |\,11\rangle}{\sqrt{2}}$$

$$|\,11\rangle = \frac{|\,01\rangle - |\,10\rangle}{\sqrt{2}}$$

aquests estats resultants s'anomenen estats de Bell o parells EPR (pels seus descobridors, Einstein, Podolsky i Rosell), i tenen unes propietats quàntiques especials que fan que els dos qbits inicials es quedin lligats tot i que no existeixi cap canal que els uneixi. Els hem de tenir presents pel que explicarem a continuació.

La teleportació quàntica és una tècnica per transportar estats quàntics sense necessitat d'un canal quàntic de comunicació que lligui l'emissor i el receptor. Funciona de la següent forma, tenim dues persones, l'Alice que serà l'emissora i en Bob que serà el receptor. L'Alice i en Bob es van trobar fa molt de temps i van generar un parell EPR, al separar-se, cada un es va endur un qbit del parell EPR. Actualment es troben separats i no coneixen les seves posicions, però l'Alice necessita enviar-li un qbit del qual no en coneix l'estat a en Bob i només disposa d'un canal clàssic de comunicació, com ho pot fer? Amb la teleportació quàntica. Consisteix, a grans trets, en el següent: l'Alice combina el qbit que ha d'enviar, amb el qbit que te del parell EPR que comparteix amb en Bob, mesura els dos qbits i li envia el resultat a en Bob pel canal clàssic, segons el resultat en Bob aplica una de quatre operacions en el seu qbit del parell EPR i el resultat serà el qbit que l'Alice volia fer-li arribar. Anem pas per pas i il·lustrem-ho amb el model de circuits. Per poder simular la teleportació amb un circuit ens calen 3 qbits, un serà el que l'Alice ha d'enviar, i els altres dos seran els que formaran el parell EPR,

també necessitem dos bits clàssics per mesurar els dos qbits de l'Alice i enviar-los a en bob.

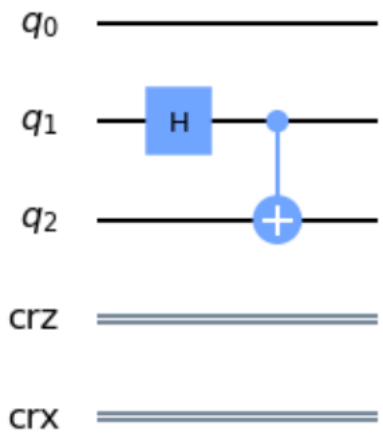


Figura 6.10: Circuit quàntic en Qiskit

Els dos qbits de l'Alice en el nostre circuit, representat a la figura 6.10, seran el q_0 i el q_1 i el d'en Bob serà el q_2, per tant el parell EPR el formaran el q_1 i el q_2, així que en primer lloc aplicarem la porta Hadamard en el q_1 i una CNOT amb el q_1 com a control i el q_2 com a objectiu. Així posarem el dos qbits en estat de Bell, per tal d'emular l'estat inicial i poder realitzar la teleportació. El següent pas és que l'Alice operi amb els seus qbits, i per això els hi aplicarà una CNOT amb el qbit que vol enviar com a control i el seu qbit del parell EPR com a objectiu, seguidament aplicarà una Hadamard a q_0, mesurarà els seus qbits i guardarà els resultats en els registres clàssics corresponents. Podem veure el resultat en circuit d'aquestes accions a la figura 6.12

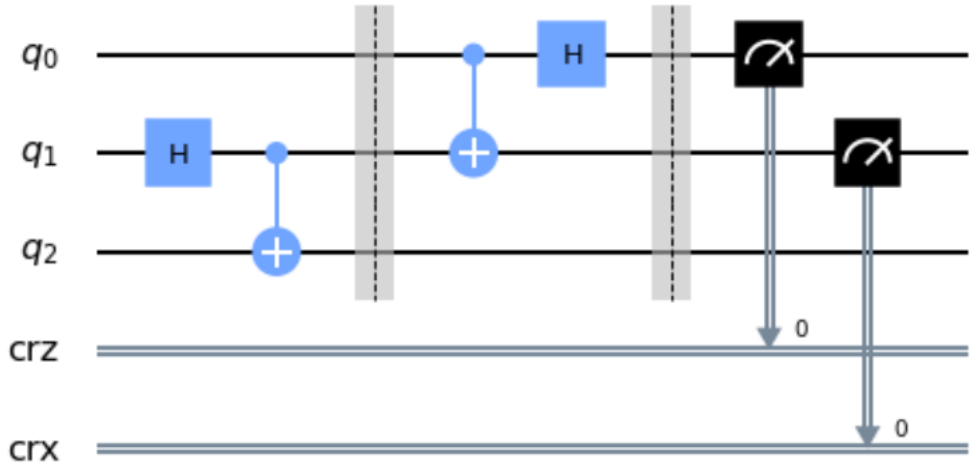


Figura 6.11: Circuit quàntic en Qiskit

Finalment, l'últim pas és enviar els bits resultants a en Bob perquè aquest realitzi o no operacions amb el seu qbit del parell EPR. Per saber com ha de procedir, l'Alice i en Bob han acordat el següent protocol, si en Bob rep 00 significarà que no ha de fer cap operació, si rep 01 aplicarà la porta X al seu qbit, si rep 10 la porta Z i si rep 11 ambdues portes, per tant el que significa és que si el q_0 col·lapsa a 1 aplicarem la porta X i si el q_1 col·lapsa a 1 aplicarem la porta Z, per tant és important l'ordre dels nostres bits, i per això hem creat dos registres clàssics diferenciats.

```
# Aquesta funció rep un QuantumCircuit (qc), un enter (qubit)
# i dos ClassicalRegisters (crz & crx).
def calcul_bob(qc, qubit, crz, crx):

    qc.x(qubit).c_if(crx, 1) # Aplicarà x si crx és 1
    qc.z(qubit).c_if(crz, 1) # Aplicarà z si crz és 1
```

Figura 6.12: Funció per aplicar portes quàntiques

A la figura 6.1.13, es por observar la funció que realitza l'últim pas de la teleportació. En ella utilitzem el mètode *c_if(registre_classic,valor)* de les portes quàntiques, que aplica la porta si el valor que conté el registre clàssic és el mateix que valor que li indiquem com a segon paràmetre.

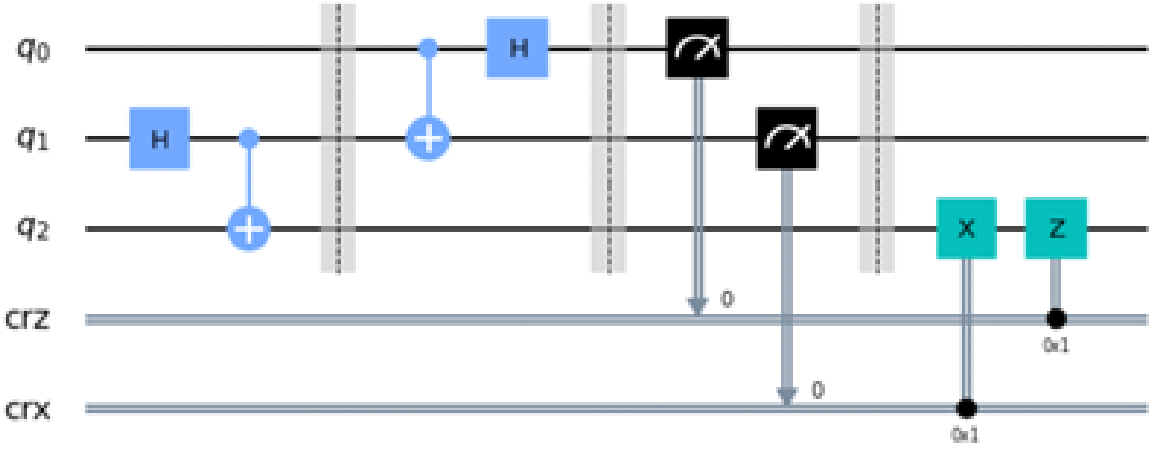


Figura 6.13: Circuit quàntic en Qiskit

Podem observar el circuit resultant a la figura 6.13, aquest circuit és el que representa una teleportació quàntica. Per veure si realment funciona, el simularem amb l'statevector_simulator, que en lloc de simular el funcionament d'un ordinador quàntic com el qasm_simulator, realitza els càlculs matemàtics i mostra els resultats amb vectors. Per poder simular-ho correctament, necessitem que l'estat del qbit que vol enviar l'Alice no sigui 0, així que crearem un estat aleatori d'un qbit amb la funció random_state() i el mostrarem amb l'esfera de Bloch,

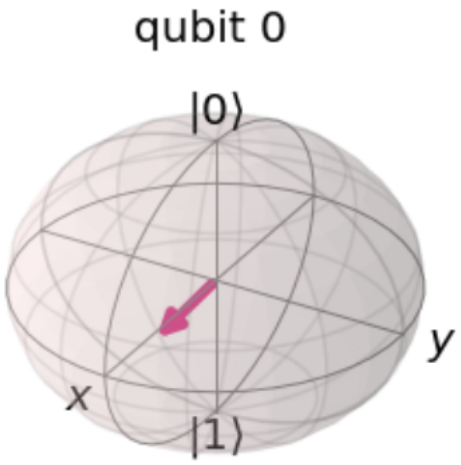


Figura 6.14: Representació d'un qbit aleatori a l'esfera de Bloch

a la figura6.14, podem observar l'estat que ens ha retornat. Inicialitzarem un qbit amb l'estat resultant que hem guardat a la variable *psi*, amb la funció Initialize() i l'assignarem al nostre q_0 amb la funció append.

podem observar el resultant representat amb esferes de Bloch a la figura .

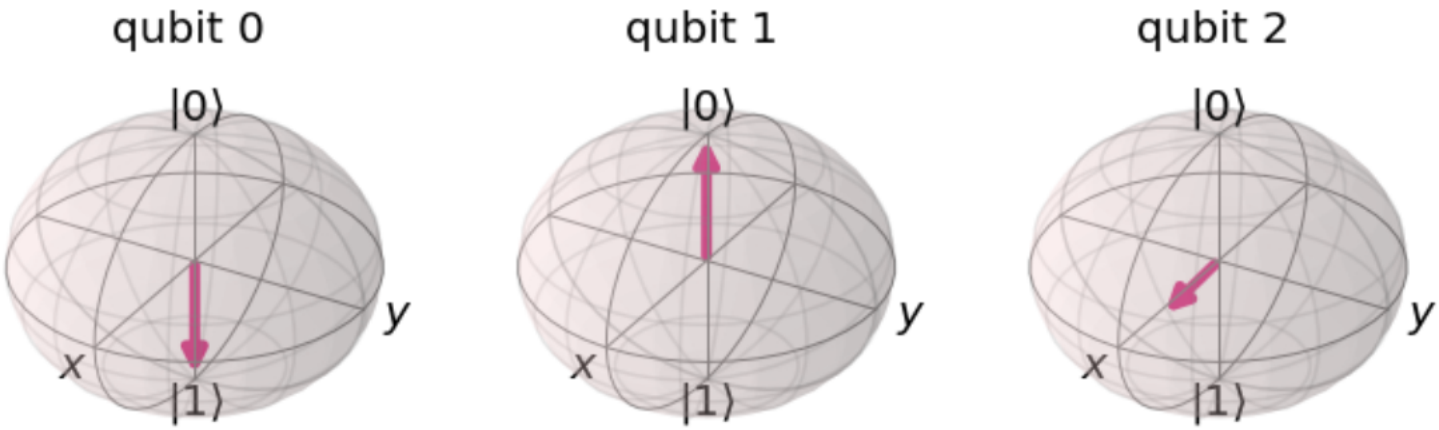


Figura 6.15: Resultat d'un circuit quàntic

Com podem observar la esfera de Bloch pel tercer qbit que és on volíem transportar es troba en la mateixa posició que el que hem creat aleatòriament, podem confirmar que el circuit ha teleportat el qbit.

6.2 Algoritmes quàntics

Dissenyar algoritmes per ordinadors quàntics és més difícil que fer-ho per ordinadors clàssics, els principals factors són els següents: en primer lloc tenim el problema de que la nostra intuïció, la que ens ajuda a la creació de algoritmes, està arrelada al món clàssic i això ens portarà a idees clàssiques que no seran eficients ni adients per el nostre algoritme quàntic, així que cal '*desactivar*' aquesta intuïció clàssica durant el procés de disseny de l'algoritme quàntic. I en segon lloc no n'hi ha prou en dissenyar un algoritme quàntic, aquest ha de ser millor que qualsevol algoritme clàssic ja existent, ja que si té el mateix rendiment que el clàssic ja no te interès.

La majoria d'algoritmes quàntics existents es descriuen amb el model de circuits quàntics, per tant tenim les mateixes limitacions que hem plantejat en el disseny de circuits, no es poden crear 'bucles', totes les operacions han de ser reversible i no es permet l'operació de copiar, però per altre banda també ens podem aprofitar dels fenòmens quàntics com la superposició i l'entrellaçament. Tot i que l'execució i la creació sigui totalment diferent, els algoritmes quàntics defineixen els mateixos problemes que els algoritmes clàssics, per tant els problemes **no decidibles** per els algoritmes clàssics seguiran sent-ho pels algoritmes quàntics. L'avantatge dels algoritmes quàntics és que, si estan ben dissenyats, poden arribar a suposar un gran canvi en la complexitat computacional d'aquell problema, reduint-la considerablement, tot i així no és fàcil, i no tots els problemes de gran complexitat poden ser millorats per algoritmes quàntics.

Dos exemples d'algoritmes que han millorat exponencialment la complexitat computacionals dels problemes respecte les seves versions clàssiques són l'algoritme de Shor i l'algoritme de Grover. L'algoritme de Shor és un algoritme de factorització d'enters, el millor algoritme clàssic per factoritzar enters te una complexitat millor que

una exponencial però molt pitjor que una polinòmica, en canvi l'algoritme de Shor te una complexitat polinòmica $O(n^3)$. Posem un exemple en temps, el que un algoritme clàssic resoldria en $28{\cdot}10^7$ anys, l'algoritme de Shor trigaria 100 segons. L'algoritme de grover és un algoritme de cerca en dades no ordenades, de forma clàssica en el pitjor del casos haurem de revisar cada dada, per tant tindrem una complexitat $d'O(n)$ on n és el nombre de dades, en canvi amb l'algoritme de Grover te una compexitat $d'O(\sqrt{n})$.

Per tenir una idea de com funciona un algoritme quàntic, mostrarem un exemple explicant i codificant amb Qiskit un dels primers algoritmes quàntics que es va dissenyar, l'algoritme de Deutsch–Jozsa.

6.2.1 Algoritme de Deutsch–Jozsa

L'algoritme de Deutsch–Jozsa és un cas simple d'un algorisme quàntic general i suposa un guany exponencial respecte el millor algoritme clàssic que descriu el mateix problema. Podem interpretar el problema que descriu com un joc, els nostres protagonistes tornaran a ser l'Alice i en Bob. L'Alice està a Amsterdam, escull un numero x entre 0 i $2^n - 1$ i li envia a en Bob que està a Boston. En Bob calcula una funció $f(x)$ i li retorna el resultat a l'Alice, aquest serà 0 o 1. Sabem que la funció que ha aplicat en Bob serà o constant, és a dir, sempre retornarà 0 o 1, o balancejada, la qual el 50% de les vegades retornarà 0 i l'altre 50% 1. Enviant-li un numero cada vegada l'Alice ha de determinar amb certesa quin tipus de funció ha escollit en Bob, de la forma més ràpida possible. Si suposem que l'Alice envia el numero representat amb n bits, en el pitjor dels cassos l'Alice li haurà d'enviar $\frac{2^n}{2+1}$ números, és a dir, la mitat més u del rang de números possibles, per poder determinar si la funció que ha escollit en Bob és constant o balancejada. Si en lloc de bits l'Alice i en Bob es poguessin enviar qbits, enviant un sol cop n qbits l'Alice podria determinar la resposta, això és el que fa l'algoritme de Deutsch–Jozsa. Per aconseguir-ho s'aprofita del **paral·lelisme quàntic**, que permet als ordinadors quàntics avaluar una funció f(x) per varis valors d'x simultàniament, això és possible gràcies a la superposició dels estats. Com ja sabem aquesta superposició no es pot observar per tant, podem avaluar un funció f(x) per varis valors d'x però no en podem observar el resultat, així doncs, hem de poder extreure la informació que volem d'aquella funció sense necessitat d'observar els resultats.

El circuit general que defineix l'algoritme és el següent:

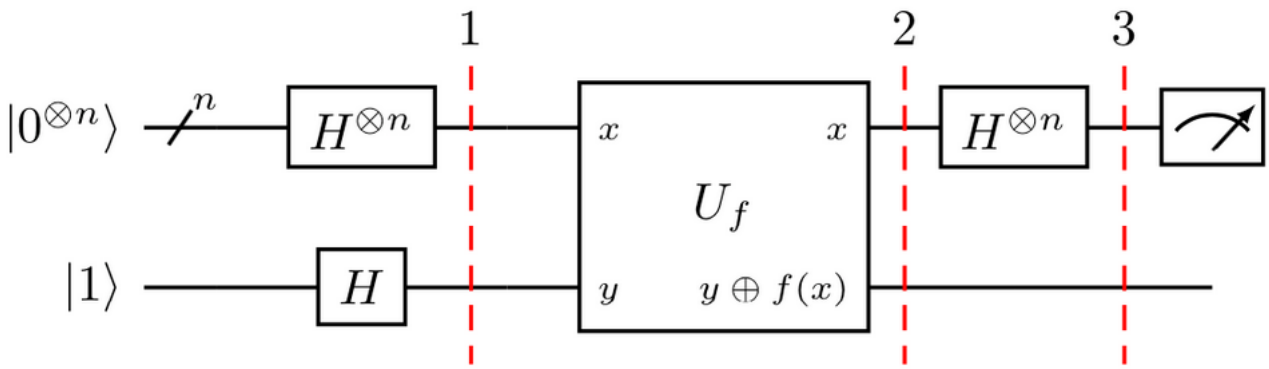


Figura 6.16: Circuit representatiu de l'algoritme de Deutsch-Jozsa

Anem pas per pas, inicialment tindrem n qbits inicialitzats a 0, que seran els que enviarà l'Alice, i un qbit inicialitzat a 1 que serà com el de control per aplicar la funció d'en Bob. El primer que farem serà posar tots els qbits en superposició aplicant la porta Hadamard a tots ells, tant al de l'Alice com al del Bob. La porta Uf és una "caixa negre" on hi tenim la funció d'en Bob i l'apliquem en els qbits de l'Alice. Finalment tornem a aplicar la porta Hadamard en els bits de l'Alice, cosa que farà que s'anul·li l'anterior, ja que $H^{\dagger} \cdot H = I$, i mesurem els qbits de l'Alice. Si el resultat de la mesura es $|\,0 \ldots 0\rangle$ significarà que la funció és constant, en canvi si el resultat és $|\,1 \ldots 1\rangle$ significarà que és balancejada.

Apliquem l'algoritme amb Qiskit, amb n=3 i una funció balancejada a la nostre "caixa negre". Per crear una funció balancejada, tan sols hem d'aplicar una CNOT al qbit d'en Bob per cada qbit de l'Alice, on els qbits de l'Alice seran els de control i el d'en Bob l'objectiu. Com que la inicialització de tots els qbits del circuit és a l'estat $|\,0\rangle$ aplicarem portes X per variar els inputs i les tornarem a aplicar després de les portes CNOT per tornar a l'estat inicial. A la figura 6.17 podem veure com quedaria el circuit de la nostre funció balancejada.

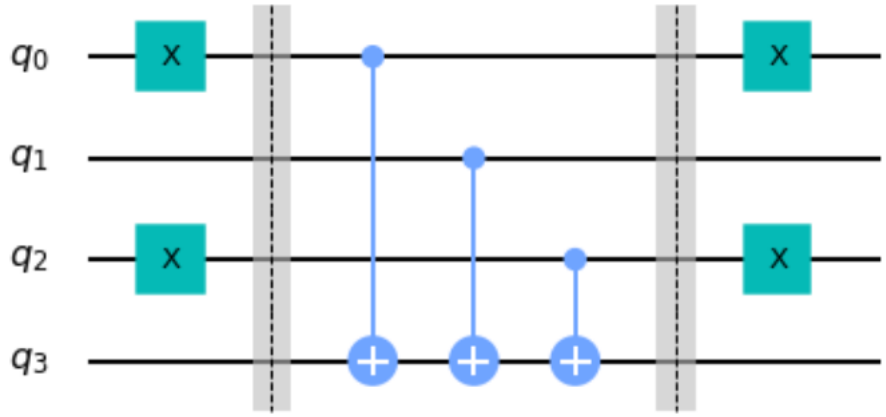


Figura 6.17: Circuit quantic en Qiskit

El circuit de la figura 6.17 serà la nostre Uf, per tant ara creem el circuit del algoritme que hem vist a la imatge 6.16

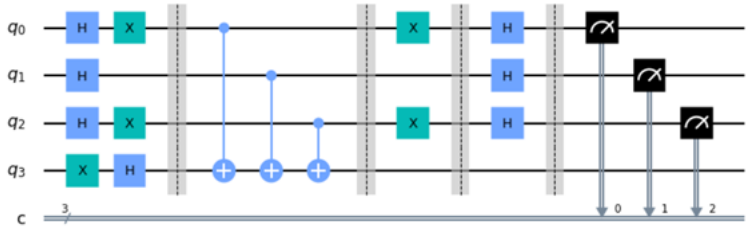


Figura 6.18: Circuit quantic en Qiskit

Com podem veure a la figura 6.18 , és com ens quedaria l'algoritme aplicar en Qiskit, així que l'exacutarem en el simulador local per veure el resultat.

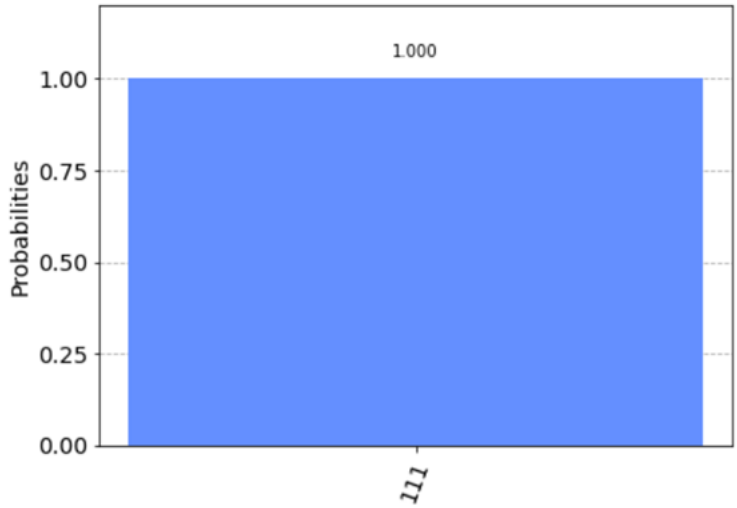


Figura 6.19: Gràfic de resultat circuit quàntic

Podem observar a la figura que ens ha donat el resultat esperat al haver-li aplicat una funció balancejada. Si en lloc de simular-lo amb el simulador l'enviem a un ordinador quàntic el resultat que es dona es el que podem observar a al figura

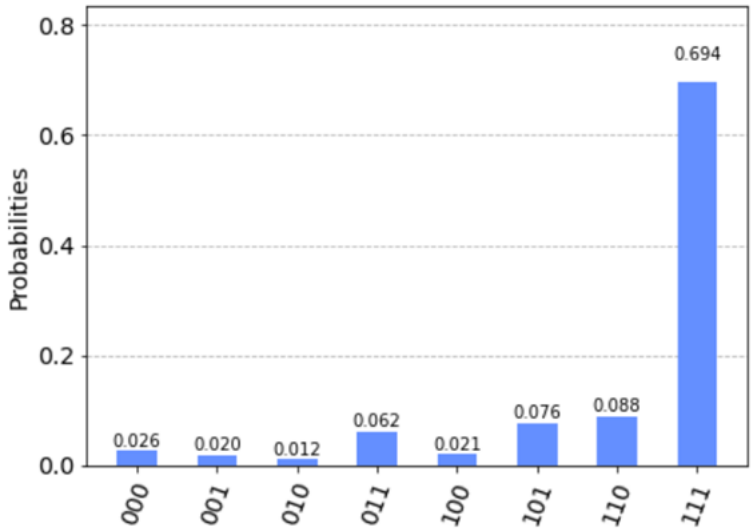


Figura 6.20: Gràfic de resultat circuit quàntic

En aquest cas no ens dona un 100% de probabilitats de l'estat 111 però això ho causa el soroll dels ordinadors quàntics, però el resultat majoritari segueix sent l'111.

Capítol 7

Conclusions

Els objectius d'aquest treball eren entendre la computació quàntica i veure en què es diferencia de la computació clàssica, hem vist que tenen una base semblant però que treballen de forma molt diferent. La computació quàntica suposa una gran millora respecte la clàssica en alguns aspectes, però encara és una tecnologia molt nova com per superar als ordinadors clàssics. A més tot i que la computació quàntica tingui millors resultats que la clàssica en alguns camps, per les tasques més bàsiques per les que utilitzem un ordinador seria molt pitjor.

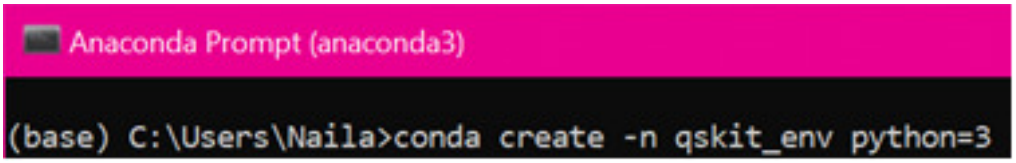
Tot i que hem arribat a entendre a grans trets en què consisteix la computació quàntica i a com programar un ordinador quàntic, ens hem adonat que els coneixements matemàtics i físics que demana per poder entendre la a fons son molt complexes ,i que per poder contribuir en l'avanç d'aquesta computació fa falta una base solida d'aquests coneixements.

Apèndix A

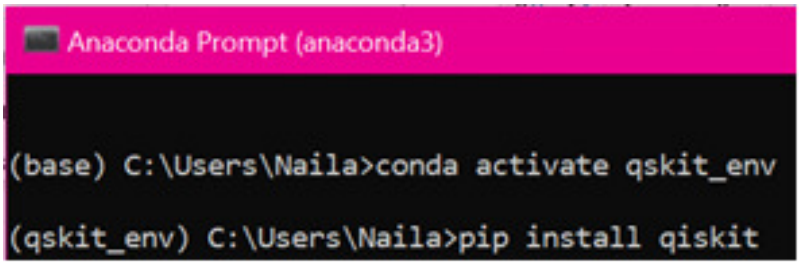
Insal·lació Qiskit

La instal·lació és senzilla, cal tenir Python 3.5 o major i et recomanen tenir Anaconda, així que instal·lem anaconda i seguim els passos de la documentació:

1. Creem un entorn de Python amb la consola d'Anaconda.



2. Entrem al nostre nou entorn i instal·lem el paquet de qiskit que conté tots els elements, Terra, Aer, Ignis, i Aqua.



3. Creem un nou notebook i importem qiskit.

Bibliografia

[1] Carles Franquesa i Niubò, *algorísmia comentada*, (2010).

[2] Michael A. Nielsen and Issac L L. Chuang, *Quantum computation and quantum information*, (2010).

[3] Noson S. Yanofsky and Mirco A. Mannucci, *Quantum computing for computer scientists*, (2008).