# Algoritmes quàntics de factorització

Alberto Cámara
alberto.camara.math@gmail.com
Tutor: Artur Travesa
Universitat de Barcelona
Facultat de Matemàtiques

10 de setembre de 2008

# Introducció

La computació quàntica és l'estudi de tècniques de processament d'informació mitjançant fenòmens propis de la mecànica quàntica. Aquesta ciència s'ha desenvolupat a partir de la dècada de 1970 en un intent de resoldre problemes propis de la física, com per exemple saber si és possible utilitzar efectes quàntics per transmetre senyals a una velocitat superior a la de la llum.

L'any 1981, Richard Feynman [3] va observar que simular l'evolució d'un sistema quàntic en un computador clàssic de manera eficient sembla impossible en general, i va proposar un model bàsic de computador quàntic mitjançant el qual es podrien implementar aquest tipus de simulacions. Aquest fet va atraure molta atenció vers la computació quàntica.

La computació quàntica s'ha desenvolupat notablement en els últims anys, tot i que es tracta d'una ciència que encara es troba en els seus inicis.

D'una banda, han estat construïts computadors quàntics senzills en laboratoris, i es treballa per aconseguir implementar computadors quàntics més complexos. Simultàniament, s'intenta resoldre problemes que amenacen la possibilitat de tractar eficientment la informació en un computador quàntic, com per exemple el fenomen de decoherència.

D'altra banda, s'han desenvolupat un model abstracte de computació quàntica i una teoria de computació que proposa algoritmes i mètodes per a ésser implementats sobre aquest model de computació.

Una de les fites més remarcables aconseguides dins d'aquesta teoria abstracta de computació és l'algoritme de Shor. L'any 1994, Peter Shor [9] va descobrir un algoritme que permet a un computador quàntic resoldre en temps polinòmic dos problemes molt importants: la factorització d'un nombre enter i el logaritme discret a $(\mathbb{Z}/p\mathbb{Z})^*$. Els millors algoritmes coneguts actualment per a resoldre aquests problemes sobre un computador clàssic tenen complexitat subexponencial.

La computació quàntica és actualment una ciència que atreu l'interès de científics de moltes disciplines. Pel que fa als matemàtics, el desenvolupament d'una teoria de complexitat computacional per als computadors

quàntics és d'una importància considerable, i tot just s'han fet els primers passos en aquesta direcció. D'altra banda, molts problemes d'interès matemàtic deriven del plantejament i de l'estudi dels algoritmes per a computadors quàntics.

L'estudi de l'algoritme de Shor és un d'aquests problemes d'interès matemàtic perquè es tracta d'un mètode per factoritzar nombres enters. En paraules de Gauss [4, Article 239]:

> *El problema de distingir nombres primers de compostos i descompondre aquests en els seus factors primers, que pertany als més importants i més útils de tota l'aritmètica, és tan conegut que seria superflu parlar abundantment d'això. [...] A part d'això, la dignitat de la ciència sembla requerir que es perfeccionin curosament tots els recursos per a la resolució d'un problema tan elegant i tan cèlebre. [...] Està fonamentat en la naturalesa del problema que qualssevol mètodes sortiran contínuament més prolixos com més grans són els nombres als quals s'apliquin.*

L'algoritme de Shor presenta fortes analogies amb el mètode $p-1$ de Pollard [2, 8.8]. A la vegada, el mètode de factorització de Lenstra [2, 10.3] pot ésser entès com una generalització del mètode de Pollard. És raonable preguntar-se si el mètode de Shor admet modificacions per tal de trobar nous algoritmes quàntics per a la factorització de nombres enters; la resposta a aquesta darrera pregunta és afirmativa.

En el primer capítol del text tractem el model matemàtic de la computació quàntica des d'un punt de vista axiomàtic: donem una definició de *bit quàntic*, de *porta lògica quàntica* i de computador quàntic, estudiem breument la manera d'implementar computacions clàssiques sobre un computador quàntic i exposem què entenem per complexitat computacional per a un algoritme quàntic.

En el segon capítol tractem tècniques per a la computació quàntica, principalment la transformada de Fourier quàntica i les seves aplicacions.

En el tercer capítol exposem l'algoritme de Shor (3.1.1), n'estudiem la complexitat i calculem la probabilitat que l'algoritme finalitzi amb èxit.

Finalment, en el quart i darrer capítol presentem el nostre resultat principal: un nou algoritme quàntic (4.2.3) per a la factorització de nombres enters lliures de quadrats mitjançant corbes el·líptiques. N'estudiem la complexitat algorítmica i calculem la probabilitat que l'algoritme finalitzi amb èxit. La complexitat asimptòtica d'aquest algoritme és la mateixa que la de l'algoritme de Shor. La probabilitat que l'algoritme finalitzi amb èxit no millora la de l'algoritme de Shor, però augmenta quan el nombre enter a factoritzar no té divisors primers de mida petita.

# Índex

# Capítol 1

# El model matemàtic de la computació quàntica

La intenció d'aquest capítol és descriure un model matemàtic de computació quàntica per tal de poder tractar algoritmes dissenyats per a computadors quàntics. També compararem en molts punts aquest model de computació amb el model de computació digital.

Un computador consta d'una sèrie de registres que contenen informació i d'una sèrie de dispositius que actuen sobre aquests registres a fi de modificar la informació que contenen, les anomenades *portes lògiques*. Per a usar el computador, cal codificar unes dades d'entrada als seus registres, aplicar les diferents portes lògiques de què consti el computador i finalment llegir les dades de sortida per tal d'obtenir el resultat del còmput.

En el model de computació que usen els computadors digitals, els registres que contenen informació s'anomenen *bits* i admeten dos estats, que codifiquem 0 i 1. Les portes lògiques es poden concebre com *interruptors* que modifiquen els valors de determinats bits del computador a partir d'unes determinades regles. Això és suficient per a codificar la nostra informació, habitualment en forma de nombres enters escrits en base 2. En aplicar les diferents portes lògiques arribarem a un estat de sortida: una sèrie de registres que llegirem habitualment en forma de nombres enters escrits en base 2, que contindran el resultat del còmput.

Un computador quàntic no és gaire diferent: ens caldrà definir com seran els seus registres, que anomenarem *qubits*, i caldrà definir una manera de codificar informació en ells. També caldrà definir les seves portes lògiques, anomenades *portes quàntiques*, que actuaran sobre aquests registres. Finalment, també necessitarem una manera de llegir les dades de sortida.

## 1.1 Bits i qubits

A l'hora de definir axiomàticament el model de la computació quàntica tindrem en compte moltes consideracions motivades per principis fonamentals de la mecànica quàntica.

El primer pas és definir els registres sobre els quals codificarem la informació que usarà el computador.

Fixem una $\mathbb{C}$-base ortonormal de $\mathbb{C}^2$, que denotarem $|0\rangle$, $|1\rangle$. Usem aquesta notació, anomenada *notació bra-ket de Dirac*, perquè és la manera habitual de denotar els estats de la mecànica quàntica. Per a més detalls, consulteu [7, 1.2].

**1.1.1 Definició.** Un qubit, o bit quàntic, és un vector unitari de $\mathbb{C}^2$. Així, un qubit el representarem com

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \qquad \alpha, \beta \in \mathbb{C},$$

amb $|\alpha|^2 + |\beta|^2 = 1$. Els dos estats $|0\rangle$ i $|1\rangle$ s'anomenen *estats bàsics*. Direm que $|\psi\rangle$ és un *estat quàntic* que és superposició dels estats bàsics.

Un principi fonamental de la mecànica quàntica ens indica que l'espai d'estats quàntics de dos qubits és el producte tensorial dels espais de cada qubit. Per això, l'estat quàntic de dos qubits vindrà donat per un element unitari de l'espai

$$\mathbb{C}^2 \otimes \mathbb{C}^2 \cong \mathbb{C}^4.$$

En aquest sentit disposem d'una base ortonormal de $\mathbb{C}^2 \otimes \mathbb{C}^2$ definida a partir de la base que hem fixat de $\mathbb{C}^2$, donada pels quatre vectors:

$$\begin{aligned} |00\rangle &:= |0\rangle \otimes |0\rangle, \\ |01\rangle &:= |0\rangle \otimes |1\rangle, \\ |10\rangle &:= |1\rangle \otimes |0\rangle, \\ |11\rangle &:= |1\rangle \otimes |1\rangle. \end{aligned}$$

Per tant, un estat quàntic de dos qubits serà un element

$$\alpha_1|00\rangle + \alpha_2|01\rangle + \alpha_3|10\rangle + \alpha_4|11\rangle, \qquad \text{amb } \sum_{i=1}^{4} |\alpha_i|^2 = 1.$$

**1.1.2 Observació.** El producte tensorial de dos estats no serà un vector unitari de $\mathbb{C}^4$ en general: caldrà normalitzar-lo després de prendre el producte tensorial.

De manera similar, l'espai dels estats quàntics de $n$ qubits serà $\mathbb{C}^{2^n}$. Escriurem els vectors de la base d'aquest espai parametritzant-los com a símbols *bra-ket* de cadenes binàries de longitud $n$:

$$|b_1 b_2 \dots b_n\rangle := |b_1\rangle \otimes |b_2\rangle \otimes \cdots \otimes |b_n\rangle,$$

amb $b_i \in \{0,1\}$ per a tot $i = 1, \dots n$.

**1.1.3 Definició.** Direm que els estats que corresponen als vectors de la base que hem triat són els estats bàsics del sistema de $n$ qubits. També codificarem els sistemes de $n$ qubits prenent únicament vectors unitaris.

La nostra manera de codificar la informació d'entrada en el computador quàntic serà molt similar a la manera que s'usa en la computació digital.

**1.1.4 Definició.** Associarem a cada nombre enter escrit en base 2

$$b = b_0 2^0 + b_1 2^1 + \ldots + b_{l-2} 2^{l-2} + b_{l-1} 2^{l-1}, \qquad b_i \in \{0,1\} \text{ per a } 0 \leq i \leq l-1$$

l'estat bàsic de $l$ qubits

$$|b_{l-1} b_{l-2} \dots b_1 b_0\rangle \in \mathbb{C}^{2^l}.$$

Denotarem

$$|b\rangle := |b_{l-1} b_{l-2} \dots b_1 b_0\rangle.$$

Observem que per codificar un enter que tingui $l$ xifres binàries caldrà fer servir un sistema de $l$ qubits. Això no representa un guany respecte de la computació digital pel que fa a la despesa d'espai. No obstant, observem que la dimensió de l'espai que conté un sistema de $l$ qubits és exponencial en $l$.

Pel que fa a la lectura d'informació i de resultats, hi haurà una diferència important amb el cas dels computadors digitals. Podem determinar si un bit digital és 0 o bé 1, però del principi d'incertesa de Heisenberg es dedueix que no és possible conèixer amb precisió l'estat d'un qubit.

**1.1.5 Definició.** *Observar* un qubit

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle; \quad \alpha, \beta \in \mathbb{C}, \quad |\alpha|^2 + |\beta|^2 = 1$$

és aplicar-li una transformació que li assigna $|0\rangle$, amb probabilitat $|\alpha|^2$, o $|1\rangle$, amb probabilitat $|\beta|^2$, amb la propietat que després d'observar $|\psi\rangle$ i d'haver obtingut $|0\rangle$ o $|1\rangle$ no és possible recuperar l'estat $|\psi\rangle$. Quan observem un qubit diem que hem efectuat una *mesura de la base computacional*.

En altres paraules, l'observació d'un qubit es duu a terme projectant-lo sobre els estats bàsics, amb unes certes probabilitats determinades pels productes escalars del qubit amb cada estat bàsic.

No és el mateix l'estat real d'un qubit i l'estat que n'observarem i, per tant, la probabilitat és inherent al procés de mesurar bits quàntics. En conseqüència, la probabilitat necessàriament ha de jugar un paper important en el nostre model computacional.

**1.1.6 Definició.** De manera anàloga al que succeeix amb un sistema d'un sol qubit, quan tinguem un sistema de $n$ qubits i vulguem observar un estat

$$|w\rangle = \sum_{s=0}^{2^n-1} \alpha_s |s\rangle,$$

on $\alpha_s \in \mathbb{C}$, obtindrem com a resultat de la nostra observació l'estat $|s\rangle$ amb probabilitat $|\alpha_s|^2$, i no serà possible recuperar $|w\rangle$.

## 1.2 Portes quàntiques i circuits quàntics

Un cop fixada la condició que els qubits són vectors unitaris de $\mathbb{C}^2$, les úniques transformacions que admet un estat quàntic són les transformacions unitàries, les transformacions lineals de l'espai que conserven la propietat d'ésser vector unitari. Aquestes transformacions són precisament les que ens interessen com a portes lògiques.

**1.2.1 Definició.** Una porta quàntica actuant sobre un estat de $n$ qubits és una transformació unitària

$$\begin{array}{rcl} U : \mathbb{C}^{2^n} & \longrightarrow & \mathbb{C}^{2^n} \\ |u\rangle & \longmapsto & U|u\rangle. \end{array}$$

Una porta quàntica $U$ és definida per una matriu unitària $A$. Recordem que les matrius unitàries són aquelles matrius $A$ tals que $A(\bar{A})^\top = (\bar{A})^\top A = \mathrm{Id}$. Per tant, una manera de construir una transformació unitària serà definir una aplicació lineal a partir de les imatges dels estats bàsics tal que la seva matriu associada en la base que hem triat sigui unitària.

**1.2.2 Exemple.** La porta quàntica NOT és una porta que actua sobre un qubit. La seva acció sobre els estats bàsics és:

$$\begin{array}{l} |0\rangle \longmapsto |1\rangle, \\ |1\rangle \longmapsto |0\rangle. \end{array}$$

El nom de la porta quàntica prové de l'evident analogia amb la porta lògica NOT que canvia el valor d'un bit digital. Segons la definició que hem donat, la matriu que li correspon a la porta quàntica NOT és

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

**1.2.3 Exemple.** La porta d'Hadamard és la porta quàntica $H$ que actua sobre un qubit i l'acció de la qual sobre els estats bàsics és la següent:

$$\begin{array}{rcl} |0\rangle & \longmapsto & \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \\ |1\rangle & \longmapsto & \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \end{array}$$

Per tant, la matriu que correspon a $H$ és la matriu

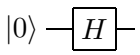
$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Un cop vista la definició de porta lògica quàntica, procedim a definir què és un circuit quàntic.

**1.2.4 Definició.** Un circuit quàntic consisteix en un conjunt finit de qubits i en un conjunt finit de portes quàntiques que actuen successivament sobre aquests qubits.

De manera molt semblant al que és habitual en el cas de la computació digital, sovint representarem un circuit quàntic mitjançant un conjunt de fils lògics, cadascun dels quals representarà un qubit. Les portes lògiques seran representades com caixes actuant sobre un subconjunt d'aquests fils. A l'esquerra del diagrama, hi posarem l'estat inicial dels qubits del circuit, i la lectura d'esquerra a dreta donarà indicació de l'ordre en el qual es van aplicant les successives transformacions de què consta el circuit.

**1.2.5 Exemple.** Prenem d'entrada un qubit inicialitzat en l'estat bàsic $|0\rangle$ i li apliquem la porta lògica d'Hadamard $H$ (1.2.3). El circuit quàntic que representa aquesta operació és el següent:

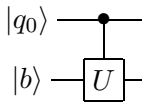



**1.2.6 Exemple.** Donada una transformació unitària $U$ que actua sobre $n$ qubits, definim la porta $U_c$ com la porta sobre $n+1$ qubits que sobre els estats bàsics efectua l'operació que segueix:

$$\begin{array}{rcll} |0\rangle \otimes |b\rangle & \longmapsto & |0\rangle \otimes |b\rangle, & \\ |1\rangle \otimes |b\rangle & \longmapsto & |1\rangle \otimes U|b\rangle, & 0 \leq b < 2^n. \end{array}$$

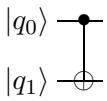
La porta $U_c$ s'anomena la porta $U$ controlada ja que, sobre els estats bàsics, l'estat del primer qubit determina si si s'aplica la porta $U$ sobre els $n$ qubits restants. És a dir, el primer qubit *controla* si s'aplica la transformació original sobre els $n$-qubits restants.

A nivell de circuits quàntics representarem la porta $U_c$ de la manera següent:



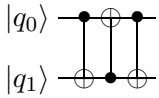

on $|q_0\rangle$ és el qubit que fa el control i $|b\rangle$ és l'estat dels $n$ qubits restants.

**1.2.7 Exemple.** Un cas particular de l'exemple que acabem de veure és l'operació NOT controlada, que sempre denotarem CNOT. A nivell de circuits, usarem el símbol

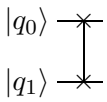



on el qubit $|q_0\rangle$ fa de control i la porta NOT actua sobre $|q_1\rangle$.

**1.2.8 Exemple.** Volem dissenyar una operació que ens permeti intercanviar els estats de dos qubits diferents. Aquesta operació, anomenada SWAP, s'utilitza freqüentment. Donat un estat de dos qubits $|q_0q_1\rangle$, es tracta del circuit:




Per simplificar, denotarem l'operació SWAP per




**1.2.9 Observació.** Com que les transformacions unitàries són sempre invertibles, qualsevol computació que fem mitjançant un circuit quàntic haurà de ser forçosament reversible.

**1.2.10 Definició.** Direm que un conjunt de portes lògiques (quàntiques) és universal si qualsevol circuit pot ésser modelat mitjançant únicament portes d'aquest conjunt.

En la computació digital és conegut que les portes AND ($\wedge$) i NOT ($\neg$) formen un conjunt universal de portes lògiques. En el cas quàntic, la porta AND no pot ser utilitzada perquè no és reversible i, en conseqüència, no dóna lloc a una transformació unitària.

Hi ha més característiques dels circuits digitals que tampoc podrem usar en el cas quàntic. En primer lloc no disposem d'una operació que copiï l'estat d'un qubit sobre un altre qubit. Tenir la possibilitat de fer això violaria un principi de la mecànica quàntica, que diu que els estats quàntics no es poden clonar. De manera similar, no disposarem en els nostres circuits d'operacions que ens permetin *bifurcar* un cable o *reunir* dos cables en un de sol. En segon lloc, als computadors clàssicament se'ls permet que els seus circuits es retroalimentin usant informació d'una part del circuit en una altra part, en allò que anomenem un *bucle*. Als circuits quàntics no els permetrem retroalimentar-se. En altres paraules, els circuits quàntics han de ser *acíclics*.

Malgrat aquestes limitacions, podem aconseguir un conjunt universal de portes quàntiques si considerem el conjunt de totes les portes que actuen sobre un únic qubit i hi afegim la porta CNOT, que actua sobre dos qubits (vegeu [5]).

A més, de cara a simular de manera efectiva la computació digital sobre un computador quàntic ens caldrà considerar una porta especial que actua sobre tres qubits. Es tracta de la porta de Toffoli. L'operació que efectua aquesta porta és un CNOT controlat. La seva acció sobre els estats bàsics es la següent: es nega el tercer qubit si i només si els dos primers qubits són $|1\rangle$. La matriu que li correspon a tal porta és

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

La utilitat de la porta de Toffoli quedarà palesa a la secció propera.

## 1.3 Computació reversible

Ja hem vist que prendre les transformacions unitàries com a conjunt de portes lògiques per al nostre model computacional força que tota computació feta d'acord amb aquest model sigui reversible. Això planteja un problema

a l'hora de simular la computació digital sobre aquest model. Hi ha molts circuits digitals que no són reversibles: n'hi ha prou si pensem que la porta AND no és reversible. Conèixer l'estat de sortida d'aquestes portes no permet saber amb certesa quin era l'estat inicial dels bits sobre els quals ha actuat una d'aquestes portes.

Si volem poder implementar un circuit clàssic sobre un computador quàntic, aquest circuit haurà de ser reversible. És desitjable tenir un mètode per modificar un circuit clàssic, a priori no reversible, per tal de fer-lo reversible i poder-lo implementar en un computador quàntic. Això és possible: donat un circuit clàssic no reversible hi ha una manera de construir un circuit reversible que realitzi el mateix còmput. També veurem que haurem de pagar un cert preu (acceptable) a l'hora de canviar el circuit original per aquest nou circuit reversible.

**1.3.1 Proposició.** *Sigui $x$ una informació d'entrada per a un computador quàntic i suposem que volem realitzar un còmput sobre $x$ per obtenir $F(x)$. Suposem que $F$ és bijectiva i que podem computar $F$ i $F^{-1}$. Aleshores podem modificar el còmput de $F(x)$ per tal de fer-lo reversible amb un augment constant en el nombre de portes lògiques aplicades i en el nombre de qubits requerit.*

Demostració: El mètode que explicarem és degut a Bennett [1] i ens limitarem a donar-ne alguns detalls.

Suposem que emmagatzemem les dades d'entrada $x$ en un registre del nostre computador quàntic que anomenem INPUT. Volem realitzar un càlcul sobre aquestes dades per obtenir $F(x)$ i codificar aquestes dades en un registre que anomenem OUTPUT.

La idea de Bennett consisteix en què si conservem certa informació al llarg de l'execució del computador, llavors el còmput de $F(x)$ serà reversible. Podem usar portes de Toffoli per simular portes AND, OR o NOT clàssiques, mitjançant una certa quantitat de qubits addicionals. L'únic problema d'implementació que aquest procediment representa prové del fet que si els qubits addicionals no són zero a l'hora d'efectuar els càlculs, llavors afectaran els resultats que observarem. Per tant, cal un mètode per posar aquests qubits addicionals a zero: aquest problema també és resolt satisfactòriament per Bennett.

El procediment a seguir és el següent:

1. Afegim qubits al nostre computador i els inicialitzem en l'estat $|0\rangle$.

2. Calculem $F(x)$. Utilitzem portes de Toffoli per simular les portes no reversibles. Els qubits de sortida que codifiquen $F(x)$ els anomenem

OUTPUT, mentre que els qubits addicionals de sortida els anomenem RECORD($F$).

3. Copiem OUTPUT a un registre prèviament inicialitzat a zero.

4. Desfem els càlculs aplicant les inverses de les portes lògiques dels passos 1 i 2. Obtenim que d'aquesta manera eliminem el primer registre OUTPUT i el registre RECORD($F$).

5. Calculem a partir de $F(x)$ les dades $(F(x), F^{-1}F(x)) = (F(x), x)$ mitjançant tots els passos previs aplicats a la funció $F^{-1}$ i prenent com a dades d'entrada $F(x)$.

6. Repetim en ordre invers tot el procés que hem explicat en els passos 1, 2 i 3 amb la funció $F^{-1}$ prenent com a dades d'entrada $F(x)$. Passem de tenir $(x, F(x))$ a tenir només $F(x)$.

Per aclarir una mica la situació, ho resumim tot en la taula que segueix. Els passos 1, 2, 3, 4 i 5 corresponen a les cinc primeres files de la taula respectivament, mentre que el pas 6 correspon a les dues últimes files d'aquesta.

| | | | |
|---|---|---|---|
| INPUT | 0 | 0 | 0 |
| INPUT | OUTPUT | RECORD($F$) | 0 |
| INPUT | OUTPUT | RECORD($F$) | OUTPUT |
| INPUT | 0 | 0 | OUTPUT |
| INPUT | INPUT | RECORD($F^{-1}$) | OUTPUT |
| 0 | INPUT | RECORD($F^{-1}$) | OUTPUT |
| 0 | 0 | 0 | OUTPUT |

Aquest mètode permet implementar el còmput de $F$ d'una manera reversible. El nombre de qubits addicionals que cal utilitzar varia en funció del nombre portes de Toffoli que siguin necessàries per implementar el circuit (una per a cada porta AND), i això pot multiplicar per una constant el nombre de qubits requerits originalment i el nombre de portes lògiques que cal aplicar. □

## 1.4 Complexitat computacional quàntica

Estudiarem la complexitat d'un algoritme en funció dels recursos que calgui utilitzar per tal d'aconseguir la seva implementació.

Els recursos que utilitzen els computadors digitals són l'espai, mesurat amb el nombre de bits necessaris per tal d'implementar l'algoritme, i el

temps, mesurat amb el nombre d'operacions bàsiques o de portes lògiques que cal aplicar a les dades d'entrada per tal d'obtenir el resultat.

Pel que fa a la computació quàntica, els recursos necessaris són l'espai que requereix la implementació d'un algoritme, mesurat amb qubits, i també mitjançant el temps que requereix la seva execució, mesurat amb el nombre d'operacions bàsiques que cal fer (o portes lògiques quàntiques que cal aplicar) per tal de passar de les dades d'entrada al resultat.

A més, suposarem que tenim la capacitat de fer les operacions següents:

1. Construir un espai d'estats adequat al computador quàntic que volem realitzar.
2. Implementar una família universal de portes quàntiques.
3. Aplicar portes quàntiques a qualsevol subconjunt de qubits del computador.
4. Preparar qualsevol estat bàsic $|x_1 \dots x_n\rangle$ aplicant un màxim de $n$ portes quàntiques.
5. Mesurar la base computacional de qualsevol subconjunt de qubits del computador.

Suposarem que mesurar la base computacional és una operació que té cost lineal en el nombre de qubits a mesurar.

# Capítol 2

# Tècniques per a la computació quàntica

## 2.1 Transformada de Fourier quàntica

La possibilitat d'efectuar transformades de Fourier discretes en temps polinòmic representa un guany important per a la computació quàntica respecte de la computació digital.

**2.1.1 Notació.** Definim

$$\exp(x) := e^{2\pi i x}.$$

Fixem un nombre enter $N$. Recordem que la transformada de Fourier és una transformació lineal unitària de $\mathbb{C}^N$ que actua aplicant un vector de nombres complexos $x_0, \ldots, x_{N-1}$ en un altre vector de nombres complexos $y_0, \ldots, y_{N-1}$ definits per

$$y_j := \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp\left(j\frac{k}{N}\right) x_k, \qquad 0 \leq j \leq N-1.$$

Amb les notacions que hem fixat per a la computació quàntica, aplicarem aquesta transformació a una base ortonormal

$$|0\rangle, |1\rangle, \ldots, |N-1\rangle$$

mitjançant l'assignació

$$|j\rangle \longmapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp\left(j\frac{k}{N}\right) |k\rangle.$$

**2.1.2 Definició.** Direm que la transformació definida per l'assignació anterior és una transformada de Fourier quàntica.

Una transformada de Fourier és una transformació unitària i, per tant, té cabuda dins de la computació quàntica. Veurem una manera efectiva d'implementar aquesta transformació usant portes d'un i de dos qubits, i un cop haurem construït un circuit quàntic que implementi una transformada de Fourier quàntica haurem demostrat implícitament que es tracta d'una transformació unitària.

Suposem ara que $N = 2^n$, $n \in \mathbb{N}$. Escriurem els nombres que denoten els estats bàsics en base 2. Per a cada $j$ tal que $0 \leq j \leq N-1$, escriurem

$$j = j_1 2^{n-1} + j_2 2^{n-2} + \cdots + j_{n-1} 2 + j_n, \qquad j_i \in \{0, 1\}.$$

**2.1.3 Notació.** Si $\varepsilon_i \in \{0, 1\}$ per a $i \in \{1, \ldots, n\}$, usarem la notació $0.\varepsilon_1\varepsilon_2 \cdots \varepsilon_n$ per referir-nos al nombre racional $\sum_{i=1}^{n} \varepsilon_i 2^{-i}$.

**2.1.4 Proposició.** *La transformada de Fourier quàntica admet l'expressió*

$$|j\rangle \mapsto \frac{1}{\sqrt{2^n}}(|0\rangle + \exp(0.j_n)|1\rangle) \otimes (|0\rangle + \exp(0.j_{n-1}j_n)|1\rangle) \otimes \cdots \\ \cdots \otimes (|0\rangle + \exp(0.j_1 \cdots j_n)|1\rangle).$$

Demostració: La imatge de $|j\rangle$ per la transformada de Fourier quàntica és

$$\frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} \exp\left(jk2^{-n}\right)|k\rangle = \\ \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \exp\left(j \sum_{l=1}^{n} k_l 2^{-l}\right)|k_1 \cdots k_n\rangle = \\ \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \bigotimes_{l=1}^{n} \exp\left(jk_l 2^{-l}\right)|k_l\rangle = \\ \frac{1}{2^{n/2}} \bigotimes_{l=1}^{n} \sum_{k_l=0}^{1} \exp\left(jk_l 2^{-l}\right)|k_l\rangle = \\ \frac{1}{2^{n/2}} \bigotimes_{l=1}^{n} \left(|0\rangle + \exp\left(j2^{-l}\right)|1\rangle\right).$$

□

A continuació expliquem com construir un circuit quàntic que implementi una transformada de Fourier quàntica. Considerem les portes quàntiques

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & \exp(2^{-k}) \end{bmatrix}$$

per a $k \in \{2, \ldots, n\}$, i considerem també la porta d'Hadamard

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

Apliquem la porta $H$ sobre el primer qubit de l'estat $|j_1 \cdots j_n\rangle$ i obtenim

$$\frac{1}{2^{1/2}}(|0\rangle + \exp(0.j_1)|1\rangle) \otimes |j_2 \cdots j_n\rangle, \tag{2.1}$$

ja que

$$\exp(0.j_1) = e^{2\pi i \frac{j_1}{2}} = \begin{cases} -1, & \text{si } j_1 = 1, \\ 1, & \text{si } j_1 = 0. \end{cases}$$

Apliquem a l'estat (2.1) la porta $R_2$ controlada amb el segon qubit. Obtenim l'estat

$$\frac{1}{2^{1/2}}(|0\rangle + \exp(0.j_1 j_2)|1\rangle) \otimes |j_2 \cdots j_n\rangle. \tag{2.2}$$

Apliquem successivament les portes $R_3, R_4, \ldots, R_n$ controlades pels qubits $|j_3\rangle, |j_4\rangle, \ldots, |j_n\rangle$, respectivament. Obtenim l'estat

$$\frac{1}{2^{1/2}}(|0\rangle + \exp(0.j_1 j_2 \cdots j_n)|1\rangle) \otimes |j_2 \cdots j_n\rangle. \tag{2.3}$$

Repetim aquest mateix procediment amb el segon qubit $|j_2\rangle$. Li apliquem primer una porta $H$ i posteriorment li apliquem les portes $R_3, R_4, \ldots, R_n$ controlades respectivament pels qubits $|j_3\rangle, |j_4\rangle, \ldots, |j_n\rangle$. Al final del procediment obtenim l'estat

$$\frac{1}{2^{2/2}}(|0\rangle + \exp(0.j_1 j_2 \cdots j_n)|1\rangle) \otimes (|0\rangle + \exp(0.j_2 j_3 \cdots j_n)|1\rangle) \otimes |j_3 \cdots j_n\rangle. \tag{2.4}$$

Si apliquem de manera successiva una porta $H$ sobre el qubit $|j_k\rangle$, seguida de les portes $R_{k+1}, R_{k+2}, \ldots, R_n$ controlades respectivament pels qubits $|j_{k+1}\rangle, |j_{k+2}\rangle, \ldots, |j_n\rangle$, acabem en l'estat

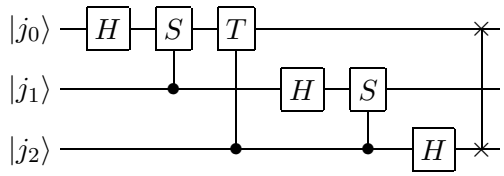
$$\frac{1}{2^{n/2}}(|0\rangle + \exp(0.j_1 j_2 \cdots j_n)|1\rangle) \otimes \cdots \otimes (|0\rangle + \exp(0.j_n)|1\rangle). \tag{2.5}$$

Si efectuem operacions SWAP per intercanviar les posicions dels qubits, aquest estat es transforma en l'estat

$$\frac{1}{2^{n/2}}(|0\rangle + \exp(0.j_n)|1\rangle) \otimes \cdots \otimes (|0\rangle + \exp(0.j_1 j_2 \cdots j_n)|1\rangle), \tag{2.6}$$

que és la imatge de $|j\rangle$ per la transformada de Fourier quàntica.

**2.1.5 Exemple.** Per ajudar a visualitzar aquesta construcció, explicitarem un circuit que realitza la transformada de Fourier quàntica sobre tres qubits. Es tracta del circuit següent:




on

$$S := R_2 = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \qquad T := R_3 = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}.$$

Acabem aquesta secció amb un recompte del nombre de portes lògiques usades per implementar la transformada de Fourier quàntica.

En el pas $j$ hem usat una porta $H$ i un total de $n-j$ portes del tipus $R_k$, sumant així un total de $n-j+1$ portes. Tenint en compte que hem fet un total de $n$ passos, la xifra s'eleva fins a la quantitat de

$$\sum_{j=1}^{n}(n-j+1) = \sum_{j=1}^{n} j = \frac{n(n+1)}{2}$$

portes lògiques. A més, cal fer com a molt $n/2$ operacions SWAP, cadascuna de les quals s'efectua aplicant tres portes CNOT. Per tant, l'ordre de l'algoritme que implementa la transformada de Fourier quàntica és $O(n^2)$.

## 2.2 Estimació de fase

Siguin $U$ un operador unitari i $|u\rangle$ un vector propi de $U$ de valor propi $\exp(\varphi) = e^{2\pi i\varphi}$, amb $\varphi \in \mathbb{R}$, $0 \leq \varphi < 1$. Suposem que desconeixem el valor de $\varphi$. El nostre objectiu en aquesta secció consisteix a donar un algoritme per estimar el valor de $\varphi$.

Per tal d'aconseguir aquest objectiu, assumirem que hem construït l'estat $|u\rangle$ i que tenim unes *caixes negres* que efectuen les operacions $U_c^{2^j}$, això és, l'operació $U^{2^j}$ controlada per un qubit auxiliar, per a uns certs exponents $j \in \mathbb{N}$ fitats per la precisió amb la qual vulguem estimar $\varphi$. Més endavant explicarem com implementar aquestes *caixes negres* per als problemes concrets que tractarem. Al final d'aquesta secció explicarem detalladament la relació entre els exponents $j$ i la precisió en l'aproximació de $\varphi$.

Usarem dos registres. En primer lloc, un registre de $t$ qubits inicialitzats a l'estat $|0\rangle$. Triarem $t$ d'acord amb la quantitat de xifres en base 2 de

precisió que vulguem en l'estimació de $\varphi$ i segons la probabilitat amb la qual vulguem que el procediment tingui èxit.

En segon lloc, codifiquem l'estat $|u\rangle$ en tants qubits com faci falta.

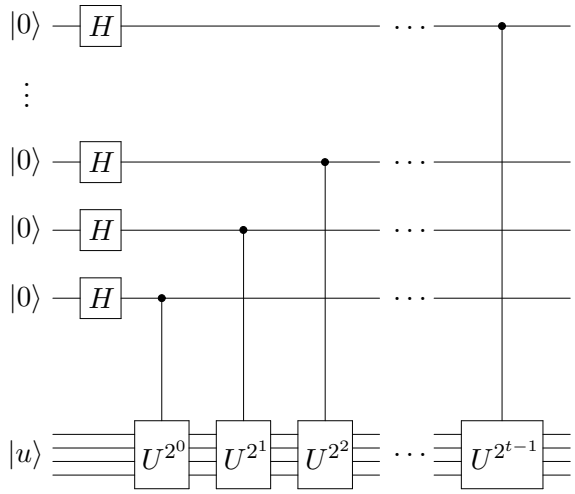
A continuació apliquem la porta d'Hadamard

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

a cada qubit del primer registre. Així, cadascun dels $t$ qubits d'aquest registre és sotmès a la transformació

$$|0\rangle \longmapsto \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle).$$

Tot seguit apliquem al segon registre les portes $U^{2^0}, U^{2^1}, \ldots, U^{2^{t-1}}$ successivament, controlades pels $t$ qubits del primer registre, de la manera següent: la porta $U^{2^0}$ és controlada pel $t$-èsim qubit del primer registre, la porta $U^{2^1}$ és controlada pel $(t-1)$-èsim qubit del primer registre, i així successivament fins arribar a la porta $U^{2^{t-1}}$, que és controlada pel primer qubit del primer registre. Aquest procediment queda explicat en la figura següent:




Els registres del computador finalitzen aquest procediment en l'estat

$$\frac{1}{2^{t/2}} \bigotimes_{j=1}^{t} (|0\rangle + \exp(2^{t-j}\varphi)|1\rangle) \otimes |u\rangle, \tag{2.7}$$

que és equivalent a l'estat

$$\frac{1}{2^{t/2}} \sum_{k=0}^{2^t-1} \exp(k\varphi)|k\rangle.$$

El motiu pel qual el computador finalitza en l'estat (2.7) és que l'operació que efectua la porta $U_c^{2^j}$ sobre el qubit de control i sobre el registre $|u\rangle$ és

$$\begin{array}{rcl} |0\rangle \otimes |u\rangle & \longmapsto & |0\rangle \otimes |u\rangle, \\ |1\rangle \otimes |u\rangle & \longmapsto & |1\rangle \otimes \exp(2^j\varphi)|u\rangle = \exp(2^j\varphi)|1\rangle \otimes |u\rangle. \end{array}$$

El pas següent consisteix a aplicar la inversa d'una transformada de Fourier quàntica. Això es pot fer per inversió del circuit construït a la secció anterior per a la transformada de Fourier quàntica, és a dir: mitjançant l'aplicació de les mateixes portes quàntiques en l'ordre invers.

Per tal d'explicar per què aquest procediment ens pot ajudar a trobar una bona aproximació per a la fase $\varphi$, suposem en primer lloc que $\varphi$ pot ésser expressat de manera exacta mitjançant $t$ xifres binàries, com a $\varphi = 0.\varphi_1 \dots \varphi_t$, amb $\varphi_i \in \{0, 1\}$. Aleshores el primer registre de l'estat (2.7) es pot reescriure en la forma

$$\begin{aligned} \frac{1}{2^{t/2}} \left(|0\rangle + \exp(0.\varphi_t)|1\rangle\right) \otimes \left(|0\rangle + \exp(0.\varphi_{t-1}\varphi_t)|1\rangle\right) \otimes \cdots \\ \cdots \otimes \left(|0\rangle + \exp(0.\varphi_1\varphi_2 \dots \varphi_t)|1\rangle\right). \end{aligned} \tag{2.8}$$

L'estat (2.7) és la imatge de l'estat $|\varphi_1 \dots \varphi_t\rangle$ per la transformada de Fourier quàntica. Per tant, si apliquem la inversa de la transformada de Fourier quàntica a (2.7) i mesurem la base computacional obtindrem $\varphi$.

Per descomptat, la inversa de la transformada de Fourier quàntica i una mesura de la base computacional ens proporcionen $\varphi$ en el cas que aquesta fase sigui expressable de manera exacta mitjançant $t$ xifres en base 2. Quan això no succeeixi, obtindrem una aproximació $\tilde{\varphi}$ de $t$ xifres en base 2 per a $\varphi$.

Explicitem l'algoritme quàntic per a l'estimació de fase.

**2.2.1 Algoritme.** Suposem que disposem d'una *caixa negra* que efectua l'operació $U_c^j$, per a $j$ natural, d'un vector propi $|u\rangle$ de $U$ de valor propi $\exp(\varphi)$ i $t = n + \lceil \log(2 + \frac{1}{2\varepsilon}) \rceil$ qubits inicialitzats a zero.

1. **Estat inicial:** $|0\rangle|u\rangle$.

2. **Superposició en el primer registre:** $\rightarrow \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle|u\rangle$.

3. **Apliquem la porta $U_c^j$ controlada pels bits del primer registre:**

$$\to \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle U^j |u\rangle = \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} \exp(j\varphi)|j\rangle|u\rangle.$$

4. **Apliquem la inversa de la transformada de Fourier quàntica:**

$$\to |\tilde{\varphi}\rangle|u\rangle.$$

5. **Mesurem el primer registre:** $\to \tilde{\varphi}$.

Obtenim com a sortida de l'algoritme una aproximació $\tilde{\varphi}$ de $\varphi$ de $n$ xifres binàries de precisió.

### 2.2.1 Complexitat de l'algoritme i probabilitat d'èxit

Ens proposem investigar què succeeix quan $\varphi$ no es pot escriure de manera exacta amb $t$ xifres binàries per tal de saber si la transformada de Fourier quàntica ens pot proporcionar una bona aproximació de $\varphi$ amb probabilitat alta.

Sigui $b$ l'enter en $\{0, \ldots, 2^t - 1\}$ tal que $b/2^t = 0.b_1 \ldots b_t$ és la millor aproximació de $t$ xifres binàries a $\varphi$ per defecte. Llavors,

$$0 \leq \delta := \varphi - \frac{b}{2^t} < 2^{-t}.$$

Després d'aplicar la inversa de la transformada de Fourier quàntica a l'estat (2.7) obtenim l'estat

$$\frac{1}{2^t} \sum_{k,l=0}^{2^t-1} \exp\left(-kl2^{-t}\right) \exp(\varphi k)|l\rangle.$$

Sigui $\alpha_l$ l'amplitud de $|b + l \pmod{2^t}\rangle$,

$$\alpha_l := \frac{1}{2^t} \sum_{k=0}^{2^t-1} \left(\exp\left(\varphi - (b+l)2^{-t}\right)\right)^k.$$

Aquesta suma correspon a una sèrie geomètrica i, per tant,

$$\alpha_l = \frac{1}{2^t} \left( \frac{1 - \exp(2^t\varphi - (b+l))}{1 - \exp\left(\varphi - (b+l)2^{-t}\right)} \right) = \frac{1}{2^t} \left( \frac{1 - \exp(2^t\delta - l)}{1 - \exp\left(\delta - l2^{-t}\right)} \right).$$

Suposem a continuació que en mesurar la base computacional obtenim com a resultat $m$. Ens proposem fitar la probabilitat d'obtenir un valor per

a $m$ tal que $|m-b|>e$, on $e$ és un enter que prenem com la tolerància d'error desitjada. Tal probabilitat és donada per

$$P(|m-b|>e)=\sum_{-2^{t-1}<l\leq-(e+1)}|\alpha_l|^2+\sum_{e+1\leq l\leq 2^t-1}|\alpha_l|^2. \qquad (2.9)$$

Per a qualsevol nombre real $\theta$, es té que $|1-\exp(\theta)|\leq 2$ i, per tant,

$$|\alpha_l|\leq\frac{2}{2^t\,|1-\exp\left(\delta-l2^{-t}\right)|}.$$

Es té que per a tot $\theta$ en l'interval $(-1/2,1/2)$ se satisfà que $|1-\exp(\theta)|\geq 4|\theta|$. En el nostre cas, quan $-2^{t-1}<l\leq 2^t$ tenim que $-1/2\leq(\delta-l2^{-t})\leq 1/2$. Per tant,

$$|\alpha_l|\leq\frac{1}{2^{t+1}(\delta-l2^{-t})}.$$

Si apliquem aquesta darrera fita a l'equació (2.9), obtenim que

$$P(|m-b|>e)\leq\frac{1}{4}\left[\sum_{l=-2^{t-1}+1}^{-(e+1)}\frac{1}{(l-2^t\delta)^2}+\sum_{l=e+1}^{2^{t-1}}\frac{1}{(l-2^t\delta)^2}\right].$$

Finalment, si usem el fet que $0\leq 2^t\delta\leq 1$, obtenim que

$$\begin{aligned}P(|m-b|>e)&\leq\frac{1}{4}\left[\sum_{l=-2^{t-1}+1}^{-(e+1)}\frac{1}{l^2}+\sum_{l=e+1}^{2^{t-1}}\frac{1}{(l-1)^2}\right]\\&\leq\frac{1}{2}\sum_{l=e}^{2^{t-1}-1}\frac{1}{l^2}\leq\frac{1}{2}\int_{e-1}^{2^{t-1}-1}\frac{dl}{l^2}=\frac{1}{2(e-1)}.\end{aligned}$$

Suposem que volem aproximar $\varphi$ amb una precisió de $2^{-n}$. En altres paraules, triem $e=2^{t-n}-1$. Si usem $t=n+p$ qubits en l'algoritme d'estimació de fase, obtenim que la probabilitat d'obtenir una aproximació correcta amb aquesta precisió és com a mínim

$$1-\frac{1}{2(2^p-2)}.$$

Resumim el resultat que hem obtingut en la proposició següent.

**2.2.2 Proposició.** *Amb les notacions anteriors, si volem obtenir una aproximació de $\varphi$ amb una precisió de $n$ xifres binàries amb probabilitat com a mínim igual a $1-\varepsilon$ cal prendre*

$$t=n+\left\lceil\log\left(2+\frac{1}{2\varepsilon}\right)\right\rceil.\ \square$$

Finalment, ens plantegem què fer en la situació en què no sabem construir individualment vectors propis per a l'operador $U$, però sabem construir un vector que n'és combinació lineal

$$|\psi\rangle = \sum_u c_u |u\rangle.$$

Suposem que cada estat $|u\rangle$ és vector propi de valor propi $\exp(\varphi_u)$. L'algoritme d'estimació de fase en aquest cas prendrà com a estat inicial l'estat $|0\rangle \sum_u c_u|u\rangle$ i ens retornarà l'estat

$$\sum_u c_u |\tilde{\varphi_u}\rangle |u\rangle.$$

És de comprovació immediata que si triem $t$ com a (2.2.2) aleshores la probabilitat de mesurar $\varphi_u$ amb una precisió de $n$ xifres binàries és com a mínim $|c_u|^2(1-\varepsilon)$.

## 2.3 Algoritme quàntic per a trobar l'ordre d'un element en un grup arbitrari

Siguin $(G,\cdot)$ un grup abelià finit i $e$ el seu element neutre. En general, calcular l'ordre d'un element $g \in G$ és un problema difícil. Ens proposem donar un algoritme quàntic per a resoldre aquest problema.

Suposarem que tenim una manera de codificar els elements $h \in G$ usant qubits. Per exemple, si $N = 2^n$ amb $n \geq 2$, els elements del grup $(\mathbb{Z}/N\mathbb{Z})^*$ es poden codificar en $n$ qubits mitjançant la representació dels enters $x \in \{0,\ldots,N-1\}$ en base 2.

Denotem per $|h\rangle$ la codificació de $h \in G$. Suposarem també que podem implementar la porta quàntica que efectua la transformació

$$|g\rangle|h\rangle \longmapsto |g\rangle|gh\rangle \tag{2.10}$$

per a $g,h \in G$ utilitzant una quantitat constant de portes quàntiques d'un o dos qubits. Direm que una transformació com 2.10 és una operació en $G$.

Sigui $r$ l'ordre d'un element $g \in G$. L'operador $U_g$ definit per

$$U_g|h\rangle := |gh\rangle$$

és un operador unitari. Això és degut a que l'existència d'invers per a $g$ dins $G$ fa que el circuit que representa aquesta operació sigui totalment reversible.

Per a $0 \leq s \leq r-1$, els estats

$$|u_s\rangle := \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left(\frac{-sk}{r}\right) |g^k\rangle$$

són vectors propis per a l'operador $U_g$, ja que

$$\begin{aligned} U_g|u_s\rangle &= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left(\frac{-sk}{r}\right) U|g^k\rangle \\ &= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left(\frac{-sk}{r}\right) |g^{k+1}\rangle \\ &= \exp\left(\frac{s}{r}\right) |u_s\rangle. \end{aligned}$$

Observem que si tinguéssim una manera de construir l'estat $|u_s\rangle$, aleshores l'algoritme d'estimació de fase ens permetria trobar una bona aproximació per a $s/r$. Un cop obtinguda aquesta aproximació, l'algoritme clàssic de fraccions continuades [10, VIII.4] ens permetria obtenir $r$. El problema és que poder construir els estats $|u_s\rangle$ pressuposa conèixer $r$.

Per tal d'evitar aquest inconvenient, disposem del resultat que segueix.

**2.3.1 Lema.** *Se satisfà que*

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = |e\rangle.$$

Demostració: Es té que

$$\begin{aligned} &\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle \\ &= \frac{1}{r} \sum_{s=0}^{r-1} \sum_{k=0}^{r-1} \exp\left(\frac{-sk}{r}\right) |g^k\rangle \\ &= \frac{1}{r} \sum_{k=0}^{r-1} \left( \sum_{s=0}^{r-1} \exp\left(\frac{-sk}{r}\right) \right) |g^k\rangle. \end{aligned}$$

Observem que se satisfà que

$$\sum_{s=0}^{r-1} \exp\left(\frac{-sk}{r}\right) = r\delta_{k0},$$

on $\delta_{ij}$ denota l'aplicació delta de Kronecker. D'aquesta manera,

$$\begin{aligned} &\frac{1}{r} \sum_{k=0}^{r-1} \left( \sum_{s=0}^{r-1} \exp\left(\frac{-sk}{r}\right) \right) |g^k\rangle \\ &= \frac{1}{r} \sum_{k=0}^{r-1} r\delta_{k0} |g^k\rangle \\ &= |e\rangle, \end{aligned}$$

com volíem veure. □

Aquest resultat ens indica que podem adaptar l'algoritme d'estimació de fase prenent com a estat inicial en el segon registre l'estat $|e\rangle$. Si usem $t = 2n + 1 + \lceil \log(2 + \frac{1}{2\varepsilon}) \rceil$ qubits per al primer registre, on $n$ és el nombre de qubits necessaris per a codificar un element de $G$, obtenim una estimació per a la fase $\varphi \simeq s/r$ amb precisió de $2n+1$ xifres binàries amb probabilitat com a mínim $(1 - \varepsilon)/r$.

Un altre requisit de cara a poder aplicar l'algoritme d'estimació de fase és el de poder implementar la *caixa negra* que efectua l'operació $U_c^{2^j}$. Per tal de construir la seqüència completa d'aquestes operacions usem un algoritme binari d'exponenciació (vegeu [10, I.6] per als detalls sobre aquest algoritme).

**2.3.2 Lema.** *Amb les notacions anteriors, és possible implementar la porta quàntica*

$$|k\rangle|h\rangle \longmapsto |k\rangle|g^k h\rangle$$

*mitjançant $O(n)$ operacions en $G$.*

Demostració: Volem computar, per a $k = k_0 2^0 + k_1 2^1 + \cdots + k_{t-1} 2^{t-1}$, la transformació

$$|k\rangle|h\rangle \longmapsto |k\rangle|g^k h\rangle = |k\rangle U_g^{k_{t-1} 2^{t-1}} \cdots U_g^{k_0 2^0} |h\rangle$$

La idea és computar reversiblement la funció $g^k$ com a funció de $k$ en un tercer registre, per multiplicar posteriorment els continguts del segon registre per $g^k$, esborrant finalment els continguts del tercer registre auxiliar usant el mètode de Bennett (vegeu secció 1.3).

Es calculen successivament els valors $g^{2^j}$ per a $0 \leq j \leq t-1$, realitzant un total de $t - 1 = O(n)$ operacions en $G$. Recordem que les operacions en $G$ són les transformacions definides per (2.10). Finalment, computem

$$g^k = (g^{k_{t-1} 2^{t-1}}) \cdots (g^{k_0 2^0}),$$

efectuant un màxim de $t-1$ operacions en $G$. Per tant, podem construir la porta quàntica requerida mitjançant $O(n)$ operacions en $G$. □

Un cop aplicat aquest algoritme per a trobar l'ordre d'un element en un grup, obtenim una estimació $s/r \simeq \varphi$. A continuació, podem aplicar l'algoritme de fraccions continuades al nombre racional $s/r$ per obtenir nombres enters $s'$ i $r'$ primers entre si tals que $s'/r' = s/r$. Aquest algoritme té complexitat $O(n^3)$ si $s$ i $r$ són nombres enters expressables en $n$ xifres en base 2.

El nombre $r'$ és el nostre candidat a esdevenir l'ordre que cerquem. Si $g^{r'} = e$, aleshores ja estem. Malauradament, en el cas en què $\mathrm{mcd}(s, r) \neq 1$ l'algoritme de fraccions continuades ens retorna un factor $r'$ de $r$. La manera més directa de resoldre aquesta situació consisteix a estudiar el fet que si triem $s$ a l'atzar en $\{0, \ldots, r-1\}$ és altament probable que $s$ i $r$ siguin coprimers. Per exemple, la quantitat de nombres primers per sota de $r$ és com a mínim

$$\frac{r}{2\log(r)}$$

si $r$ és prou gran i, per tant, la probabilitat que $s$ sigui primer és com a mínim $1/(2\log(r))$. Si repetim l'algoritme $2\log(r)$ vegades observarem una fase $s/r$ on $s$ i $r$ siguin coprimers amb probabilitat molt alta. D'aquesta manera podrem obtenir $r$ mitjançant l'algoritme de fraccions continuades.

Finalitzem aquesta secció explicitant l'algoritme que hem treballat.

**2.3.3 Algoritme.** Siguin $(G, \cdot)$ un grup i $e$ el seu element neutre. Suposem que podem codificar els elements de $G$ en un sistema de $n$ qubits i que podem implementar la transformació 2.10 usant un nombre constant de portes quàntiques. Siguin $g \in G$ i $V$ una *caixa negra* que efectua la transformació

$$|j\rangle|h\rangle \to |j\rangle|g^j h\rangle$$

per a $h \in G$ i $j$ un nombre natural. Suposem que tenim

$$t = 2n + 1 + \left\lceil \log(2 + \frac{1}{2\varepsilon}) \right\rceil$$

qubits inicialitzats a $|0\rangle$ i $n$ qubits inicialitzats en l'estat $|e\rangle$.

1. **Estat inicial:** $|0\rangle|e\rangle$,
2. **Superposició en el primer registre:**
$$\to \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle|e\rangle,$$
3. **Apliquem $V$:**
$$\to \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle|g^j\rangle \simeq \frac{1}{\sqrt{r2^t}} \sum_{s=0}^{r-1} \sum_{j=0}^{2^t-1} \exp\left(\frac{sj}{r}\right) |j\rangle|u_s\rangle,$$
4. **Apliquem la transformada de Fourier inversa sobre el primer registre:**
$$\to \frac{1}{r} \sum_{s=0}^{r-1} \widetilde{|s/r\rangle}|u_s\rangle,$$

5. **Mesurem el primer registre:** $\rightarrow \widetilde{s/r}$,
6. **Apliquem fraccions continuades:** $\rightarrow r$.

# Capítol 3

# Algoritme de Shor

En aquest capítol i en el proper ens dedicarem a l'estudi de dos algoritmes quàntics per a la factorització de nombres enters. El primer d'ells és l'algoritme de Shor [9], que va ésser publicat originalment l'any 1994.

Aquest algoritme pren com a dada d'entrada un nombre enter $N$ compost de $n$ bits i en retorna un factor no trivial. La complexitat d'aquest algoritme és $O(n^3)$. En altres paraules: és polinòmic en el nombre de bits de $N$. La seva aparició va ser celebrada perquè va demostrar que el problema de factoritzar un nombre enter es pot resoldre en temps polinòmic usant un computador quàntic. El mètode per a computadors clàssics més eficient conegut actualment és el garbell de cossos de nombres, que té complexitat subexponencial

$$O\left(e^{n^{1/3}}\log(n)^{2/3}\right).$$

## 3.1 Plantejament de l'algoritme

L'algoritme de Shor és una aplicació de l'algoritme quàntic per trobar l'ordre d'un element en un grup arbitrari que hem explicat al capítol anterior. El grup que s'utilitza és el grup d'unitats $(\mathbb{Z}/N\mathbb{Z})^*$ dels nombres enters mòdul $N$.

Per a un enter $x \in \{2, \dots N-1\}$ triat aleatòriament amb la condició $\text{mcd}(x, N) = 1$, l'algoritme 2.3.3 ens retorna el mínim enter positiu $r$ tal que

$$x^r \equiv 1 \pmod{N}.$$

Explicarem en aquesta secció com trobar un factor no trivial de $N$ a partir d'aquesta informació.

En primer lloc, expliquem com codificar en un computador quàntic la informació referent al grup $(\mathbb{Z}/N\mathbb{Z})^*$. Codifiquem les classes de residus mòdul

$N$ usant els enters del conjunt $\{0, 1, \ldots, N-1\}$. Si $n$ és el mínim enter tal que $N \leq 2^n$, necessitem $n$ qubits per a codificar aquests nombres enters.

L'operació que cal implementar és la multiplicació mòdul $N$. Això vol dir que per a obtenir $ab$ amb $a, b \in \{0, \ldots, N-1\}$, calcularem el producte de $a$ per $b$ com a nombres enters i posteriorment efectuarem la divisió entera d'aquest producte per $N$. Si usem l'algoritme per a la multiplicació a mà en base 2, caldrà aplicar $O(n^2)$ portes quàntiques per a obtenir el producte de dos nombres de $n$ bits (exactament el mateix resultat que es té per a un computador clàssic). Hi ha algoritmes per a la multiplicació que redueixen asimptòticament el nombre d'operacions (vegeu [9, 3]), però l'algoritme per a la multiplicació a mà ja satisfà els nostres interessos actuals. Dividirem per $N$ mitjançant l'algoritme per a la divisió a mà en base 2, que té complexitat $O(n^2)$. Per tant, necessitem $O(n^2)$ portes quàntiques per efectuar una multiplicació mòdul $N$.

Per tal d'implementar l'exponenciació en $(\mathbb{Z}/N\mathbb{Z})^*$ mitjançant un mètode binari d'exponenciació necessitem

$$t = 2n + 1 + \left\lceil \log\left(2 + \frac{1}{2\varepsilon}\right)\right\rceil$$

qubits per al primer registre per a obtenir una aproximació de fase amb precisió de $2n+1$ xifres binàries amb probabilitat com a mínim $1-\varepsilon$. Per a exponents d'aquesta mida, la implementació de l'exponenciació (vegeu la secció 2.4) requereix de $t-1$ multiplicacions mòdul $N$ per a calcular els valors $x^{2^j}$, per a $0 \leq j \leq t-1$ i un màxim de $t-1$ multiplicacions modulars per a calcular

$$x^k = x^{k_{t-1}2^{t-1}} \cdots x^{k_1 2^1} x^{k_0 2^0},$$

si $k = \sum_{i=0}^{t-1} k_i 2^i$. En aquest sentit, cal efectuar $O(n)$ multiplicacions mòdul $N$, cadascuna de les quals suposa efectuar $O(n^2)$ operacions bàsiques. Per tant, podem fer l'exponenciació binària en aquest grup mitjançant $O(n^3)$ operacions bàsiques.

Triem $x$ a l'atzar en $\{2, 3, \ldots, N-1\}$; l'algoritme 2.3.3 ens retorna l'ordre $r$ de $x$ mòdul $N$. Suposem que $r$ és un nombre parell. Aleshores, per a $y := x^{r/2} \in (\mathbb{Z}/N\mathbb{Z})^*$ es té que

$$(y-1)(y+1) \equiv y^2 - 1 \equiv x^r - 1 \equiv 0 \pmod{N}.$$

Es tenen les possibilitats:

1. $y \equiv 1 \pmod{N}$. Això no pot passar, ja que aquest fet implica que l'ordre de $x$ mòdul $N$ és $r/2$, la qual cosa és una contradicció.

2. $y \equiv -1 \pmod{N}$. En aquesta situació no podrem dir res, i l'algoritme no ens proporcionarà cap factor no trivial de $N$.

3. $y \not\equiv \pm 1 \pmod{N}$. Aleshores $\mathrm{mcd}(y-1, N)$ i $\mathrm{mcd}(y+1, N)$ són factors no trivials de $N$.

Remarquem que l'algoritme falla en els casos en què $r$ és senar o bé, quan $r$ és parell, si $y \equiv -1 \pmod{N}$. Resumim el procediment que acabem d'explicar en un algoritme.

**3.1.1 Algoritme.** (Shor). Sigui $N$ un enter compost.

1. Si $N$ és parell retornem el valor 2.
2. Determinem si $N = a^b$, amb $a \geq 3, b \geq 2$. En cas que així sigui, retornem el valor $a$.
3. Triem a l'atzar $x \in \{2, 3, \ldots, N-1\}$. Calculem $\mathrm{mcd}(x, N)$. En cas que aquest màxim comú divisor no sigui 1, retornem el valor $\mathrm{mcd}(x, N)$.
4. Usem l'algoritme 2.3.3 per a trobar l'ordre $r$ de $x$ mòdul $N$.
5. Si $r$ és senar retornem un missatge d'error i finalitzem. En cas contrari, procedim al pas 6.
6. Calculem $y := x^{r/2} \in (\mathbb{Z}/N\mathbb{Z})^*$.
7. Si $y \equiv -1 \pmod{N}$ retornem un missatge d'error i finalitzem. En cas contrari procedim al pas 8.
8. Calculem $\mathrm{mcd}(y-1, N)$ i retornem aquest valor com a factor no trivial de $N$.

Els passos 1, 2, 3, 5, 6, 7 i 8 de l'algoritme es poden efectuar mitjançant algoritmes clàssics de complexitat $O(n^3)$ o inferior. Resta calcular la complexitat computacional del pas 4. Per implementar aquest pas necessitem un computador quàntic amb dos registres: un primer de $t = 2n + 1 + \lceil \log(2 + 1/(2\varepsilon)) \rceil$ qubits i un segon de $n$ qubits, i cal efectuar $O(n^3)$ operacions.

Això implica que la despesa total en espai és de $O(n)$ qubits. Pel que fa al temps d'execució, cal efectuar $O(n^3)$ operacions.

## 3.2 Probabilitat d'èxit

Ens disposem estudiar la probabilitat que l'algoritme 3.1.1 finalitzi amb èxit. D'una banda, cal fitar la probabilitat que el valor de $r$ obtingut al pas 4 sigui senar. De l'altra banda, cal fitar inferiorment la probabilitat que $r$ sigui parell i $x^{r/2} \equiv -1 \pmod{N}$.

**3.2.1 Notació.** Per a un primer $p$ i un enter $n$ qualssevol, $v_p(n)$ denota la valoració $p$-àdica de $n$, és a dir: la màxima potència de $p$ que divideix $n$.

**3.2.2 Lema.** *Siguin $p$ un primer senar i $d := v_2(\varphi(p^\alpha))$, on $\varphi$ denota la funció d'Euler. Aleshores la probabilitat que l'ordre d'un element $x \in (\mathbb{Z}/p^\alpha\mathbb{Z})^*$ triat a l'atzar tingui valoració 2-àdica igual a $d$ és exactament $1/2$.*

Demostració: Notem que $\varphi(p^\alpha) = p^{\alpha-1}(p-1)$ és un nombre enter parell, ja que $p$ és senar. Per tant, $d \geq 1$ i $(\mathbb{Z}/p^\alpha\mathbb{Z})^*$ és un grup cíclic d'ordre parell.

Sigui $r$ l'ordre de $x$ mòdul $p^\alpha$, de manera que $v_2(r) \leq d$. Es té que $v_2(r) < d$ si i només $x$ és quadrat en $(\mathbb{Z}/p^\alpha\mathbb{Z})^*$. Com que exactament la meitat dels elements de $(\mathbb{Z}/p^\alpha\mathbb{Z})^*$ són quadrats, la probabilitat que l'ordre d'un element triat a l'atzar tingui valoració 2-àdica igual a $d$ és $1/2$. □

**3.2.3 Proposició.** *Suposem que $N = p_1^{\alpha_1} \cdots p_m^{\alpha_m}$ és la factorització de $N$ en factors primers diferents, amb $N$ senar i $m > 1$. Sigui $x \in (\mathbb{Z}/N\mathbb{Z})^*$ escollit a l'atzar, i sigui $r$ l'ordre de $x$ mòdul $N$. Aleshores*

$$P\left(r \text{ és senar o } (\ r \text{ parell i } x^{r/2} \equiv -1 \pmod N)\right) \leq \frac{1}{2^m}.$$

Demostració: Mitjançant el teorema xinès del residu, triar $x$ a l'atzar en $(\mathbb{Z}/N\mathbb{Z})^*$ és equivalent a triar $x_j \in (\mathbb{Z}/p_j^{\alpha_j}\mathbb{Z})^*$ per a $1 \leq j \leq m$ i imposar que $x \equiv x_j \pmod{p_j^{\alpha_j}}$ per a tot $j$.

Sigui $r_j$ l'ordre de $x_j$ mòdul $p_j^{\alpha_j}$. Siguin $d_j = v_2(r_j)$ i $d = v_2(r)$. Mostrarem que per tal de tenir $r$ senar o $r$ parell i $x^{r/2} \equiv -1 \pmod N$ és necessari que $d_j$ prengui el mateix valor per a cada valor de $j$. Pel lema 3.2.2, la probabilitat que això passi és com a molt $1/2^m$.

Suposem en primer lloc que $r$ és senar. Es té que $r_j \mid r$ per a tot $j$ i, per tant, $v_2(r_j) = 0$ per a tot $j$. En segon lloc, suposem que $r$ és parell i que $x^{r/2} \equiv -1 \pmod N$. Aleshores

$$x^{r/2} \equiv -1 \pmod{p_j^{\alpha_j}}, \qquad 1 \leq j \leq m$$

i, per tant, $r_j \nmid \frac{r}{2}$. Com que $r_j \mid r$, cal que sigui $v_2(r_j) = v_2(r)$ per a tot $j$. □

La proposició ens diu que la probabilitat de factoritzar un enter $N$ compost i senar amb l'algoritme de Shor és com a mínim

$$1 - \frac{1}{2^m},$$

on $m$ és el nombre de primers diferents que divideixen $N$. En el pitjor dels casos, en què $m = 2$, aquesta probabilitat és com a mínim $3/4$. Això ens garanteix que repetint l'algoritme un nombre relativament petit de vegades obtindrem un factor no trivial de $N$ amb probabilitat gran.

# Capítol 4

# Un algoritme de factorització mitjançant corbes el·líptiques

## 4.1 Corbes el·líptiques sobre $\mathbb{Z}/N\mathbb{Z}$

Siguin $N$ i $n$ dos nombres enters, $n > 0$. Considerem el conjunt

$$E := \left\{(x_1, \ldots, x_{n+1}) \in (\mathbb{Z}/N\mathbb{Z})^{n+1};\ \mathrm{mcd}(x_1, \ldots, x_{n+1}, N) = 1\right\}.$$

El grup d'unitats $(\mathbb{Z}/N\mathbb{Z})^*$ actua sobre $E$ per multiplicació:

$$\begin{array}{rcl} (\mathbb{Z}/N\mathbb{Z})^* \times E & \longrightarrow & E \\ (u, (x_1, x_2, \ldots, x_{n+1})) & \longmapsto & (ux_1, ux_2, \ldots, ux_{n+1}). \end{array}$$

**4.1.1 Definició.** El conjunt d'òrbites de $E$ per aquesta acció de $(\mathbb{Z}/N\mathbb{Z})^*$ s'anomena el $n$-espai projectiu sobre $\mathbb{Z}/N\mathbb{Z}$. Denotem aquest conjunt per $\mathbb{P}_n(\mathbb{Z}/N\mathbb{Z})$. Com és habitual, denotarem l'òrbita de $(x_1, \ldots, x_{n+1})$ per $(x_1 : \ldots : x_{n+1})$.

**4.1.2 Exemple.** Sigui $N = p$ un nombre primer. Aleshores $E = \mathbb{F}_p^{n+1} \setminus \{0\}$ i el que acabem de definir no és res més que el $n$-espai projectiu sobre $\mathbb{F}_p$.

La definició d'espai projectiu sobre un anell com $\mathbb{Z}/N\mathbb{Z}$ és molt semblant a la definició que es dóna habitualment per a un cos finit, però l'estructura de l'espai resultant és diferent, fins al punt que fa molt més complicat treballar amb moltes nocions habituals en la geometria projectiva, com per exemple la d'intersecció entre dues varietats lineals.

És ben conegut que el $n$-espai projectiu sobre un cos admet una descomposició com a unió disjunta d'un $n$-espai afí i un hiperplà anomenat *hiperplà de l'infinit*, que té estructura de $(n-1)$-espai projectiu. En el cas de $\mathbb{P}_n(\mathbb{Z}/N\mathbb{Z})$ aquesta descomposició varia sensiblement.

Els punts $P = (x_1 : \ldots : x_{n+1}) \in \mathbb{P}_n(\mathbb{Z}/N\mathbb{Z})$ que satisfan $x_{n+1} = 0$ estan en bijecció de manera òbvia amb $\mathbb{P}_{n-1}(\mathbb{Z}/N\mathbb{Z})$ i els anomenarem *punts a l'infinit*.

D'altra banda, si $x_{n+1} \in (\mathbb{Z}/N\mathbb{Z})^*$, aleshores $P$ admet un únic representant de la forma $(y_1, \ldots, y_n, 1)$. Assignant a $P$ el punt $(y_1, \ldots, y_n) \in \mathbb{A}_n(\mathbb{Z}/N\mathbb{Z})$ obtenim una bijecció entre els punts amb la darrera coordenada invertible i el $n$-espai afí sobre $\mathbb{Z}/N\mathbb{Z}$. Direm que aquests punts són *punts afins*.

Resta considerar un tercer cas, en què $1 < \mathrm{mcd}(x_{n+1}, N) < N$. Aquests punts, que no són punts a l'infinit ni punts afins, els anomenarem *punts especials* i el seu conjunt el denotarem per $\mathbb{P}_n^s(\mathbb{Z}/N\mathbb{Z})$. Obtenim una descomposició de $\mathbb{P}_n(\mathbb{Z}/N\mathbb{Z})$ com la reunió disjunta

$$\mathbb{P}_n(\mathbb{Z}/N\mathbb{Z}) = \mathbb{P}_{n-1}(\mathbb{Z}/N\mathbb{Z}) \uplus \mathbb{A}_n(\mathbb{Z}/N\mathbb{Z}) \uplus \mathbb{P}_n^s(\mathbb{Z}/N\mathbb{Z}).$$

**4.1.3 Exemple.** Descrivim els punts de la recta projectiva sobre $\mathbb{Z}/4\mathbb{Z}$. D'una banda, hi ha un punt a l'infinit, $(1 : 0)$, i els punts afins són

$$\{(0 : 1), (1 : 1), (2 : 1), (3 : 1)\}.$$

Finalment, tenim el punt $(1 : 2)$ com a punt especial.

Pel que fa al propòsit d'aquest treball, una corba el·líptica sobre $\mathbb{Z}/N\mathbb{Z}$ serà el conjunt de punts de $\mathbb{P}_2(\mathbb{Z}/N\mathbb{Z})$ les coordenades dels quals satisfacin una equació de Weierstrass admissible.

Suposarem d'ara endavant que $\mathrm{mcd}(6, N) = 1$. D'aquesta manera evitem haver de tractar les complicacions de les equacions de Weierstrass en característiques 2 i 3. Com que la nostra intenció final serà trobar un factor no trivial de $N$, aquesta hipòtesi no suposa cap restricció en la pràctica.

Treballarem en el pla projectiu $\mathbb{P}_2(\mathbb{Z}/N\mathbb{Z})$. Fixem coordenades $(x : y : z)$ pels punts del pla projectiu i prenem la recta d'equació $Z = 0$ com a recta de l'infinit en aquestes coordenades.

**4.1.4 Definició.** Siguin $a, b \in \mathbb{Z}/N\mathbb{Z}$ tals que $\mathrm{mcd}(4a^3 + 27b^2, N) = 1$. La corba el·líptica $E(a, b)$ és l'equació homogènia de tercer grau

$$E(a, b) : \quad Y^2Z = X^3 + aXZ^2 + bZ^3.$$

Direm que els punts de la corba el·líptica són els elements del conjunt

$$E(a, b)(\mathbb{Z}/N\mathbb{Z}) := \left\{(x : y : z) \in \mathbb{P}_2(\mathbb{Z}/N\mathbb{Z});\ y^2z = x^3 + axz^2 + bz^3\right\}.$$

**4.1.5 Observació.** Si cerquem els punts $(x : y : z)$ de $E(a, b)(\mathbb{Z}/N\mathbb{Z})$ a l'infinit arribem immediatament a la condició $x^3 = 0$. Si $\mathbb{Z}/N\mathbb{Z}$ no té nilpotents, això és, si $N$ és lliure de quadrats, cal que sigui $x = 0$. En aquest

cas, l'únic punt a l'infinit és el que té coordenades $(0:1:0)$. En el cas en què $N$ no és lliure de quadrats, la situació esdevé més complicada ja que la corba pot tenir altres punts a l'infinit.

Si $N = p$ és un nombre primer no hem fet res més que donar la definició de corba el·líptica en forma de Weierstrass sobre $\mathbb{F}_p$. Si $N$ no és primer, per a cada divisor primer $p$ de $N$ podem considerar la reducció mòdul $p$ de la corba $E(a,b)$.

**4.1.6 Definició.** Sigui $N'$ un divisor de $N$. Per a cada parella d'elements $a, b \in \mathbb{Z}/N\mathbb{Z}$ tals que $\mathrm{mcd}(4a^3 + 27b^2, N) = 1$, denotem per $\bar{a}$ i $\bar{b}$ les seves respectives reduccions mòdul $N'$. Direm que la corba el·líptica sobre $\mathbb{Z}/N'\mathbb{Z}$

$$E_{N'}(a,b): \quad Y^2Z = X^3 + \bar{a}XZ^2 + \bar{b}Z^3,$$

és la reducció mòdul $N'$ de la corba $E(a,b)$.

Observem que la condició $\mathrm{mcd}(4a^3 + 27b^2, N) = 1$ garanteix que per a cada divisor primer $p$ de $N$ la corba $E_p(a,b)$ és donada per una forma de Weierstrass no singular i que per tant es tracta d'una corba el·líptica sobre $\mathbb{F}_p$.

De manera semblant al que succeeix quan treballem sobre un cos arbitrari, els punts d'una corba el·líptica sobre $\mathbb{Z}/N\mathbb{Z}$ també tenen estructura de grup. A més, aquesta llei de grup està totalment determinada per les reduccions mòdul $p$ de la corba, per a cada divisor primer $p \mid N$. El nostre proper objectiu és tractar aquesta llei d'addició.

En tota la resta de la secció, suposarem que $N$ es un nombre enter lliure de quadrats. El resultat següent relaciona el conjunt de punts d'una corba el·líptica sobre $\mathbb{Z}/N\mathbb{Z}$ i els conjunts de punts de les seves reduccions mòdul $p$ per a cada divisor primer $p \mid N$.

**4.1.7 Teorema.** *Siguin $N$ un nombre enter lliure de quadrats i $a, b$ dos nombres enters tals que* $\mathrm{mcd}(4a^3 + 27b^2, N) = 1$. *La reducció mòdul $p$ per a cada primer $p \mid N$ indueix una aplicació bijectiva*

$$E(a,b)(\mathbb{Z}/N\mathbb{Z}) \longrightarrow \prod_{p|N} E_p(a,b)(\mathbb{F}_p).$$

Demostració: Es tracta d'una conseqüència del teorema xinès del residu. Procedim per inducció sobre el nombre de divisors primers de $N$. Si $N$ és primer el resultat és obvi.

Suposem a continuació que $N = pN'$ amb $p \nmid N'$ i $p$ primer. L'aplicació que apareix en l'enunciat factoritza trivialment a través de l'aplicació

$$E(a,b)(\mathbb{Z}/N\mathbb{Z}) \longrightarrow E_p(a,b)(\mathbb{F}_p) \times E_{N'}(a,b)(\mathbb{Z}/N'\mathbb{Z}). \tag{4.1}$$

Per hipòtesi d'inducció, l'aplicació

$$E_{N'}(a,b)(\mathbb{Z}/N'\mathbb{Z}) \longrightarrow \prod_{p|N'} E_p(a,b)(\mathbb{F}_p)$$

és bijectiva i, per tant, és suficient demostrar que (4.1) és una bijecció.

En primer lloc, suposem que $(x_1 : y_1 : z_1)$ i $(x_2 : y_2 : z_2)$ són les coordenades de dos punts de $E(a,b)(\mathbb{Z}/N\mathbb{Z})$ que s'apliquen en la mateixa parella de punts de $E_p(a,b)(\mathbb{F}_p) \times E_{N'}(a,b)(\mathbb{Z}/N'\mathbb{Z})$. Tindrem que

$$\begin{cases} (x_1,y_1,z_1) \equiv u(x_2,y_2,z_2) \pmod{p}, \\ (x_1,y_1,z_1) \equiv v(x_2,y_2,z_2) \pmod{N'}, \end{cases}$$

on les congruències són coordenada a coordenada i $u, v$ són nombres enters tals que $\mathrm{mcd}(u,p) = 1$ i $\mathrm{mcd}(v,N') = 1$. Pel teorema xinès del residu, existeix un únic enter $w$ mòdul $N$ tal que

$$\begin{cases} w \equiv u \pmod{p}, \\ w \equiv v \pmod{N'}. \end{cases}$$

A més, observem que $w$ és invertible mòdul $N$, ja que no hi té factors en comú. Se satisfan les congruències

$$\begin{cases} (x_1,y_1,z_1) \equiv w(x_2,y_2,z_2) \pmod{p}, \\ (x_1,y_1,z_1) \equiv w(x_2,y_2,z_2) \pmod{N'}. \end{cases}$$

Podem aplicar novament el teorema xinès del residu per deduir que

$$(x_1 : y_1 : z_1) = (x_2 : y_2 : z_2).$$

Per tant, hem demostrat que l'aplicació és injectiva.

Per veure que l'aplicació és exhaustiva sigui

$$((\xi_1 : \eta_1 : \zeta_1), (\xi_2 : \eta_2 : \zeta_2)) \in E_p(a,b)(\mathbb{F}_p) \times E_{N'}(a,b)(\mathbb{Z}/N'\mathbb{Z}).$$

Pel teorema xinès del residu, existeix una única terna $(x,y,z)$ d'enters mòdul $N$ tals que

$$\begin{cases} (x,y,z) \equiv (\xi_1,\eta_1,\zeta_1) \pmod{p}, \\ (x,y,z) \equiv (\xi_2,\eta_2,\zeta_2) \pmod{N'} \end{cases}$$

Comprovem que $\mathrm{mcd}(x,y,z,N) = 1$. Per a cada divisor primer $l \mid N$ es té que

$$\begin{cases} l \nmid \mathrm{mcd}(\xi_1,\eta_1,\zeta_1) & \text{si } l = p, \\ l \nmid \mathrm{mcd}(\xi_2,\eta_2,\zeta_2) & \text{si } l \neq p. \end{cases}$$

En qualsevol de les dues situacions deduïm que $l \nmid \mathrm{mcd}(x,y,z)$.

Resta demostrar que $(x : y : z) \in E(a,b)(\mathbb{Z}/N\mathbb{Z})$. Això és equivalent a demanar que se satisfaci la congruència

$$y^2 z \equiv x^3 + axz^2 + bz^3 \pmod{N}.$$

El teorema xinès del residu ens indica que aquesta congruència se satisfà, perquè és certa simultàniament mòdul $p$ i mòdul $N'$. □

Recordem breument les fórmules d'addició dels punts d'una corba el·líptica sobre un cos finit de $p$ elements.

**4.1.8 Proposició.** *Siguin $p$ un nombre primer i $E$ una corba el·líptica sobre $\mathbb{F}_p$ donada per l'equació de Weierstrass (en coordenades afins)*

$$Y^2 = X^3 + aX + b, \qquad a, b \in \mathbb{F}_p.$$

*$E$ té un únic punt a l'infinit $O := (0 : 1 : 0)$. Siguin $P = (\xi_1, \eta_1)$, $Q = (\xi_2, \eta_2) \in E(\mathbb{F}_p)$ dos punts diferents de $O$. Definim:*

$$\lambda := \begin{cases} \dfrac{\eta_2 - \eta_1}{\xi_2 - \xi_1}, & \text{si } \xi_2 \neq \xi_1, \\ \dfrac{3\xi_1^2 + a}{2\eta_1}, & \text{si } \xi_2 = \xi_1 \text{ i } \eta_1 \neq 0, \end{cases}$$
$$\xi_3 := \lambda^2 - \xi_1 - \xi_2,$$
$$\eta_3 := -\eta_1 - \lambda(\xi_3 - \xi_1).$$

*En els casos en què està definit, es té que $(\xi_3, \eta_3) \in E(\mathbb{F}_p)$. Definim una llei binària per les regles següents:*

*Si $P = O$, definim $P + Q := Q$. Si $Q = O$, definim $P + Q = P$. Suposem a continuació que $P, Q \neq O$.*

*Si $P$ i $Q$ satisfan que $\xi_2 \neq \xi_1$, definim $P + Q := (\xi_3, \eta_3)$.*

*Finalment, si $P$ i $Q$ satisfan que $\xi_2 = \xi_1$, aleshores necessàriament es té que $\eta_2 = \pm\eta_1$. Si és $\eta_2 = \eta_1 \neq 0$, posarem $P + Q = P + P := (\xi_3, \eta_3)$ i, si és $\eta_2 = -\eta_1$, posarem $P + Q := O$.*

*Aquesta llei binària dóna estructura de grup abelià al conjunt $E(\mathbb{F}_p)$.* □

Si treballem a $\mathbb{Z}/N\mathbb{Z}$ i escrivim aquestes fórmules en coordenades projectives, podem definir una llei binària en $E(a,b)(\mathbb{Z}/N\mathbb{Z})$. Recordem que $N$ és lliure de quadrats i que $\mathrm{mcd}(6, N) = 1$. Donats ara dos punts $P = (x_1 : y_1 : z_1), Q = (x_2 : y_2 : z_2) \in E(a,b)(\mathbb{Z}/N\mathbb{Z})$, definim les quantitats:

$$L := x_2 z_1 - x_1 z_2,$$
$$M := 2y_1 z_1,$$
$$A := y_2 z_1 - y_1 z_2,$$
$$B := 3x_1^2 + a z_1^2,$$
$$C := 2x_1 z_2 + x_2 z_1$$

i, a partir d'aquestes:

$$x_3 := \begin{cases} A^2Lz_1z_2 - L^3(x_1z_2 + x_2z_1), & \text{si } L \neq 0, \\ B^2Mz_1z_2 - M^3(x_2z_1 + x_1z_2), & \text{si } L = 0 \text{ i } y_2z_1 + y_1z_2 \neq 0, \end{cases}$$

$$y_3 := \begin{cases} -L^3y_1z_2 - A^3z_1z_2 + AL^2C, & \text{si } L \neq 0, \\ -M^3y_1z_2 - B^3z_1z_2 + BM^2C, & \text{si } L = 0 \text{ i } y_2z_1 + y_1z_2 \neq 0, \end{cases}$$

$$z_3 := \begin{cases} L^3z_1z_2, & \text{si } L \neq 0, \\ M^3z_1z_2, & \text{si } L = 0 \text{ i } y_2z_1 + y_1z_2 \neq 0. \end{cases}$$

**4.1.9 Lema.** *En els casos en què està definit,* $(x_3 : y_3 : z_3) \in E(a,b)(\mathbb{Z}/N\mathbb{Z})$.

Demostració: Les fórmules que defineixen $(x_3 : y_3 : z_3)$ provenen de reescriure l'addició definida mòdul $p$ a la proposició 4.1.8. Per a comprovar-ho, només cal substituir

$$\xi_i = \frac{x_i}{z_i}, \quad \eta_i = \frac{x_i}{z_i}$$

en les fórmules de la proposició 4.1.8, per a $i = 1, 2$. Fent les manipulacions òbvies, s'obté

$$(\xi_3 : \eta_3 : 1) = (x_3 : y_3 : z_3).$$

Això ens indica que la reducció de $(x_3 : y_3 : z_3)$ mòdul $p$ és un punt de $E_p(a,b)(\mathbb{F}_p)$ per a cada divisor primer $p \mid N$. Pel teorema 4.1.7 deduïm que $(x_3 : y_3 : z_3) \in E(a,b)(\mathbb{Z}/N\mathbb{Z})$. □

A continuació definim una addició de punts. Posem $O := (0 : 1 : 0)$ i siguin $P = (x_1 : y_1 : z_1), Q = (x_2 : y_2 : z_2)$ dos punts de $E(a,b)(\mathbb{Z}/N\mathbb{Z})$.

En primer lloc, si $P = O$ definim $P + Q := Q$ i si $Q = O$ definim $P + Q = P$.

A continuació, si $y_2z_1 + y_1z_2 = 0$, prenem $P + Q := O$. En cas contrari, definim $P + Q := (x_3 : y_3 : z_3)$.

**4.1.10 Teorema.** *L'addició que acabem de definir dóna a* $E(a,b)(\mathbb{Z}/N\mathbb{Z})$ *estructura de grup abelià. Aquesta addició correspon amb la que s'obté per transport d'estructura mitjançant la bijecció obtinguda al teorema 4.1.7 i, en particular, converteix les aplicacions de reducció mòdul* $p \mid N$

$$E(a,b)(\mathbb{Z}/N\mathbb{Z}) \longrightarrow E_p(a,b)(\mathbb{F}_p)$$

*en homomorfismes de grups abelians.*

DEMOSTRACIÓ: S'aplica el mateix raonament que per demostrar el lema 4.1.9. El fet que la llei d'addició coincideixi amb la donada a 4.1.8 canviant coordenades projectives per coordenades afins i la bijecció obtinguda al teorema 4.1.7 demostren que la llei d'addició coincideix amb l'estructura transportada a partir del grup abelià $\prod_{p|N} E_p(a,b)(\mathbb{F}_p)$.

Per tant, aquesta llei d'addició dóna a $E(a,b)(\mathbb{Z}/N\mathbb{Z})$ estructura de grup abelià i les aplicacions de reducció mòdul $p \mid N$ esdevenen homomorfismes de grups, ja que resulten de la composició

$$E(a,b)(\mathbb{Z}/N\mathbb{Z}) \to \prod_{p|N} E_p(a,b)(\mathbb{F}_p) \to E_p(a,b)(\mathbb{F}_p).$$

□

**4.1.11 Notació.** Per a un primer $p \mid N$ i per a un parell de punts $P, Q \in E(a,b)(\mathbb{Z}/N\mathbb{Z})$, escriurem

$$P \equiv Q \pmod{p}$$

si la reducció mòdul $p$ d'ambdós punts coincideix.

## 4.2 Un algoritme quàntic de factorització mitjançant corbes el·líptiques

**4.2.1 Notació.** En tota aquesta secció $N$ denotarà un nombre enter lliure de quadrats i tal que $\mathrm{mcd}(6, N) = 1$.

L'algoritme 2.3.3 per a trobar l'ordre d'un element en un grup arbitrari pot ser utilitzat també per al grup de punts d'una corba el·líptica sobre $\mathbb{Z}/N\mathbb{Z}$. Obtenim un algoritme que permet trobar l'ordre d'un punt en una corba el·líptica sobre $\mathbb{F}_p$ en un temps polinòmic en el nombre de xifres binàries de $p$. Si treballem una mica més podem obtenir un algoritme per a factoritzar nombres enters.

En primer lloc explicarem com codificar en un computador quàntic els punts d'una corba el·líptica sobre $\mathbb{Z}/N\mathbb{Z}$. Aquests punts són elements del pla projectiu $\mathbb{P}_2(\mathbb{Z}/N\mathbb{Z})$ i, per tant, es poden representar mitjançant coordenades $(x : y : z)$ amb $x, y, z \in \{0, 1, \ldots, N-1\}$ i tals que $\mathrm{mcd}(x, y, z, N) = 1$.

Sigui $n$ el mínim nombre enter tal que $N \leq 2^n$. Aleshores $N$, així com també $x, y$ i $z$, es poden representar mitjançant $n$ xifres en base 2. Per tant, la manera de codificar un punt serà mitjançant $3n$ qubits, a raó de $n$ qubits per codificar cada coordenada del punt a la manera usual. Si $P = (x : y : z)$, aleshores el representarem amb tres registres

$$|P\rangle := |x\rangle|y\rangle|z\rangle.$$

Tot i que aquesta representació no té en compte l'acció de $(\mathbb{Z}/N\mathbb{Z})^*$ i, per tant, aquesta manera de representar el punt $P$ no és única, aquest fet no suposarà cap inconvenient.

L'operació de grup en aquest cas és la suma de punts definida a la secció anterior. Per a sumar dos punts codificats usant $3n$ qubits per a cadascun, cal efectuar un màxim de 92 operacions en $\mathbb{Z}/N\mathbb{Z}$, que són multiplicacions i sumes de nombres enters i divisions enteres per $N$. Usem l'algoritme per multiplicar a mà en base 2, que requereix de $O(n^2)$ operacions bàsiques. Pel que fa a les sumes i divisions enteres, l'algoritme per efectuar aquestes operacions a mà en base 2 és de complexitat menor o igual. En conseqüència, sumar dos punts sobre una corba el·líptica mòdul $N$ requereix de l'ús de $O(n^2)$ operacions bàsiques.

Procedim a exposar l'algoritme per factoritzar $N$. Sigui

$$O := (0 : 1 : 0) \in \mathbb{P}_2(\mathbb{Z}/N\mathbb{Z}).$$

Triem a l'atzar $x, y, a \in \{0, 1, \ldots, N-1\}$ i calculem el valor

$$b := y^2 - x^2(x + a) \in \mathbb{Z}/N\mathbb{Z}.$$

Si $\mathrm{mcd}(4a^3 + 27b^2, N) = 1$, aleshores $E(a, b)$ és una corba el·líptica sobre $\mathbb{Z}/N\mathbb{Z}$ i se satisfà que $P = (x : y : 1) \in E(a, b)(\mathbb{Z}/N\mathbb{Z})$.

Per a aquest punt i aquesta corba podem usar l'algoritme 2.3.3. Obtenim l'ordre $r$ del punt $P$ en $E(a, b)(\mathbb{Z}/N\mathbb{Z})$, això és: el mínim nombre enter positiu tal que $rP = O$. El càlcul de la complexitat que presenta aplicar l'algoritme 2.3.3 en aquest context és molt semblant als càlculs que hem realitzat per l'algoritme de Shor, llevat del fet que un element del grup $(\mathbb{Z}/N\mathbb{Z})^*$ es codifica mitjançant $n$ qubits i, en canvi, codificar un element del grup $E(a, b)(\mathbb{Z}/N\mathbb{Z})$ requereix $3n$ qubits.

Per tal d'implementar l'exponenciació en $E(a, b)(\mathbb{Z}/N\mathbb{Z})$, que correspon al càlcul dels múltiples d'un punt, necessitem

$$t = 6n + 1 + \left\lceil \log\left(2 + \frac{1}{2\varepsilon}\right)\right\rceil$$

qubits per al primer registre per a obtenir una aproximació de fase amb precisió de $6n + 1$ xifres binàries amb probabilitat com a mínim $1 - \varepsilon$. Per a exponents d'aquesta mida, la implementació de l'exponenciació binària (vegeu la secció 2.4) requereix de $t - 1$ operacions per a calcular els valors $2^j P$, per a $0 \leq j \leq t - 1$ i un màxim de $t - 1$ operacions per a calcular

$$kP = k_{t-1}2^{t-1}P + \cdots + k_1 2^1 P + k_0 2^0 P,$$

si $k = \sum_{i=0}^{t-1} k_i 2^i$ és l'expressió del multiplicador $k$ de $P$. La complexitat que obtenim és la mateixa que en el cas de l'algoritme de Shor: cal efectuar $O(n)$

operacions en el grup, cadascuna de les quals suposa efectuar $O(n^2)$ operacions bàsiques. Podem fer, per tant, l'exponenciació binària en aquest grup mitjançant $O(n^3)$ operacions bàsiques. L'espai necessari també és $O(n)$.

Cal explicar com passem de conèixer l'ordre d'un punt en una corba a conèixer un factor no trivial de $N$. Suposem que $r$ és parell i sigui

$$Q := \frac{r}{2}P.$$

Aleshores $2Q = O$. Per a tot primer $p \mid N$ es tindrà que

$$2Q \equiv O \pmod{p}.$$

En aquesta situació hi ha només dues possibilitats: o bé $Q \equiv O \pmod{p}$, o $Q \not\equiv O \pmod{p}$. Això ens porta a distingir tres situacions possibles:

1. $Q \equiv O \pmod{p}$ per a tot primer $p \mid N$. Aquesta situació no la trobarem mai: el teorema 4.1.7 ens garanteix que en aquesta situació $Q = O$, fet que implica que l'ordre de $P$ és $r/2$ enlloc de $r$. Això és una contradicció.

2. $Q \not\equiv O \pmod{p}$ per a tot primer $p \mid N$. En aquesta situació no podrem dir res més, i l'algoritme no proporcionarà cap factor no trivial de $N$.

3. Existeix un primer $p \mid N$ tal que $Q \equiv O \pmod{p}$ i també existeix un primer $q \mid N$ tal que $Q \neq O \pmod{q}$. En tal cas, suposem que $Q = (t : u : v)$. Per als primers $p \mid N$ tals que $Q \equiv O \pmod{p}$ es tindrà que
$$u \equiv 0 \pmod{p},$$
ja que aquesta és una propietat de les corbes el·líptiques sobre un cos. En canvi, per a la resta dels divisors primers $p \mid N$, es tindrà que $\mathrm{mcd}(u, p) = 1$. Per tant, tindrem que $1 < \mathrm{mcd}(u, N) < N$ i el nombre enter $\mathrm{mcd}(u, N)$ és un factor no trivial de $N$.

**4.2.2 Definició.** Si estem en la tercera situació direm que $Q$ factoritza $N$. En canvi, si ens trobem en la segona situació direm que $Q$ no factoritza $N$.

Resumim el procediment que hem exposat en un algoritme.

**4.2.3 Algoritme.** Sigui $N$ un enter lliure de quadrats i tal que $\mathrm{mcd}(6, N) = 1$.

1. Triem a l'atzar $x, y, a \in \{0 \ldots, N-1\}$.

2. Calculem $b := y^2 - x^2(x + a)$. Calculem $d := \mathrm{mcd}(4a^3 + 27b^2, N)$. Si $1 < d < N$ retornem $d$ i finalitzem. Si $d = N$ retornem un missatge d'error i finalitzem l'algoritme. Si $d = 1$ passem al pas 3.

3. Usem l'algoritme 2.3.3 per a calcular l'ordre $r$ del punt $P = (x : y : 1)$ en el grup $E(a, b)(\mathbb{Z}/N\mathbb{Z})$.

4. Si $r$ és senar retornem un missatge d'error i finalitzem. En cas contrari avancem al pas 5.

5. Calculem $Q = \frac{r}{2}P = (t : u : v)$.

6. Calculem $d := \text{mcd}(u, N)$. Si és $d = N$ retornem un missatge d'error i finalitzem (en aquest cas $Q$ no factoritza $N$). Si $1 < d < N$ retornem $d$ i finalitzem (és el cas en què $Q$ factoritza $N$).

Observem que en el pas 6 no podem tenir $d = 1$, ja que en aquest cas $Q = O$, fet que ja hem observat que no es pot donar.

Observem també que els càlculs de màxims comuns divisors es poden efectuar mitjançant l'algoritme d'Euclides i que el càlcul de $Q$ en el pas 5 es pot efectuar mitjançant un mètode d'exponenciació binària aplicat al grup $E(a, b)(\mathbb{Z}/N\mathbb{Z})$. Per tant, els passos 1, 2, 4, 5 i 6 es poden efectuar amb una despesa de $O(n^3)$ operacions.

Resta considerar la complexitat algorítmica del pas 3, que ja hem justificat que s'implementa en $O(n)$ espai i $O(n^3)$ operacions. En aquest sentit ens trobem que la complexitat de l'algoritme és la mateixa que la obtinguda per al mètode de Shor: necessitem $O(n)$ qubits i realitzar $O(n^3)$ operacions bàsiques per tal d'implementar l'algoritme.

Remarquem que l'algoritme falla en els casos en què $d = N$ en el pas 2, $r$ és senar o bé en els casos en què $r$ és parell i $Q$ no factoritza $N$.

## 4.3 Probabilitat d'èxit

Ens disposem a calcular la probabilitat que l'algoritme 4.2.3 retorni un factor no trivial de $N$. A diferència del càlcul de probabilitat que hem fet a la secció 3.2, en què el grup $(\mathbb{Z}/N\mathbb{Z})^*$ era fix, en la situació en què ens trobem ara el grup $E(a, b)(\mathbb{Z}/N\mathbb{Z})$ és aleatori. A més, l'estructura de $(\mathbb{Z}/N\mathbb{Z})^*$ és més senzilla que la de $E(a, b)(\mathbb{Z}/N\mathbb{Z})$.

Estudiem la probabilitat que l'algoritme falli en el pas 3.

**4.3.1 Proposició.** *Siguin $N$ un enter lliure de quadrats tal que $\text{mcd}(6, N) = 1$, $x, y, a \in \{0, \ldots, N-1\}$, $b := y^2 - x^2(x + a)$ i $d := \text{mcd}(4a^3 + 27b^2, N)$. Aleshores,*

$$P(d = N) = \frac{1}{N}.$$

DEMOSTRACIÓ: Denotem $\Delta(a,b) := 4a^3 + 27b^2$.

Tenim que

$$P(d = N) = P(\forall\, p \mid N \quad p \mid d) = \prod_{p|N} P(p \mid d).$$

Per a un primer $p \mid N$,

$$P(p \mid d) = P\left(y^2 \equiv x^3 + ax + b \text{ i } \Delta(a,b) \equiv 0 \pmod p\right).$$

Per tant, calcularem aquesta última probabilitat:

$$P\left(y^2 \equiv x^3 + ax + b \text{ i } \Delta(a,b) \equiv 0 \pmod p\right) =$$
$$\frac{1}{p^3} \#\left\{(x,y,a) \in \mathbb{F}_p^3 :\ y^2 \equiv x^3 + ax + b,\ \Delta(a,b) \equiv 0 \pmod p\right\} =$$
$$\frac{1}{p^3} \#\left\{(x,y,a,b) \in \mathbb{F}_p^4 :\ y^2 \equiv x^3 + ax + b,\ \Delta(a,b) \equiv 0 \pmod p\right\} =$$
$$\frac{1}{p^3} \sum_{a,b \in \mathbb{F}_p} \#\left\{(x,y) \in \mathbb{F}_p^2 :\ y^2 \equiv x^3 + ax + b,\ \Delta(a,b) \equiv 0 \pmod p\right\} =$$

$$\frac{1}{p^3} \sum_{a,b \in \mathbb{F}_p, \Delta(a,b) \equiv 0 \pmod p} \#\left\{(x,y) \in \mathbb{F}_p^2 :\ y^2 \equiv x^3 + ax + b \pmod p\right\}. \quad (4.2)$$

Si $\Delta(a,b) \equiv 0 \pmod p$, la corba projectiva sobre $\mathbb{F}_p$ definida per l'equació

$$y^2 z = x^3 + axz^2 + bz^3$$

és de gènere zero i és, per tant, birracional amb $\mathbb{P}_1(\mathbb{F}_p)$. En conseqüència,

$$\#\left\{(x,y) \in \mathbb{F}_p^2 :\ y^2 \equiv x^3 + ax + b,\ \Delta(a,b) \equiv 0 \pmod p\right\} = p.$$

Si apliquem aquest fet a l'equació 4.2 obtenim que aquesta darrera equació és equivalent a

$$\frac{1}{p^2} \sum_{a,b, \Delta(a,b) \equiv 0 \pmod p} 1 =$$
$$\frac{1}{p^2} \#\left\{(a,b) \in \mathbb{F}_p^2 :\ \Delta(a,b) \equiv 0 \pmod p\right\} =$$
$$\frac{p}{p^2} = \frac{1}{p},$$

ja que $\left\{(a,b) \in \mathbb{F}_p^2 :\ \Delta(a,b) \equiv 0 \pmod p\right\}$ és, de nou, el conjunt de punts afins d'una corba projectiva birracionalment equivalent a $\mathbb{P}_1(\mathbb{F}_p)$.

Obtenim que, per a un primer $p \mid N$,

$$P(p \mid d) = \frac{1}{p},$$

i si efectuem el producte per a cada divisor primer de $N$ obtenim el resultat desitjat. □

Observem que la probabilitat que l'algoritme falli en el pas 2 és molt petita si $N$ és gran. Podem modificar l'algoritme demanant que en cas de fallar en el pas 2 tornem al pas 1 un nombre finit de vegades. D'aquesta manera, el pas 2 retornarà un factor no trivial de $N$ o ens permetrà passar al pas 3 amb probabilitat molt gran.

Necessitem informació sobre l'estructura del grup de punts d'una corba $E(a,b)$ sobre $\mathbb{Z}/N\mathbb{Z}$ triada a l'atzar. El resultat següent ens resultarà particularment útil.

**4.3.2 Proposició.** *Siguin $p$ un nombre primer i $E$ una corba el·líptica sobre $\mathbb{F}_p$. Aleshores*

$$\left| P\left(2 \mid \#E(\mathbb{F}_p)\right) - \frac{2}{3} \right| \leq \frac{1+10\sqrt{2}}{\sqrt{p}}.$$

Demostració: Cas particular de [6, Theorem 1.1]. □

El teorema que segueix ens permetrà fitar la probabilitat de trobar un factor no trivial de $N$ mitjançant l'algoritme 4.2.3.

**4.3.3 Teorema.** *La probabilitat que l'algoritme 4.2.3 no proporcioni un factor no trivial d'un nombre enter $N = p_1 \cdots p_m$, amb $p_i \neq p_j$ si $i \neq j$ i $p_i \neq 2, 3$ primers, està fitada superiorment per la quantitat*

$$\left(\frac{2}{3} + \frac{3}{2}\varepsilon\right)^m + \left(\frac{2}{3} + \varepsilon\right)^m,$$

*on*

$$\varepsilon := \max_{1 \leq i \leq m} \frac{1+10\sqrt{2}}{\sqrt{p_i}}.$$

Demostració: Fitarem superiorment la probabilitat que l'algoritme falli. Suposem que hem triat un punt $P$ d'ordre $r$ per a la corba $E := E(a,b)$. Es té que

$$P(\text{ l'algoritme no factoritza } N) = P(2 \nmid r) + P(2 \mid r \text{ i } Q \text{ no factoritza } N).$$

Fitarem per separat cadascun dels sumands. Denotem per $r_i$ l'ordre de $P$ mòdul $p_i$. Pel teorema 4.1.7, es té que $r$ és senar si i només si $r_i$ és senar per a $1 \leq i \leq m$. Per tant,

$$P(2 \nmid r) = \prod_{i=1}^{m} P(2 \nmid r_i).$$

Estudiem cada factor en funció de la paritat de $E_{p_i}(\mathbb{F}_{p_i})$:

$$P(2 \nmid r_i) = P\left(2 \nmid \#E_{p_i}(\mathbb{F}_{p_i})\right) + P\left(2 \mid \#E_{p_i}(\mathbb{F}_{p_i})\right) P\left(2 \nmid r_i;\ 2 \mid \#E_{p_i}(\mathbb{F}_{p_i})\right),$$

on l'última probabilitat que apareix és una probabilitat condicionada. Aquesta expressió és equivalent a

$$\left(1 - P\left(2 \mid \#E_{p_i}(\mathbb{F}_{p_i})\right)\right) + P\left(2 \mid \#E_{p_i}(\mathbb{F}_{p_i})\right) P\left(2 \nmid r_i;\ 2 \mid \#E_{p_i}(\mathbb{F}_{p_i})\right),$$

expressió que podem fitar gràcies a la proposició 4.3.2. A més, quan $\#E_{p_i}(\mathbb{F}_{p_i})$ és parell es té que com a mínim la meitat dels punts són d'ordre senar. Per tant, obtenim la fita uniforme

$$P(2 \nmid r_i) \leq 1 - \left(\frac{2}{3} - \frac{1+10\sqrt{2}}{\sqrt{p_i}}\right) + \frac{1}{2}\left(\frac{2}{3} + \frac{1+10\sqrt{2}}{\sqrt{p_i}}\right) \leq \frac{2}{3} + \frac{3}{2}\varepsilon.$$

Per tant,

$$P(2 \nmid r) = \prod_{i=1}^{m} P(2 \nmid r_i) \leq \prod_{i=1}^{m} \left(\frac{2}{3} + \frac{3}{2}\varepsilon\right) = \left(\frac{2}{3} + \frac{3}{2}\varepsilon\right)^m.$$

Procedim a continuació amb la fita per al segon sumand. La probabilitat que volem fitar és

$$P(2 \mid r \text{ i } Q \text{ no factoritza}) = P(2 \mid r \text{ i } \forall i \quad \frac{r}{2}P \neq O \pmod{p_i}).$$

Sigui $s := v_2(r) \geq 1$. La condició $\frac{r}{2}P \neq O \pmod{p_i}$ és equivalent a demanar que sigui $v_2(r_i) = s$ per a tota $i$. D'altra banda, per a que $r$ sigui parell és necessari i suficient que algun dels $r_i$ sigui parell. Per tant,

$$P(2 \mid r \text{ i } Q \text{ no factoritza}) = P(\exists i \quad 2 \mid r_i \text{ i } \forall i \quad v_2(r_i) = s) = P(\forall i \quad v_2(r_i) = s).$$

Per tal que sigui $v_2(r_i) = s$ és condició necessària que $2^s \mid \#E_{p_i}(\mathbb{F}_{p_i})$. Per tant,

$$P(2 \mid r \text{ i } Q \text{ no factoritza}) \leq P(\forall i \quad 2^s \mid \#E_{p_i}(\mathbb{F}_{p_i})) \leq P(\forall i \quad 2 \mid \#E_{p_i}(\mathbb{F}_{p_i}))$$
$$= \prod_{i=1}^{m} P(2 \mid \#E_{p_i}(\mathbb{F}_{p_i})) \leq \left(\frac{2}{3} + \varepsilon\right)^m,$$

utilitzant novament per a l'última desigualtat la proposició 4.3.2. Si sumem les fites obtingudes per a cadascun dels sumands obtindrem la fita esperada. □

Per tal que la fita trobada sigui útil ens hem d'assegurar que $N$ no té divisors primers petits. La fita millora a mesura que la mida dels divisors primers de $N$ creix, i també a mesura que augmenta el nombre de divisors primers de $N$. No obstant, per a tot $\varepsilon$,

$$\left(\frac{2}{3}+\frac{3}{2}\varepsilon\right)^m+\left(\frac{2}{3}+\varepsilon\right)^m \geq 2\left(\frac{2}{3}\right)^m.$$

En el cas més difícil, en què $N = p_1p_2$ amb $p_1 < p_2$ primers, tindrem que

$$\left(\frac{2}{3}+\frac{3}{2}\varepsilon\right)^2+\left(\frac{2}{3}+\varepsilon\right)^2 \geq \frac{8}{9}.$$

A tall d'exemple, calculem quina ha de ser la mida de $p_1$ i $p_2$ per tal que l'algoritme 4.2.3 factoritzi $N$ amb probabilitat com a mínim $1/10$. Cal que

$$\left(\frac{2}{3}+\frac{3}{2}\varepsilon\right)^2+\left(\frac{2}{3}+\varepsilon\right)^2 \leq \frac{9}{10},$$

desigualtat equivalent a

$$60\varepsilon + 585\varepsilon^2 \leq 2,$$

o sigui, a

$$\varepsilon \leq \frac{-10+\sqrt{230}}{39}.$$

Caldrà, per tant, que

$$p_1 > 13098. \tag{4.3}$$

**4.3.4 Observació.** Notem que en la pràctica no és restrictiu suposar que $N$ no té divisors més grans que la fita (4.3) ja que, abans d'aplicar un mètode de factorització com els algoritmes 3.1.1 o 4.2.3, podem comprovar prèviament que $N$ no és divisible per primers petits.

**4.3.5 Exemple.** Estudiem la probabilitat que trenquem un criptosistema RSA de 1024 bits mitjançant l'algoritme 4.2.3.

Volem aplicar l'algoritme 4.2.3 a un nombre enter $N = pq$, amb $N$ enter de 1024 xifres en base 2, i $p$ i $q$ primers de 508 i 516 xifres en base 2, respectivament. Es tindrà que $p > 2^{507}$ i, per tant,

$$\sqrt{p} > 2^{253}.$$

En conseqüència,

$$\varepsilon \leq \frac{1+10\sqrt{2}}{2^{252}}.$$

Per tant, la probabilitat que cerquem està fitada inferiorment per

$$1-\left(\frac{2}{3}+\frac{3}{2}\varepsilon\right)^2-\left(\frac{2}{3}+\varepsilon\right)^2 \simeq \frac{1}{9}.$$

**4.3.6 Observació.** Si podem repetir l'execució de l'algoritme vàries vegades, la probabilitat total de trobar un factor no trivial de $N$ augmenta. Així, obtenim una manera de trobar un factor no trivial de $N$ amb probabilitat gran.

# Bibliografia


[1] C. H. Bennett. *Logical reversibility of computation*, IBM J. Res. Develop., 17 (1973), p.525-532.

[2] H. Cohen. *A Course in Computational Algebraic Number Theory*. New York: Springer-Verlag, 1996.

[3] R. P. Feynman. *Simulating physics with computers*. Int. J. Theor. Phys., 21 (1982), p.467.

[4] C.F. Gauss. *Disquisicions aritmètiques*. Barcelona: Societat Catalana de Matemàtiques, 1996.

[5] E. Fredkin, T. Toffoli. *Conservative logic*. Internat. J. Theoret. Phys., 21 (1982). p.219-253.

[6] E. W. Howe. *On the group orders of elliptic curves over finite fields*. Comp. Math., 85 (1993), p.229-247.

[7] M. A. Nielsen, I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge: Cambridge University Press, 2000.

[8] J. H. Silverman. *The Arithmetic of Elliptic Curves*. New York: Springer, 1986.

[9] P. W. Shor. *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*. SIAM J. Comp., 26 (1997), p.1484-1509.

[10] A. Travesa. *Aritmètica*. Barcelona: Edicions de la Universitat de Barcelona, 1998.